Converging Systems
Third-Party CS-Bus™ Device Driver Toolkit (DDK)
Programming Guide

Revision: 2.3.r
Date: December 19, 2015

Thank you in advance for deciding to create device drivers for one or more of our CS- Bus controller devices. We have prepared an easy–to-understand device driver toolkit to help you create the best possible device driver in the shortest period of time. Please review the next few sections in advance of proceeding to the body of this document to determine the specific type of device driver that makes the most sense for your products and your customers. Examples have also been included within Section V of this manual for ease in validating device driver work.

Step 1: Selection of Interface type. CS-Bus products can be supported through a variety of interface channels, including keypads, infrared remotes, dry contact, screen trigger interfaces, 0-10v dimmable fluorescent panels, RS-232-C and Ethernet (Telnet or UDP) communications. This paper focuses on the two last types of communications-RS-232-C and Ethernet (Telnet or UDP). ***We recommend that where possible new device drivers should support both RS-232-C as well as Ethernet (Telnet or UDP) communications to provide the greatest choice for your customer base.***

Please see the following section within this document depending on the type of device driver to be written.

Type of Communication	Section
RS-232-C	Section 1, A
Ethernet (Telnet)	Section 1, B
Ethernet (UDP)	Section 1, C

Step 2: Selection of Device Driver Type (one-way or by-directional). Two types of device drivers are possible with CS-bus devices. Standard, one-way drivers which simply issue a command are typical within the control user interface world. However, of late, many newer systems also permit bi-directional feedback (two-way drivers) to report the status of the device that is being controlled. Specifically, if a light is turned on, a user interface such as touchscreen could change the color of the touch button or status indicator based on the actual status of that light (whether On or Off). CS-Bus technology permits both types of drivers to be created.

Please see the following section within this document depending on the type of device driver to be written.

Type of Driver	Section
One directional (no feedback)	Section II & Section III
Bi-directional (feedback)	Section IV

Section I-Communication Options

Remote control of CS-Bus products is possible using simple ASCII commands over RS-232-C serial communications or identical ASCII commands using UDP Protocol over Ethernet with a pair of datagram sockets. Typically these ASCII commands are considered "control" commands which will cause one or more devices to respond (e.g. motor up/down, LED brightness up/down/off). Alternatively, there are other commands which are considered set-up or "commissioning" commands which are only supported through the e-Node™ Ethernet Adapter/Firewall using the eNode™ Pilot (commissioning) application. These "commissioning" commands are generally used only for setup for a supported device (e.g. review/set motor direction defaults, review LED light level presets, LED sequence rate and dissolve rate, review/set shared addresses, and review/set trigger-type events). These specialized "commissioning" commands are beyond the scope of this "CS-Bus Command Summary Programmer's Guide." Please consult Converging Systems for more information here.

Please refer to the applicable section below for communication parameters and software command syntax relating to the type of communication channel being used. Converging Systems recommends that for simplicity and for driver robustness, the programmer create a single unified driver that permits both RS-232-C as well as Ethernet connectivity.

A. RS-232-C Serial Communication. Serial communication is possible with devices which are connected to the CS-Bus using the IBT-100™ Intelligent Bus Translator/Firewall.

1. Serial Communication Parameters

Set up your serial communication software (from your third-party automation system) with the following serial communication parameters. These parameters cannot be changed within the IBT-100.

Baud Rate	Parity	Valid Data	Stop Bit	Hand-Shaking
57600	N	8	1	None

2. Software Command Syntax/Software Commands. Software commands are identical for Serial communication, Ethernet communication using Telnet, or Ethernet communication using UDP. See Section II below.

B. Ethernet Communication using Telnet (recommended). Ethernet Communication using a Telnet server option available within the e-Node Ethernet Adapter.

1. E-node Communication

Establish communication with the IP address of the e-Node ™ (i.e. 192.168.1.100 or other address for e-Node that has either been assigned through DHCP or static addressing) using the e-Node's Telnet Server functionality. It is recommended for most applications where a single control system (automation or lighting system) needs to

communicate with the e-Node Ethernet adapter that Telnet communication is adopted. For special applications where multiple systems need to communicate with the e-Node Ethernet adapter concurrently, one platform might select Telnet while the other platform might select UDP (see 1.C below). Alternatively, both platforms might select UDP communication. It should be noted that currently, the e-Node Ethernet adapter has only one (1) available Telnet channel available so more than one concurrent Telnet session on separate channels is not currently possible.

2. Telnet Server

The Telnet Server function is set by default on the e-Node to be **DISABLED**. In order to enable the Telnet server, go the **View e-Node** menu screen within the e-Node Pilot application and expand the "+" mark in front of the e-Node and select the **Telnet** entry. Next, select **Enable** for the Server function. Finally, hit the **Restart** button to restart the e-Node with the Telnet server enabled.

3. Telnet Server Port

The Telnet Server function is hardcoated within the e-Node to utilize Port 23.

Mode of Communication	Port
Transmission **to** and **from** e-Node	23

4. Authentication. The Telnet Server within the e-Node supports plaintext authentication. In the plaintext method, the user name and password are sent as plaintext for authentication by the Server. Authentication can be turned off or on as needed in order to operate with specific third-party control systems. Authentication can be DISABLED or ENABLED using the e-Node Pilot Application.

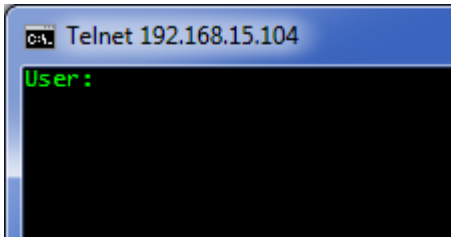
By default, the factory defaults for the username and passwords are as follows:

Plaintext Factory Default Parameter	Setting*
Username	E-NODE
Password	ADMIN

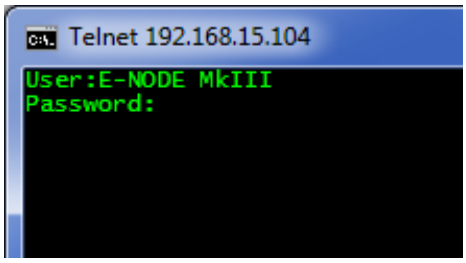
*Note: These settings can be changed either within the e-Node Pilot application for through the built-in e-Node webpage compatible with most browsers. See separate document entitled "Pilot Commissioning Guide."

.

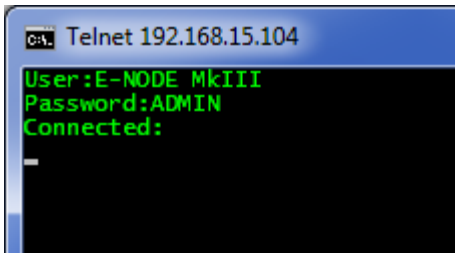
a. Initial Authentication. Provided **Telent** is **Enabled** and **Authentication** is **Enabled** within the E-Node, here is what appears on a Telnet session once Port 23 is accessed.



