CONVERGING SYSTEMS

TechNotes

Revision 7/7/2024

Zone/Group/Node Addressing

(don't assign the same ZGN Addresses to multiple devices*)

Overview:.

CS-Bus controllers can be address with a unique **Z**one/**G**roup/Node (ZGN) address. Up to 254 entries can be used for each field. The first field is the **Zone** (or largest range), the middle field is the **Group**, and the last field is the **Node**. **No two loads can share the same Z/G/N address**. As an example, if you will be populating a pair of two controllers within each of two rooms on two floors of a building here would be the suggested addressing that could be used.

	Floor One	**Floor Two**
Room 1	2.1.1 for first controller in room. 2.1.2 for second controller in this room	
Room 2	2.2.1 for first controller in room. 2.2.2 for second controller in this room	
Room 3		3.1.1 for first controller in room. 3.1.2 for second controller in this room
Room 4		3.2.1 for first controller in room. 3.2.2 for second controller in this room

Group Addressing. In certain cases, it is desirable is simply send a **wildcard** address for a group of controllers to all respond in unison rather than programming each individually to respond through macros. There are two problems with macros in general. One is that often they are executed serially which means that if you had two hundred loads referenced within a macro, the timing of the execution of the last command sent out might be delayed from the first command sent out. In this case, not all LEDs would turn on or OFF at the same time, potentially. The second issue involves the actual programming time required to program scores or even hundreds of commands for a simple ALL OFF button.

Within the CS-Bus software protocol is the concept of utilizing a "0" within any address field as a surrogate for defined numbers ranging from 1 to 254 within that same field. Thus, if you issued a

command of #2.1.0.LED=ON:<cr>, all units with addresses of 2.1.1 to 2.1.254 would immediately respond. Please see the table below for an example of how various wildcards could be used.

Specific controller address	Specific command that will trigger targeted controller
2.1.1	2.1.0 or 2.0.0 or 0.0.0
2.1.2	2.1.0 or 2.0.0 or 0.0.0
2.1.3	2.1.0 or 2.0.0 or 0.0.0
2.2.1	2.2.0 or 2.0.0 or 0.0.0
2.2.2	2.2.0 or 2.0.0 or 0.0.0
2.2.254	2.2.0 or 2.0.0 or 0.0.0
5.254.4	5.254.0 or 5.0.0 or 0.0.0

NOTIFY Command Background Converging Systems has a **NOTIFY** function which automatically provides color state feedback (from the targeted controller) provided a unique **Z**one/**G**roup/**N**ode (**Z/G/N**) address is provided with an action/argument payload to that specific controller. Specifically, if a command to invoke a color change is directed to a controller that has a **Z/G/N** address of 2.1.1, that specific controller with that address will respond back to the automation system as to its specific color state if and only if there is a color state change impacted on that specific controller.

In some cases, as has been discussed above, there might be a requirement to send a group command or all hail command to more than one controller. In this case, the group command would be directed not to a single controller or load but to a series of controllers. To reduce bus traffic when a series of controllers is given the same command, ***the status of the first controller whose node number is 1 greater than the wildcard command of "0" will respond and will be automatically remapped to the wildcard address of "0" from which the command emanated**** (which reduces bus traffic by up to 243 messages). The logic here is that if 254 controllers are all told to turn **Red**, only the surrogate for that group of controllers will respond and within the CS-Bus messaging logic that surrogate is the controller with a node of "1." So, for example, if a **#2.1.0.LED.VALUE=240.0.0:<cr>** command is transmitted to 254 controllers, they will all turn to **Red**, but only the controller with an address of **2.1.1** will respond with its new color status. In this case, a command on the bus from that surrogate controller would come back as follows: **!2.1.1.LED.VALUE=240.0.0** (the exclamation mark indicates that it is a message from CS-Bus device rather from an automation controller). Please see the diagram on the next page for the theory of operation here.

*Note: this is in 2018 updates to our ILC-400 firmware initially

SHADOW Address Background Exclusive to the e-Node 2000/4000 which supports one to 254 ILC-xx0 controllers, a new/secondary addressing scheme is available. This secondary addressing scheme (called **Shadow Addressing**) allows any ILC-xx0 to be additional connected to separate/additional addressing groups easily for which a larger set of ILC-xx0 devices is desired to be triggered while still allowing individual or smaller group control when just the primary ZGN address is invoked.

As an example, assume that there are four ILC-xx0 devices located in the dining room (with addresses of **2.1.1**, **2.1.2**, **2.1.3** and **2.1.4**). Each of those individual (dining room) devices can be controlled initially (without **Shadow** address) in two ways

-one, with their individual address (i.e., **2.1.1**) and
-two, with a wildcard address (i.e., 2.1.**0** for all of four of them operating together).

