

## **Developer Notes for INSTEON Firmware Update I2CS**

Revision Date = 2012-02

We are making improvements in device firmware to help ensure INSTEON commands are sent in a more reliable manner. This includes the adding of checksum (CS) to commands that set a remote device's properties, bit flags byte and database records. Also, internal protections have been added such as the ability to prevent device from acting on a message from a device not currently in its database, enhanced buffer corruption prevention and preventing invalid hop count messages from being hopped on the network.

INSTEON devices shipping with these changes will be incompatible with any software that does remote linking, database reads or writes and property reads and writes with currently shipping INSTEON devices and that has not implemented the changes described in this document.

## List of changes:

Failed commands will be NAKed with the following error code in CMD2

Error codes:

- 0xFF = Sender's device ID not in responder's database
- 0xFE = Load sense detects no load (confirm this one)
- 0xFD = Checksum is incorrect
- 0xFC = Pre NAK in case database search takes too long
- 0xFB = illegal value in command

Note: Arrange controller records (0xE2) to be in the first 150 links in the database to ensure database search does not return NAK 0xFC.

All direct commands will be ignored if the sender's ID is not in the I2CS device's database with the exceptions below. Additionally, INSTEON devices with Database NAK enabled will also reply with a NAK and 0xFF in cmd2 to indicate that the ID is not in the database. (See Appendix A for list of supported devices)

### Direct messages that do not require sender's ID in database

Command Name	Cmd1
Exit Link Mode	0x08
Enter Link Mode Extended	0x09
Enter Unlink Mode Extended	0x0A
Status Request (This will be implemented going forward, not yet shipped to developers)	0x19

Example of Database NAK enabled device response:

```
02 62 10 06 D0 0F 11 80
02 62 10 06 D0 0F 11 80 06
02 50 10 06 D0 18 D4 1D AB11 FF
```

INSTEON Engine Version command (0x0D) will return 0x02 in cmd2 for I2CS.

- 0x00 = I1 INSTEON device
- 0x01 = I2 INSTEON device
- 0x02 = I2CS enabled device
- NAK 0xFF = I2CS enabled device and sender ID not in responder dB

Peek/Poke commands removed

Enter Linking Mode command (0x09) is now an extended message and requires checksum if enabled

Exit Linking Mode command (0x08) added. This can be sent with or without the extended bit flag set.

Checksums added to certain commands. Data14 will contain a 2s compliment of cmd1 through 2nd to last data record in the last data record (See Appendix B for checksum calculation)

Database writes will be checksummed.

Property writes will be checksummed.

Bit flag writes (20) will be checksummed.

The return data of a Database Read will have a checksum

Only XXF or XX7 are allowed for parameters for an address of a record in a 2F write command

Success Report Broadcast added to some devices.

Application Retries added to some devices.

X10 Removed in some devices.

## Instructions on remotely linking to a reliable INSTEON device with the PLM

(Note: Linking can also be accomplished by manually going into linking mode using the Set button. See the specific device's manual for instructions)

Test to see if the device includes these I2CS enhancements.

Send the "INSTEON Engine Version" command (0x0D) and check for one of these responses:

1. Device returns a NAK (The sender's ID is not in the I2CS device's database)

```
02 62 10 06 D0 0F 0D 00
02 62 10 06 D0 0F 0D 00 06
02 50 10 06 D0 18 D4 1D AB 0D FF
```

2. 0x02 in cmd2 (The sender's ID is in the I2CS device's database)

```
02 62 10 06 D0 0F 0D 00
02 62 10 06 D0 0F 0D 00 06
02 50 10 06 D0 18 D4 1D 2B 0D 02
```

Go into local linking mode on the PLM to finish linking:

```
02 64 01 01
02 64 01 01 06
```

Send "go into linking mode" extended with calculated checksum to device:

```
02 62 10 06 D0 1F 09 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F6
02 62 10 06 D0 1F 09 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F6 06 02
50 10 06 D0 18 D4 1D 2B 09 01
02 53 01 01 10 06 D0 00 00 00
```

Example of command sent with an incorrect checksum with NAKed response:

```
02 62 10 06 D0 1F 09 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F5
02 62 10 06 D0 1F 09 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 F5 06
02 50 10 06 D0 18 D4 1D AB 09 FD
```

Linking complete; the device now has the sender's INSTEON ID in its database.



## Success Report Broadcast

Devices that support the new Success Report Broadcast will send the following message at the end of a group broadcast (usually triggered by a button push)

### Standard length Success Report Broadcast Command

Description: Sent immediately following a group broadcast.

Example (Hex): AA BB CC 11 03 01 CF 06 01 (where AA.BB.CC is the Device's ID, cleanup of cmd1 = 0x11, group = 0x01, 1 out of 3 devices failed to cleanup correctly)

SD Command	Message Direction	From Address (3 bytes)	To Address (3 bytes)	Message type	Cmd1 (1 byte)	Cmd2 (1 byte)	Notes
<b>Broadcast cleanup</b>	To device	Sender's ID	Hi byte = cmd1 being Cleaned up  Med byte = Number of devices to be cleaned up  Lo byte = Group Number	Group Broadcast	0x06	0x00 -> 0xFF  (Number of Failed Cleanups)	

## Application Retries

On I2CS enabled devices that include the Application Retries feature, control software has the ability to control the number of times cleanup messages are sent after group broadcasts (usually triggered by a button push on the device). Data1 of the device's database record controls the way the cleanups are handled. It may be useful to set the number of cleanups to a higher number for controlled devices that are of such high importance that the command absolutely needs to be confirmed as received. Examples of these important devices can include pool or spa pumps or irrigation controllers.