After the appropriate **User** name is provided (default is **E-NODE MkIII**), the system prompts the communicating software for **Password** as follows:



After the appropriate Password is provided (default is ADMIN), the system replies with "Connected," and is now ready for commands and queries.



b. Authentication Protection. Should there be a break in communication after the initial connection is made for various reasons (power failure, cable unplugged), the communicating software should monitor for "User" and "Password" and automatically trigger the above **Initial Authentication Procedure** to once again reestablish communication. This is the preferred approach over alternative approaches which might without monitoring bus traffic, send over **User** and **Password** data every "x" number of seconds as a precautionary technique but which adds a tremendous amount of overhead to the bus and should be avoided.

5. Software Command Syntax/Software Commands. Software commands are identical for Serial communication, Ethernet communication using Telnet, or Ethernet communication using UDP. See Section II below**.**

C. Ethernet Communication using UDP datagram sockets. Ethernet Communication using UDP (User Datagram Protocol) with a pair of datagram sockets is possible using the e-Node Ethernet Adapter.

1. E-node Communication

Establish communication with the IP address of the e-Node ™ (i.e. 192.168.1.1 or other address for e-Node that has either been assigned through DHCP or static addressing). For more information, see "CS-Bus Controllers Installation, Programming and Interface Guide."

2. Datagram Sockets

Mode of Communication	UDP Port
Transmission **to** e-Node (send port for third-party controlling computer)	5000*
Inquiry of Data **from** e-Node (listen port for third party controlling computer)	4000*

*Note: UPD ports can be changed from the above factory defaults with the e-Node Pilot application. See separate document entitled "Pilot Commissioning Guide." Go the **View e-Node** menu screen and select **UDP** and change to

your preferred ports within Pilot.

Establish communication with the IP address of the e-Node™ (i.e. 192.168.1.1 or other address for e-Node™ that has either been assigned through DHCP or static addressing). For more information, see "CS-Bus Controllers Installation, Programming and Interface Guide."

3. Software Command Syntax/Software Commands. Software commands are identical for Serial communication, Ethernet communication using Telnet, or Ethernet communication using UDP. See Section II below**.**

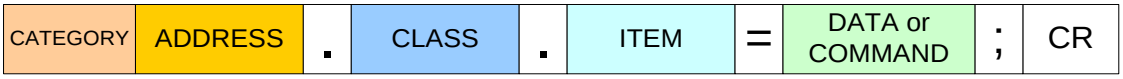
Section II-Syntax and Commands

In order to address a particular CS-Bus compatible device, it is necessary to have previously assigned a unique **Z.G.N**. (zone/group/node) address to each device on the CS-Bus. For more information on how to assign addresses refer to the appropriate documentation that came with your particular device. If you do not know the particular address of a target unit, you can use the wildcard descriptor instead.

#0.0.0.P=CC;↵

However, if you use this format, all units on the bus will respond to that command. Certain commands (e.g. Q=QY) will only respond to a particular address that does not have a "**0**" value within the **Z** or **G** or **N** fields.

Messages are in the dot delimited format as:



Field		Description	Type*
CATEGORY	#	Forward messages (standard type messages directed toward device on CS-bus)	1,2,3,4,5,6,7
	!	Response Positive or an unsolicited message (Signal)	9,11
	*	Negative response	10
ADDRESS	ZONE . GROUP . NODE	ZGN address format Three divisions for Zone, Group, Node. Each with a value 0 -> 254	
. CLASS	. DEVICE	In a Type 1 command message (standard command/action type) it is the device type (LED, MOTOR)	2,3,9
. ITEM	. ITEM	Specifies the item information If this field is omitted then the data field is assumed to be a	

Field		Description	Type*
	command.		
	OMITTED	Assumes Type 1 command message	1
= DATA	= DATA	This is the value of the any data	3
	= ?	Query message	2,4,5
	= COMMAND	Command	1

*Type refers to type of message more thoroughly described in the CS-Bus Messaging manual. For the purposes of standard device drivers, please disregard.

Section III-Supported Control Commands

Supported software commands are specified below.

Note: If a command is valid, the command will be echoed back. If however the command is invalid, an asterisk "*****" followed by the partial command, will be returned. In case of the "*", check the command and try again.

Messages

Data/Command			Comment	IMC-100T	IMC-100C/ BRIC[1]	ILC-100	ILC-400[4]	DMX e-Node MkIII
Device	Item	Data / Command						
.MOTOR		=DOWN	***Motor Down*** Commands port to send motor Down	✓	✓			
		=UP	***Motor Up*** Commands port to send motor Up	✓	✓			
		=STOP	***Motor Stop*** Commands port to stop motor	✓	✓			
		=RETRACT	***Motor Retract*** Moves all motors to the "home" position		✓			
		=STORE**,**#	***Stores Preset Position*** *Stores current motor position to a preset at* # (1-n) Note-see specific models below for available values -BRIC. Values for # are (1-9) -IMC-100C . Values for # are (1-6)		✓			

Data/Command			Comment	IMC-100T	IMC-100C/ BRIC[1]	ILC-100	ILC-400[4]	DMX e-Node MkIII
Device	Item	Data / Command						
		=RECALL,#	***Recalls Preset Position*** *Moves motors to a previously stored preset value at # (0-n)* Note: See applicable range for values under =STORE,# RECALL,0 is special value that cannot be user programmed and is the "home" position		✓			
	.STATUS	=?	***MOTOR STATUS*** Specifies the status of motor. Returns a binary 8 bit value. Bit 0 – Motor Active Bit 1 – Direction (0-retract, 1-deploy) Bit 2 – Fully extended limit Bit 3 – Fully retracted limit Bit 4 – intermediate stop Bit 5 – Position reading valid		✓			
	.PRESET.X	=XX.XX =?	***MOTOR PRESET SET or READ BACK*** Sets or retrieves the stored value of an Preset location (i.e. typically 1-9) for the position of the motor from the home to the fully deployed (range is 0.00 to 100.00%)		✓ (not on BRIC)			
	.POSITION	=?	***MOTOR POSITION READ BACK*** Retrieves the position of the motor from the home to the fully deployed (range is 0.00 to 100.00%)		✓ (not on BRIC)			
.LED		=DISABLE	***Inhibit Activity LED*** Disables on-board amber "Activity" LED	✓				
		=ENABLE	***Enable Activity LED*** Enables on-board amber "Activity" LED	✓				
		=ON	***Activity LED ON*** Turns "activity" LED on	✓				
		=ON =ON:XX	***P(ulse) W(idth) M(odulation) LED ON*** Turns on PWM LED to state when OFF was issued --- **Optional Ramping Parameter:** Time (XX) can be set from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII			✓ ≥3.7	✓3/4cs ✓Mono ≥1.10	✓
		=OFF =OFF:XX	***PWM LED OFF*** Turns PWM LED off --- **Optional Ramping Parameter:** Time (XX) can be set from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII	✓	✓	✓ ≥3.7	✓3/4cs ✓Mono ≥1.10	✓
		=FLASH	***Activity LED FLASH MODE*** Repeatedly flashes (on-board amber) "activity" LED	✓				