But now let's introduce **Shadow Addresses**. A Shadow address allows one area (our **2.1.x** devices) to be logically linked to other octets (i.e., like a **3.1.x** or **4.1.x** address). So, for our example, assume now that we have Kitchen with devices with addresses of **3.1.1/3.1.2**/etc. and a Den with devices with addresses of **4.1.1/4.1.2**.etc. By using the additional feature of **Shadow Address**, we could get all the Kitchen lights (**2.1.x**) to mimic the actions of those in the Kitchen or Den. In effect, a primary address of any ILC-xx0 controller can now be locally associated with two other ZGN addresses (or wildcard ZGN addresses).

Here is our example in a tabular format

ILC-xx0 unit	Primary **Z**one/**G**roup/**N**ode address	Shadow 1 Address	Shadow 2 Address
Kitchen 1	2.1.1	3.1.0	4.1.1
Kitchen 2	2.1.2	3.1.0	4.1.1
Kitchen 3	2.1.3	3.1.0	4.1.1
Kitchen 4	2.1.4	3.1.0	4.1.1
Dining Room 1	3.1.1	5.1.1	4.1.0
Dining Room 2	3.1.2	5.1.1	4.1.0
Den Unit 1	4.1.1	6.1.1	
Den Unit 2	4.1.2	6.1.1	

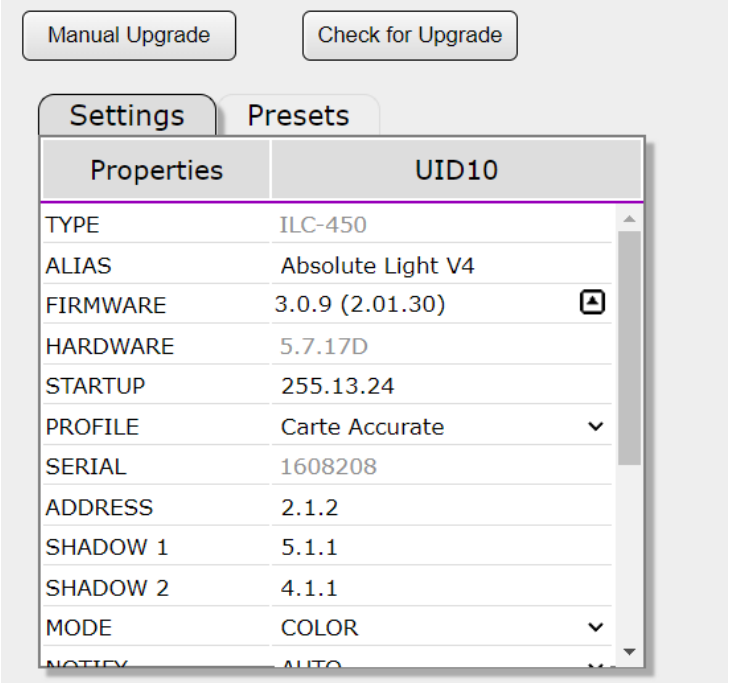
Summary of what this type of programming will provide
-each unit in the **Kitchen** can be controlled individually (i.e., 2.1.1 will control Kitchen 1)
-all of the **Kitchen lights** can be controlled with a Wildcard (i.2., a 2.1.0 will control all of the Kitchen
-each unit in the **Dining room** can be controlled individually (i.e., 3.1.1 and 3.1.2 will work with each
-all units in the **Dining Room** can be controlled together (i.e., a 5.1.1 will control them both).
-each unit in the **Den** can be controlled individually (i.e., 4.1.1 and 4.1.2 will work with each
-all units in the **Den** can be controlled together (i.e., a 6.1.1 will control them both).

-Now in a special case, if all lights in the Kitchen and the Dining Room and Den need to operate together, a 4.1.0 wild card can be triggered

In summary
-all of the Kitchen units can be controlled together,
-all units in the Dining Room can be controlled together
-all units in the Dining Room + Kitchen can be controlled together,
-all units in the Den can be controlled together, and finally
-all units I the Kitchen + Dining Room + Den can be controlled together

Why is this important?
In very large systems, sending over hundreds of commands to control large systems (one light at time) not only requires significant dealer programming time, but it also can overwhelm bus traffic. Since Converging System does make automation systems, we are simply streamlining data communication for these vendors in order to optimize responsiveness of the overall system.

-



*Note: For those of you who have made it this far, it is possible to address two (or more) devices with the same **ZGN** address for certain cases. In this event and ***although this is not generally recommended***, turn off **NOTIFY** for the second and any other duplicate devices. The issue here is that if a command is sent to a particular **ZGN** address and **NOTIFY** is on for all of them, all units with the same

ZGN address will try to send a bi-directional feedback unsolicited response **at the same time** which may impact the reliability of back-channel information and the responsiveness of the bus. Turning off NOTIFY for duplicate units eliminates that bus-conflict. But the primary reason we do not recommend assigned duplicate ZGN address to devices, is that you have lost the ability to turn on a single unit (which although you may not see a need for it today, the homeowner always comes up with new ideas later on which eliminates truck roll if you follow our advice initially.