### All-Link Database (ALDB/L) Overview

The ALDB/L starts at the top of external (serial) EEPROM and grows downward. Top of memory is 0x0FFF. Each ALDB/L Record is 8 bytes long, so the first record starts at 0x0FF8, the second record starts at 0x0FF0, and so on down to 0x0300 for a total of 416 links. In what follows, the 3-byte INSTEON Address contained in a record is called the *Device ID* or sometimes just the *ID*. The high byte (MSB) of the Device ID is *ID2*, the middle byte is *ID1*, and the low byte (LSB) is *ID0*.

### External EEPROM Structure Overview

Location		Comments
0x0FF8	0xA2 01 AA BB CC FF FE 00	All-Link Database Responder Record
0x0FF0	0xE2 01 AA BB CC FF FE 00	All-Link Database Controller Record
0x0FD8		
.....		
0x0300		Last Record, 416 total links allowed
0x02XX	N/A	Addressing below 0x0300 is ignored by database

### ALDB/L Record Format

Database entries for controller records

Linear ALL-Link Database (ALDB/L) Record Format		
Field	Length (bytes)	Description
Record Control	1	0xE2 (for controller record) See <b>Record Control Info</b>
Group	1	ALL-Link Group Number of button on device. 0x01 for SwitchLinc type devices 0x01->0x08 for KeypadLinc type devices.
ID	3	Device ID (ID2, ID1, ID0 in that order)
Data 1	1	<b>0x00 = Broadcast for a cleanup</b> <b>0xFF = no clean-ups</b> <b>0x01-&gt;0xFE Application retries if needed as clean-ups</b>
Data 2	1	0x00 -> 0xFF (ignored value)
Data 3	1	0x00 for SwitchLinc type devices 0x01->0x08 for KeypadLinc type devices.

## Record Control Info

BIT							
7	6	5	4	3	2	1	0
Active	Controls me	1	SmartHop	SmartHop	0	High Water	0

Note: Bits 3 and 4 of the database FLAGS byte contain a smart hop count. These bits are set for controller records and can update every time the local device cleans up that record based on the cleanup ACKs received hop count. This is to pick the "best" starting hop count for a device. Hop counts can be either 0,1,2 or 3. bits 3 and 4 should be cleared when manipulating the database and they will ratchet up if needed.

Example:

0xAA =

1010 1010 =

(Active, controller, 1, smarthop = 1, so start cleanup at 1 hop, 0, high water set, 0



## Appendix A:

List of devices and associated feature set:

Note: items in yellow text are TBD

	X10	ID NAK	Checksum NAK	Broadcast	App Retries
KeypadLinc Dimmer		X	X	X	X
KeypadLinc Relay		X	X	X	X
KeypadLinc Relay DB (277V)		X	X	X	X
SwitchLinc Dimmer DB 1000W		X	X	X	X
SwitchLinc Dimmer DB 600W		X	X	X	X
SwitchLinc Relay Beeper	No X10	X	X	X	X
SwitchLinc Dimmer Beepr	No X10	X	X	X	X
ToggleLinc Dimmer Beeper	No X10	X	X	X	X
OutletLinc Dimmer DB		X			
OutletLinc Relay	No X10	X			
LampLinc Dimmer DB		X			
FanLinc		X			
240V Controller	No X10	X			
ApplianceLinc	No X10	X			
In-LincLinc Dimmer	No X10	X			
In-LineLinc Relay	No X10	X			
LampLinc	No X10	X			
LampLinc Dimmer , 3 pin (2456D3)	No X10	X			
Outdoor ApplianceLinc	No X10	X			
RemoteLinc 2			X		
Motion Sensor (Installer)			X		
TriggerLinc			X		
RF Modem 25J11 (InHome)			X		
Venstar	No X10		X		
IOlinc	No X10		X		
Access Point	N/A				
RemoteLinc	N/C				
ControLinc	N/C				

iGateway	PLM				
PLM	PLM				
SmartLinc	PLM				
TouchLinc	PLM RF	X	X	X	X

## Appendix B:

Example of Checksum:

01 02 03 04 05 06 1F 2F 00 01 02 0F FF 08 E2 01 08 B6 EA 00 1B 01 11
From 01.02.03 to 04.05.06
a record at 0FFF (A valid boundary)
08 bytes a record that 04.05.06 will control
Group 1 the responder is 08.B6.EA (00 1B 01 DNC)
11 is the check sum

Int	Hex	
47	2F	
0	00	
1	01	
2	02	
15	0F	
255	FF	
8	08	
226	E2	
1	01	
8	08	
182	B6	
234	EA	
0	00	
27	1B	
1	01	
1007	3EF	Sum
	10	Compliment (Last byte)
	11	Add 1