Data/Command			Comment	IMC-100T	IMC-100C/ BRIC[1]	ILC-100	ILC-400[4]	DMX e-Node MkIII
Device	Item	Data / Command						
		=EFFECT,#	***PWM EXECUTES AN EFFECTS PROCESSOR*** Executes a previously stored effect. # (1-4) 1=Sequences from Preset 1 through sequential Presets until the End Point Preset with programmed RGB values of 240.240.240 or programmed RGBW values of 240.240.240.240 is encountered at which point the cycle jumps back to Preset 1 (without displaying the End Point Preset) and repeats in perpetuity until a "Stop" or subsequent command is encountered. 2=Flame 3=Sequence – Cycles through the entire color wheel (with saturation and brightness maintained) 4=Random Color -- Cycles through colors, but hue and saturation are generated at random (,# is currently not required and defaults to EFFECT,1) Note: Effects is terminated when further commands are received **Compatibility**: Primary function supported by V.x.x or later of ILC-100 firmware (for Effect 1-4), V x.x or later of ILC-400 firmware (for Effects 1-4), and V.0.2 or later of e-Node/DMX MII firmware (for EFFECT,1 only).			✓	✓3 cs	✓
		=STORE,#	***STORE PRESET LEVEL*** Stores a lighting preset at # (1-24) Note: any combination of hue, saturation, brightness describing a "color" can be saved in this manner.			✓	✓3/4cs ✓Mono	✓
		=RECALL,# =RECALL,#:XX	***RECALL PRESET LEVEL*** Recalls a previously stored preset. # (1-24) --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII			✓ ≥3.7	✓3/4cs ✓Mono ≥1.10	✓
		=SUN,S =SUN,S:XX	***RECALL A CLB-CHRONOBIOLOGICAL LIGHTING LEVEL (i.e. Circadian Rhythm)*** Executes a factory stored lighting continuum for representing sunlight from natural nighttime to noon-day sun. S (0-240) 0=Nighttime-no illumination 240=Full noon-day sun Note: Level is terminated when further commands are received --- **Optional Ramping Parameter:** Time (N) can be from 0 sec (instantaneous) to 64800 sec (18hr)[5] **Compatibility**: Ramp option requires e-Node MIII				✓4cs ≥1.10	n/a

Data/Command			Comment	IMC-100T	IMC-100C/ BRIC[1]	ILC-100	ILC-400[4]	DMX e-Node MkIII
Device	Item	Data / Command						
		=SUN_UP =SUN_UP:XX	***PWM CLB UP*** Scrolls through CLB levels upwards (to noon-day sun) Note: Ramping continues until interrupted by receiving a "STOP" command --- **Optional Ramping Parameter:** Time (N) can be from 0 sec (instantaneous) to 64800 sec (18hr)[5] **Compatibility**: Ramp option requires e-Node MIII				✓3 cs ≥1.10	
		=SUN_DOWN =SUN_DOWN:XX	***PWM CLB DOWN*** Scrolls through CLB levels downwards (to black) Note: Ramping continues until interrupted by receiving a "STOP" command. --- **Optional Ramping Parameter:** Time (N) can be from 0 sec (instantaneous) to 64800 sec (18hr)[5] **Compatibility**: Ramp option requires e-Node MIII				✓3 cs ≥1.10	
	.DISSOLVE.X **Note**: Firmware releases prior Version 2.03 supported <u>global</u> changes only of all dissolve settings in unison without the trailing **.X** variable. Commands without the **.X suffix** are interpreted similarly as a **.DISSOLVE.0** or wildcard command to change all possible variables of Dissolve in unison.	=XX	***PWM SET DISSOLVE RATE*** .DISSOLVE.X=XX Specifies the rate XX (in seconds) that it takes to transition from one state to another for a particular Dissolve feature (X) Value for (X) 1—Dissolve function for transitions between from one state and another using direct value commands such as SET,L; SAT, S; HUE,H; RED,R, GREEN,G, BLUE,B; COLOR;VALUE 2—Dissolve function for transitions between ON and OFF and between PRESETS (RECALL,X) 3—Dissolve function for transition from one state another with the following effects -EFFECT(1) -EFFECT(4) 4—Time to complete a full cycle with the following EFFECT function. (Min is 14 seconds—max is 240 seconds -EFFECT(3) 0—Wildcard command to change all possible Dissolve Functions in unison **Compatibility**: Primary function supported by V.2.03 or later of ILC-100 (for individual Dissolve settings #1-#3) and V.3.3 or later (for Dissolve #4).			✓ ✓ ✓ ✓ ✓	✓3/4cs ✓Mono ✓3/4cs ✓Mono ✓3/4cs ✓Mono ✓3/4cs ✓Mono ✓3/4cs ✓Mono	✓ ✓ ✓ n/a n/a
	▪SEQRATE	=XX	***PWM SET SEQUENCE RATE*** Used with the following effects to switch from one preset to another. Specifies the time (after any dissolve) that the preset color is maintained before transitioning to the next in sequence. -EFFECT(1) -EFFECT(4)			✓	✓3/4cs	✓

Data/Command			Comment	IMC-100T	IMC-100C/ BRIC[1]	ILC-100	ILC-400[4]	DMX e-Node MkIII
Device	Item	Data / Command						
	HSV Color Space Adjustments (preferred method)[2]							
		=HUE_UP	***PWM HUE UP*** Scrolls through colors red->green->blue Note: Ramping continues until interrupted by receiving a "STOP" command			✓	✓3/4cs	✓
		=HUE_DOWN	***PWM HUE DOWN*** Scrolls through colors red->blue->green Note: Ramping continues until interrupted by receiving a "STOP" command			✓	✓3/4cs	✓
		=HUE,H =HUE,H:XX	***PWM SET HUE VALUE*** Sets hue target, and dissolves to a hue of value H (0-240) --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII			✓ ≥3.7	✓3/4cs ≥1.10	✓
		=SAT_UP	***PWM SATURATION UP*** Fades up saturation Note: Ramping continues until interrupted by (i) receiving a "STOP" command or (ii) until an upper or lower limit is reached			✓	✓3/4cs	✓
		=SAT_DOWN	***PWM SATURATION DOWN*** Fades down saturation Note: Ramping continues until interrupted by (i) receiving a "STOP" command or (ii) until an upper or lower limit is reached			✓	✓3/4cs	✓
		=SAT,S =SAT,S:XX	***PWM SET SATURATION LEVEL*** Sets saturation target, and dissolves to a saturation of value S (0-240) --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII			✓ ≥3.7	✓3/4cs ≥1.10	✓
		=FADE_UP	***PWM FADE UP*** Fades up brightness (lightness) Note: Ramping continues until interrupted by (i) receiving a "STOP" command[1] or (ii) until an upper or lower limit is reached			✓	✓3/4cs ✓Mono	✓
		=FADE_DOWN	***PWM FADE DOWN*** Fades down brightness (lightness) Note: Ramping continues until interrupted by (i) receiving a "STOP" command[1] or (ii) until an upper or lower limit is reached			✓	✓3/4cs ✓Mono	✓

Data/Command			Comment	IMC-100T	IMC-100C/ BRIC[1]	ILC-100	ILC-400[4]	DMX e-Node MkIII
Device	Item	Data / Command						
		=SET, L =SET, L:XX	***PWM SET BRIGHTNESS LEVEL*** Fades to a brightness (lightness) L (0-240) --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII			✓ ≥3.7	✓3/4cs ✓Mono ≥1.10	✓
		=STOP	***PWM ADJUSTMENT STOP*** Stops the selected auto ramping (fade, saturation, or hue as applicable)			✓	✓3/4cs ✓Mono	✓
		=HSV,H.S.V =HSV,H.S.V :XX	***PWM "HSV" COLOR SETTING*** Specifies the color in HSV (range is 0 to 240) and dissolves to that color. (i.e. .COLOR=? returns specifics of present color. If in transition, it is the color at that instant). --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII			 ≥3.7	✓4cs ≥1.10	>3.7
	.COLOR	=H.S.V	***PWM "HSV" COLOR SETTING (legacy command)*** Specifies the color in HSV (range is 0 to 240) and dissolves to that color. (i.e. .COLOR=? returns specifics of present color. If in transition, it is the color at that instant).			✓	✓3/4cs	NA (see LED=COLOR,H.S.L)
	.PRESETH.X	=XXX.XXX.XXX	***SET PRESET (HSV Color Space) (legacy command)*** Specifies the color in HSV mode (range is 0-240) for preset #. (i.e. .PRESETH.X=? returns specifics of color in HLS mode for preset "X" while .PRESETH with no modifier returns the color in HLS of whatever the current color setting is. PRESETH.0 is a wildcard for setting all devices in unison.) **Note**: Setting Presets do not affect the current LED state.			✓	✓3/4 5 s	
	RGB Color Space Adjustments[2]							
		=RED,R =RED,R:XX	***PWM SET RED LEVEL*** Sets red channel target, and fades to a hue of value R (0-240). --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII			✓ ≥3.7	✓3/4cs ≥1.10	✓

Data/Command			Comment	IMC-100T	IMC-100C/ BRIC[1]	ILC-100	ILC-400[4]	DMX e-Node MkIII
Device	Item	Data / Command						
		=GREEN,G =GREEN,G:XX	***PWM SET GREEN LEVEL*** Sets red channel target, and fades to a hue of value G (0-240). --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII			✓ ≥3.7	✓3/4cs ≥1.10	✓
		=BLUE,B =BLUE,B:XX	***PWM SET BLUE LEVEL*** Sets red channel target, and fades to a hue of value B (0-240). --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII			✓ ≥3.7	✓3/4cs ≥1.10	✓
		=WHITE,W =WHITE,W:XX	***PWM SET WHITE LEVEL*** Sets white channel target, and fades to a white value of W (0-240). --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII				✓3/4cs ≥1.10	✓
		=RGBW,r.g.b.w =RGBW,r.g.b.w :XX	***PWM "RGBW" COLOR SETTING*** Specifies the color in RGBW (range is 0 to 240) and dissolves to that color. --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII				✓4cs ≥1.10	
		=RGB,r.g.b =RGB,r.g.b:XX	***PWM "RGB" COLOR SETTING*** Specifies the color in RGB (range is 0 to 240) and dissolves to that color. --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII			✓ ≥3.7	✓3cs ≥1.10	✓
	.VALUE	=R.G.B	***PWM "RGB" COLOR SETTING (legacy command—see new command LED=RGB,R,G,B)*** Specifies the color in RGB (range is 0 to 240) and dissolves to that color. (i.e. .VALUE=? returns specifics of present color. - With a 3 color device return data structure will be the format of #Z.G.N.LED.VALUE=R.G.B -With a 4-color device, the return data structure will be in the format of #Z.G.N.LED.VALUE=R.G.B.W -With a 1-color device, the return data structure will be in the format of #Z.G.N.LED.VALUE=W).			✓	✓3 cs	NA

Data/Command			Comment	IMC-100T	IMC-100C/ BRIC[1]	ILC-100	ILC-400[4]	DMX e-Node MkIII
Device	Item	Data / Command						
	.VALUE	=R.G.B.W	***PWM "RGBW" COLOR SETTING (legacy command - see new command LED=RGB,R,G,B,W))*** Specifies the color in RGB (range is 0 to 240) and dissolves to that color. (i.e. .VALUE=? returns specifics of present color. - With a 3 color device return data structure will be the format of #Z.G.N.LED.VALUE=R.G.B -With a 4-color device, the return data structure will be in the format of #Z.G.N.LED.VALUE=R.G.B.W -With a 1-color device, the return data structure will be in the format of #Z.G.N.LED.VALUE=W).				✓4 cs	NA
	.PRESET.X	=XXX.XXX.XXX	***SET PRESET (RGB Color Space)*** Specifies the color in RGB or preset # (range is 0 to 240). (i.e. .PRESET.X=? returns specifics of color for preset "X" while .PRESET with no modifier returns the color in RGB of whatever the current color setting is. PRESET.0 is a wildcard for setting all devices in unison.) **Note**: Setting Presets do not affect the current LED state. **Compatibility**: Primary function supported by V.1.8 of ILC-100 firmware. Primary function implementing wildcard replacement for preset # supported by V.2.03 of ILC-100.			✓	✓3 cs	✓
	.PRESET.X	=XXX.XXX.XXX.XXX	***SET PRESET (RGBW Color Space)*** Specifies the color in RGBW or preset # (range is 0 to 240). (i.e. .PRESET.X=? returns specifics of color for preset "X" while .PRESET with no modifier returns the color in RGBW of whatever the current color setting is. PRESET.0 is a wildcard for setting all devices in unison.) **Note**: Setting Presets do not affect the current LED state.				✓4cs	✓
	Correlated Color Temperature (CCT) Adjustments[3]							
		=CCT,XXXX =CCT,XXXX:XX	***SETS LED COLOR TO CCT*** Specifies the color temperature (CCT) of RGB or RGB+W luminaries in degrees Kelvin (range is 1800 to 7000). --- **Optional Ramping Parameter:** Time (XX) can be from 0 sec (instantaneous) to 64800 sec (18 hrs.)[5] **Compatibility**: Ramp option requires e-Node MIII			✓ ≥3.2 ≥3.7	✓3 c ✓4 c ≥1.10	

Data/Command			Comment	IMC-100T	IMC-100C/ BRIC[1]	ILC-100	ILC-400[4]	DMX e-Node MkIII
Device	Item	Data / Command						
		=CCT_UP	***CCT FADE UP*** Fades up CCT (color temperature) Note: Ramping continues until interrupted by (i) receiving a "STOP" command[1] or (ii) until an upper limit is reached			✓ ≥3.2	✓3 c ✓4 c ≥1.10	
		=CCT_DOWN	***CCT FADE DOWN*** Fades down CCT (color temperature) Note: Ramping continues until interrupted by (i) receiving a "STOP" command[1] or (ii) until a lower limit is reached			✓ ≥3.2	✓3 c ✓4 c ≥1.10	

General Notes-- If entry within "Item" column is blank, do not include.

Footnotes:

[1] These commands are available through the e-Node Ethernet Adapter with the BRIC only. Separate RS-232c serial commands are available in a separate device driver guide for the BRIC.

[2] These settings can also be performed using UID address or Z.G.N address. (FW Version 1.8 or later)

[3] These settings can also be performed using UID address (TBD) or Z.G.N address. (FW Version 3.2 or later)

[4] Device has RGB mode ("3cs"), RGB+W mode ("4cs") and 1-4 channel Monochrome Mode ("mono"). Various commands are applicable in respective modes. When a function is available in "3cs" and "4cs" it is referred to as "3/4cs"

[5] Ramp Time Optional Parameter. If a value of XX is not provided, the current set times for Dissolve,1 and Dissolve,2 will be read alternatively. This new in-line command/ramping alternative syntax is particularly useful for new software drivers which integrate in some cases a ramp time variable along with a lighting command in one data stream (as opposed to second data stream setting the Dissolve Rate before a specific lighting command is invoked). This saves on bus traffic and is ideally suited toward those software applications which prefer to issue a lighting command and specific dissolve rate in one data stream.

Section IV-Bi-Directional Driver Background

A. Bi-Directional Messaging Overview. CS-Bus controllers support two different type of bi-directional messaging.

Recommended Approach. The primary and recommended method is for the user interface (UI) ***to monitor response traffic*** on the CS-Bus and constantly update the UI's own registers to provide accurate feedback of current status to the user, often in the form of icons, graphics, color state changing buttons or similar type of user feedback. **This approach is recommended** for it does not burden the CS-Bus with any type of superfluous traffic thereby and therefore provides a higher degree of reliability for those CS-Bus devices which rely on the CS-bus for their communication needs. See Section "B.1" below as well as the examples in Section V to determine if this approach is best for you.

Secondary Approach. The second method and generally the less preferable method permits the user interface (UI) to poll the status from a particular device on the CS-Bus (with a particular Z/G/N assignment) and then upon receipt of the response back from that device, integrate the response into icons, graphics or other type of user feedback. Typically this approach works well when a user presses a "status" button on his specific user interface and then and only then, is the inquiry inserted onto the CS-bus. ***The derivative of this approach that is highly NOT recommended has the UI constantly polling each and every device on the CS-Bus which introduced a tremendous amount of unplanned traffic onto the CS-Bus***. For obvious reasons, this constant polling approach (rather than the occasional polling method) is highly NOT recommended. See Section "B.2" below for syntax and more information.

B. Messaging Structure. The messaging structure for the above two approaches is described below. Before settling on a particular message approach for bi-directional communication, ***make sure to read all sections below first***.

1. **Bus Monitoring Approach**. When a standard command is transmitted onto the CS-Bus, such as a command to turn on an LED or move a MOTOR, the exact same command after being received by the target controller (i.e. motor controller or LED controller) is mirrored back onto the CS-Bus. See **Table** below for a standard command transmitted on the CS-Bus and the return message.

Message Generated by Transmitting Device (keypad, IR, serial, Ethernet (UDP), etc.) onto CS-Bus	**Return Message Appearing on CS-Bus**
#2.1.1.LED=ON; ↵	#2.1.1.LED=ON; ↵
#1.1.1.MOTOR=UP; ↵ **and then** #1.1.1.MOTOR=STOP; ↵ **at a particular location**	#1.1.1.MOTOR=UP; ↵ **and then** #1.1.1.MOTOR=STOP; ↵ **(note: no exact positioning is available with this type of bus monitoring)**

2. **Status Inquiry Approach**. This approach permits specific Inquiry Message types to be transmitted onto the CS-Bus to collect current status on a particular device on the CS-Bus (with a specific Z/G/N address only). ***This approach only works if a device has been previously "Discovered" and has been assigned a Node address other than the factory default of zero (i.e. the Z/G/N address of #1.1.1 or #2.1.1 for instance contains a specific node address of a "1" while the default factory condition of #1.1.0 or #2.1.0 has an undiscovered or null node setting of "0").*** Typically, unless the Discovery process was employed to set a non-zero Node

address, the Status Inquiry Approach cannot be used. Please see Appendix 1 within this document for a simple procedure that can be used to assign a non-zero value to the device's Node address. **Once this procedure is followed, THEN AND ONLY THEN can the Status Inquiry Approach can be utilized.** Consider the warnings described above under "Secondary Approach" before selecting this approach. See Table below for Inquiry-type Messages that can be inserted onto the CS-Bus in this case.

Message Generated by Transmitting Device (keypad, IR, serial, Ethernet (UDP), etc.) onto Bus	**Inquiry Command Syntax**	**Return Message Appearing on CS-Bus** (available with FW versions 2.5 or later of IMC-100x and version 1.8 or later of ILC-100 and later) **Note**: these messages are returned from the device. If the bus is being monitored, all messages generated by transmitting devices will also be seen on the bus prior to the below return messages.
#2.1.1.LED=RECALL,1; ↵ (this sets LED to a particular preset level in this case Preset #1)	#2.1.1.LED.STATUS=? (this is to inquire what the current RGB setting is)	!2.1.1.LED.STATUS=xxx.xxx.xxx; (this returns back the actual setting of the LED which in this case was set to Preset #1.) (See note[1])
	#2.1.1.LED.PRESET.1=? (this is to inquire what the stored value of Preset 1 is)	!2.1.1.LED.PRESET.1=xxx,xxx,xxx; (See note[1])
#1.1.1.MOTOR=UP; ↵ **and then** #1.1.1.MOTOR=STOP; ↵ **at a particular location**	#1.1.1.MOTOR.POSITION=? (this is to inquire what the current position of the motor is)	!1.1.1.MOTOR.POSITION=xx.xx; **(note: exact positioning is available with this type of bus monitoring on IMC-100T)**

Note: [1] An alternative also exists, that an inquiry of a device using its UID (unique ID) can be made as well. The concept of a UID is beyond the scope of this document and is thoroughly described within the separate "CS-Bus Messaging Manual." For programming purposes, the UID's suffix "n" is a discrete number between 1 and 65535.

Section V-Programming Examples

A. One-Way Device Drivers

1. Programming Examples-For IMC-100 Controller (Motor Controller)

Desired Action	Syntax	Notes
Activate motor to move downwards to a particular location		
If you know the address	**#1.1.1.MOTOR=DOWN;↵** (followed by a)	This is a specific Z/G.N address for the unit to be controlled in a specific direction
	#1.1.1.MOTOR=STOP:↵	To stop the automatic movement at a particular location. Note: If a STOP command is not sent, the automatic movement continues until the end of the continuum is reached (either a "0" or a "100")
If you do ***not*** know the address	**#0.0.0.MOTOR=DOWN;↵** (followed by a)	This is a wildcard Z/G/N address for all units to move in common direction
	#0.0.0.MOTOR=STOP:↵	To stop the automatic movement at a particular location Note: If a STOP command is not sent, the automatic movement continues until the end of the continuum is reached (either a "0" or a "100")

2. Programming Examples-For ILC-100 Controller (Lighting Controller)

Action	Syntax	Notes
Turn LED load to Preset #1		
--if you know the address	**#2.1.1.LED=RECALL,1;↵**	This is a specific Z/G.N address for the unit to be controlled
--if you do not know the address	**#0.0.0.LED=RECALL,1;↵**	This is a wildcard Z/G/N address for all units to move in common direction
Turn Up the brightness of a particular color (hue) already ON to a particular level	**#2.1.1.LED=FADE_UP;↵** (followed by a)	To initiate the automatic fade UP command (this continues until a STOP command is sent)
	#2.1.1.LED=STOP;↵	To stop the automatic fade at a particular level. Note: If a STOP command is not sent, the automatic FADE continues until the end of the continuum is reached (either a "0" or a "240").
Change the Dissolve Rate to a new dissolve rate	**#1.1.1.LED.DISSOLVE=3;↵**	To change the LED dissolve rate to a dissolve rate of 3 (seconds).

Turn LED load to OFF		
--if you want to turn a single controller OFF	**#2.1.1.LED=OFF;↵**	This is a specific Z/G.N address for the unit to be controlled
--if you want to turn all controllers OFF	**#0.0.0.LED=OFF;↵**	This is a wildcard Z/G/N address for all units to turn OFF

B. Two-Way (Bi-Directional) Device Drivers

Refer to the appropriate section below for the type of approach utilized to receive two-way feedback. See Section IV.A for more information

1. Two-Way Drivers Using Bus Monitoring Approach

a. Programming Examples-For IMC-100 (Motor) Controller (Bus Monitoring Approach)

Desired Activity	**Monitored Bus Traffic-No Specific Inquiry Command sent in this case**	**Notes**
Determine location of motors connected to multiple Motor Controllers		No single bus traffic command can be used to determine this information. See entry below for method by which each device individually can be monitored
Determine location of specific motor connected to a specific Motor Controllers		No single bus traffic command can be used to determine this information. See entry below for method by which specific commands directed toward a specific device can be monitored
Determine direction motor traveled last time a command was issued	**#1.1.1.MOTOR=DOWN; ↵**	With the "Bus Monitoring Approach" only the direction that motor was commanded to move is available.
Determine if a motor was directed to move to a specific Intermediate Preset	**#1.1.1.MOTOR=RECALL,2; ↵**	This monitored command, indicates that the motor with a particular Z/G/N address of 1.1.1 was directed to move to Preset #2. Note: if a system allows additional movement by using a UP or DOWN button this information may not be accurate if a subsequent UP or DOWN command was issued.
Determine if the specific motor has reached a "home" position		***Note:*** [1]
Determine if a specific motor connected to a Motor Controller has reached its "fully extended position"		***Note:*** [1]

Note [1]: Unique capabilities exist that require additional documentation-please consult Converging Systems' Technical Support for more information.

b. Programming Examples-For ILC-100 (Lighting) Controller (Bus-Monitoring Approach)

Desired Activity	Monitored Bus Traffic-No Specific Inquiry Command sent in this case	Notes
Determine the status of all LED loads connected to all Lighting Controllers		No single bus traffic commands can be used to determine this information. See entry below for method by which each device individually can be monitored
Determine status of LED load of a particular Lighting Controller	**#2.1.1.LED=PRESET,1; ↵**	Monitor last message for particular LED. In this case, the Lighting Controller with bus address Z/G/N of 2.1.1. was instructed to move to Preset #1.

2. Two-Way Drivers Using Status Inquiry Approach

a. Programming Examples-For IMC-100 (Motor) Controller (Status Inquiry Approach)

Desired Activity	Invoked Inquiry Command	Notes
Determine the status of all motors connected to all Motor Controllers	**Not Applicable**	Inquiries can only be directed toward a specific Motor Controller (one with a particular ZGN address including a node address)
Determine the stored Intermediate Preset location (#1 in this case) for a particular Motor Controller (with a particular Z/G/N address)	**#1.1.4.MOTOR.PRESET.1=?; ↵**	Determines the stored data value for Preset #1 for a Motor Controller with a Z/G/N address of 1.1.4. ***(under development)***
Determine the position of the motor for a particular Motor Controller (with a particular Z/G/N address)	**#1.1.4.MOTOR.POSITION=?; ↵**	Determines the current location for connected motor to a Motor Controller with Z/G/N address of 1.1.4. ***(under development)***

b. Programming Examples-For ILC-100 (Lighting) Controller (Status Inquiry Approach)

Desired Activity	Invoked Inquiry Command	Notes
Determine the status of all LED loads on multiple Lighting Controllers	**Not Applicable**	Inquiries can only be directed toward a specific Lighting Controller (one with a particular ZGN address including a node address)

Determine status of LED load on a particular Lighting Controller with a specific Z/G/N address	**#2.1.1.LED=STATUS; ↵**	Determines the Red, Green, Blue levels for a particular Lighting Controller with a Z/G/N address of 2.1.1.
Determine setting for a particular Preset within a particular Lighting Controller)	**#2.1.1.LED=PRESET,4; ↵**	Determines the Red, Green, Blue levels for a previously set Preset #4 for a particular Lighting Controller with a Z/G/N address of 2.1.1

Appendix 1
Procedure to Assign a CS-Bus Device a Non-Zero Node address or UID Address

Bi-directional communication is a unique feature of CS-Bus controllers. This type of two-way communication enables innovative device drivers to be constructed that can actually poll a device to inquire of its status. This would be particularly useful where a touchscreen application was desired that could show the relative position of an element (window covering, projection screen, etc.) connected to a motorized device or the status or color being driven by a lighting controller. Two-way communication is available either using the IBT-100 serial interface adapter or the e-Node Ethernet interface adapter. You must have either one of these devices in order to enable bi-directional communication.

Such two-way communication, however, requires that a particular targeted CS-Bus controller (motor controller or lighting controller) has a unique address assigned such that it can be polled accurately. CS-Bus controllers come from the factory with a "wildcard" or null settings (of "0") which allows them to operate "out of the box" quickly BUT which prevents them from being polled or queried without some type of installer intervention.

This section has been provided to assist the installer in assigning a unique **Z/G/N** address to a CS-Bus device such that the unique bi-directional communication feature can be utilized. Depending upon what hardware is available to the installer, specific procedures may or may not be available. See the logic table below for the correct procedures given the hardware available to you.

Step	Questions
Step #1	Do you have an e-Node Ethernet adapter. If yes, go to Step #2, otherwise go to Step #3
Step #2	With an e-Node, you can easily program a non-zero Z/G/N address into a targeted CS-Bus controller. Refer to Section A below for easy directions on how to program a unique non-zero Z/G/N address into a CS-bus device. **Note: you can also initially program the CS-Bus controller with a discrete UID address and then simply update its existing Z/G/N address using a simple application. Please refer to Section B below for this more complicated procedure.**
Step #3	Do you have an IBT-100 serial interface adapter. If yes, go to Step #4 below, otherwise go to Step #5
Step #4	With an IBT-100 serial interface adapter, you can easily program a non-zero Z/G/N address into a CS-Bus device. Refer to Section C below for easy directions on how to program a non-zero Z/G/N address into a CS-bus device. **Note: You cannot program a UID address into a CS-bus device with an IBT-100. Greater flexibility is available using the UID addressing scheme; however, bidirectional communication is still possible using the IBT-100. If you are interested in learning more about the types of enhanced features available with the e-Node, contact your dealer for more information.**
Step #5	If you do not have an IBT-100, please contact your dealer to secure either an IBT-100 or an e-Node Ethernet interface adapter.

A. Programming a non-zero Z/G/N Address into a CS-Bus Controller using e-Node (Quick Approach)

If you have an e-Node, and wish to enable bi-directional polling and query mode on a targeted CS-Bus controller, it is preferable to assign a UID address to the targeted CS-Bus controller first before tampering with Z/G/N addresses. ***If you are certain that your installation is either a new installation where previous addresses have not been made, then this procedure would be advised.*** If this is the case, please follow the directions within Section B instead.

However, if you do not wish to go through the process of assigning a unique UID to a CS-Bus controller, you can still assign a non-zero Z/G/N address into a targeted CS-Bus controller using an e-Node. Refer to the following table for necessary steps here.

User Steps	**Resulting Action**
1. Connect an e-Node to a targeted CS-Bus device. See the directions that came with your e-Node on its proper setup and configuration. Launch the e-Node Pilot application.	
2. Select the "**View Traffic**" menu option and right click within the "**e-Nodes Found**" box. Then select "**Refresh."**	You should see within 5-50 seconds the name of your e-Node if it is properly configured appear within the "**e-Node Found"** window, provided the e-Node is powered up, and operational on your network
3. Determine what address you wish to assign to your CS-Bus controller. If you have a simple system with just one ILC-100 lighting controller, use the address of #2.1.1. If you have a simple system with just one IMC-100 motor controller, use the address of #1.1.1. Proceed to Step #4 below for directions here. ***If you have a more complicated system, please consult the factory for more information.*** **Note: If your CS-Bus controller is part of a multi CS-Bus controller installation where there are more than one motor and/or lighting controllers interfaced together, it is likely that the individual CS-Bus controller(s) have already been assigned unique Z/G/N addresses (in addition to having unique UID addresses assigned as well)! In this case, it would be wise to verify any pre-existing address assignments. If you were to change a Z/G/N address assignments. the pre-existing (working) system will most likely FAIL. To avoid this type of disaster, please following the steps below within Section B to determine if your device(s) already has pre-existing UID addresses assigned. Should they have pre-existing UID addresses assigned, it is most likely that one or more non-zero Z/G/N addresses have also been assigned—therefore your system is already enabled for bi-directional communication and query mode.**	
4. Select the "**View Traffic**" menu option. If your target CS-Bus controller is an ILC-100 lighting controller go to Step #5 below, if your targeted CS-Bus controller is an IMC-100 motor controller go to Step #6 below.	
5. ILC-100 Address setting. Type in the following command within the "**Message window**" and implement the exact address within the command stream below that you which the targeted CS-Bus controller to respond. If you wish to change the address specified below, simply substitute your new address with the three fields after the "#" header. **#2.1.1.LED=OFF** Then select the "**Send**" button. Now proceed to	

Step #7. **Note: the e-Node Pilot application automatically adds a semicolon and a carriage return after every command sequence within the Message Window so it is unnecessary with this application to add these required components.**	
6. IMC-100 Address setting. Type in the following command within the "**Message window**" and implement the exact address within the command stream below that you which the targeted CS-Bus controller to respond. If you wish to change the address specified below, simply substitute your new address with the three fields after the "#" header. **#1.1.1.MOTOR=UP** Then select the "**Send**" button. Now proceed to Step #7. **Note: the e-Node Pilot application automatically adds a semicolon and a carriage return after every command sequence within the Message Window so it is unnecessary with this application to add these required components.**	
7. Now, press and hold the Discovery Button (small push button) on the targeted CS-Bus Controller for approximately **5** seconds during which time the on-board LED signaling lamp will turn off and then turn back on. Once it turns back on, ***immediately*** release the Discovery Button.	If this procedure has been properly followed, the targeted CS-Bus controller will have been assigned a new Z/G/N address and bi-directional communication and query mode is now possible

B. Programming a non-zero UID Address into a CS-Bus Controller using e-Node prior to setting a non-zero Z/G/N address (More Fullproof/ComprehensiveMethod)

If you have an e-Node, and wish to enable bi-directional polling and query mode on a targeted CS-Bus controller, it is often preferable to assign a UID address to the targeted CS-Bus controller (***only if one has not already been assigned***) prior to randomly modifying the internal addresses of a CS-Bus controller. Refer to the following table for necessary steps here. After a unique UID address is assigned to a CS-Bus controller, it is quick work to update a null address to a non-zero value. The directions within this step should be followed in this case.

User Steps	**Resulting Action**
1. Connect an e-Node to a targeted CS-Bus device. See the directions that came with your e-Node on its proper setup and configuration. Launch the e-Node Pilot application.	
2. Select the "**View Traffic**" menu option and right click within the "**e-Nodes Found**" box. Then select "**Refresh."**	You should see within 5-50 seconds the name of your e-Node (if it is properly configured) appear within the "**e-Node Found"** window, provided the e-Node is powered up, and operational on your network.
3. Determine if your targeted device has already been assigned a UID address. Check this by selecting the "**View Map**" menu option and selecting the "**Discover Devices"** button**.** Provided (i) that your device is properly connected and (ii)	

your e-Node is properly connected and configured, if no device entry **name** for a CS-Bus controller appears under your Discovered e-Node entry, your targeted CS-Bus device has no UID address assigned. In this case, you may now assign a unique UID address to the CS-Bus controller. Proceed to Step #4 below. **Note**: **If you do see the targeted CS-Bus controller appear under your Discovered e-Node listing, then you have confirmed that the targeted CS-Bus device has already been assigned a unique UID address and now you can easily update a previous null value setting for a Z/G/N address using e-Node and the e-Node Pilot application. Proceed to Step #6 here.**	
4. Setting UID Address. Determine what address you wish to assign to your CS-Bus controller. If you have a simple system with just one lighting or motor controller, use the address of "1." Proceed to Step #5 below for directions here. ***If you have a more complicated system, please consult the factory for more information.*** **Note: If your CS-Bus controller is part of a multi CS-Bus controller installation where there are more than one motor and/or lighting controllers interfaced together, it is likely that the individual CS-Bus controllers already have a unique UID address assigned! In this case check for its unique address already assigned by following the directions within Step #5 below.**	
5.Select the "View Map" menu option and enter a unique UID address greater than 0 (but less than 65535) within the empty box under the "**UID Set"** button and then select the "**Set**" button. You will then be instructed to depress the Discovery Button (small push button) on the targeted CS-Bus Controller. **Immediately** after depressing the Discovery button, release it (within 0.5 second) and the LED signaling indicator will immediately flash back on. Now proceed to Step #6	If this procedure has been properly followed, the presence of the just "Assigned" CS-Bus controller will appear under the CS Network icon on the left side of this menu screen. Your CS-Bus controller now has a unique UID address and bi-directional communication is now possible. Proceed to Step #6 below.
6. Viewing the details of a UID assigned device. While still in the "View Map" view, left click first on the **?** icon in front of the targeted CS-Bus controller and then left click once again on the **+** mark in front of the specific CS-Bus controller to expand its contents. Left click on the "**Bus**" entry and search for the primary Address listing. If the primary Address has a zero in the most left field, you have confirmed that it has a factory default setting of a zero-node value. In order to set this to a non-zero value, proceed to Step #7.	
7. Setting non-zero Z/G/N address. Determine what address you wish to assign to your CS-Bus controller. If you have a simple system with just one ILC-100 lighting controller, use the address of #2.1.**1**. If you have a simple system with just one IMC-100 motor controller, use the address of #1.1.**1**. To change this value, simply double click on the triad address within the Properties window until it the triad address turns blue (i.e. 2.1.1). Then modify the address using your keyboard	If this procedure has been properly followed, the targeted CS-Bus controller will have been assigned a new Z/G/N address and bi-directional communication and query mode is now possible.

accordingly. Then hit “Enter” on your keyboard to save the new entry. ***If you have a more complicated system, please consult the factory for more information.*** **Note: If your CS-Bus controller is part of a multi CS-Bus controller installation where there are more than one motor and/or lighting controllers interfaced together, it is likely that the individual CS-Bus controller have already been assigned unique UID address(es) as well as non-zero Z/G/N addresses! If you see that the addresses have non-zero entries for the primary Address, you should not change these values—bidirectional communication and inquiry mode is already enabled. Should you attempt to change the previously set non-zero addresses, this effort might conflict with one or more other logical operations, and your existing system WILL FAIL. Please contact your dealer for more information here.**	

C. Programming a non-zero Z/G/N Address into a CS-Bus Controller using IBT-100

If you have an IBT-100, and wish to enable bi-directional polling and query mode on a targeted CS-Bus controller, it is easy to assign a non-zero Z/G/N address to the targeted CS-Bus controller. Refer to the following table for necessary steps here.

User Steps	**Resulting Action**
1. Connect an IBT-100 to a targeted CS-Bus device. Make sure that it is plugged into the Powered Port on the CS-Bus controller or the IBT-100 will not operate. See the directions that came with your IBT-100 on its proper setup and configuration	
2. Launch Hyperterminal or similar application (Baud rate is 57,600, N, 8,1, None for IBT-100).	
3. Determine what address you wish to assign to your CS-Bus controller. If you have a simple system with just one ILC-100 lighting controller, use the address of #2.1.1. If you have a simple system with just one IMC-100 motor controller, use the address of #1.1.1. Proceed to Step #4 below for directions here. ***If you have a more complicated system, please consult the factory for more information.*** **Note: If your CS-Bus controller is part of a multi CS-Bus controller installation where there are more than one motor and/or lighting controllers interfaced together, it is likely that the individual CS-Bus controller have already been assigned unique UID address(es)! You will not be able to check for this with Hyperterminal and the IBT-100--an e-Node is required here. Should you attempt to enter a new address which conflicts with one or more existing address, your existing system WILL FAIL. Please contact your dealer for more information here.**	
4. If your targeted CS-Bus controller is an ILC-100 lighting controller go to Step #5 below, if your targeted CS-Bus controller is an IMC-100 motor controller go to Step #6 below.	
5. ILC-100 Address setting. Type in the following command within typing window of Hyperterminal and implement the exact address within the	You should see your commands reflected back onto the typing window as you type these commands. Should they not appear, you know

command stream below that you which the targeted CS-Bus controller to respond. If you wish to change the address specified below, simply substitute your new address with the three fields after the "#" header. **#2.1.1.LED=OFF; ↵** Then select the "**Send**" button. Now proceed to Step #7.	that either your Hyperterminal is not properly configured, that you have selected the incorrect COM port or something else is wrong with your system. Typically the IBT-100 is not plugged into the Powered Port of the CS-Bus controller.
6. IMC-100 Address setting. Type in the following command within the "**Message window**" and implement the exact address within the command stream below that you which the targeted CS-Bus controller to respond. If you wish to change the address specified below, simply substitute your new address with the three fields after the "#" header. **#1.1.1.MOTOR=UP; ↵** Then select the "**Send**" button. Now proceed to Step #7.	You should see your commands reflected back onto the typing window as you type these commands. Should they not appear, you know that either your Hyperterminal is not properly configured, that you have selected the incorrect COM port or something else is wrong with your system. Typically the IBT-100 is not plugged into the Powered Port of the CS-Bus controller.
7 Now, press and hold the Discovery Button (small push button) on the targeted CS-Bus Controller for approximately **5** seconds during which time the on-board LED signaling lamp will turn off and then turn back on. Once it turns back on, ***immediately*** release the Discovery Button.	If this procedure has been properly followed, the targeted CS-Bus controller will have been assigned a new Z/G/N address and bi-directional communication and query mode is now possible