INSTE⏻N

Modem Developer's Guide

smartlabs

Table of Contents

Revision History

Release Date	Author	Description
01-30-07	PVD	Abstracted from INSTEON Developers Guide.
02-12-07	PVD	Added daughter card sections.
02-14-07	PVD	Released for proofreading.
03-01-07	PVD	More information on the PLM, comparison to PLC.
03-27-07	PVD	Fixed bytecount in IM Command **0x62** *Send INSTEON Standard or Extended Message*.
03-28-07	PVD	Added IM Command **0x58** *ALL-Link Cleanup Status Report*.
03-29-07	PVD	Updated explanation of IM Command **0x6F** *Manage ALL-Link Record*.
04-02-07	PVD	Updated explanation of IM Commands **0x61** *Send ALL-Link Command*, **0x56** *ALL-Link Cleanup Failure Report*, and **0x58** *ALL-Link Cleanup Status Report*.
04-06-07	PVD	IM Command **0x58** *ALL-Link Cleanup Status Report* also sent when IM interrupts its own Cleanup sequence.
04-17-07	PVD	Corrected <X10 Flag> value in IM Commands **0x63** *Send X10* and **0x52** *X10 Received*.
04-19-07	PVD	Added page number subscripts to links.
09-14-07	PVD	Fixed command number typo for IM Command **0x72** *RF Sleep*.
10-11-07	PVD	Rewrote explanation for IM Command **0x6F** *Manage ALL-Link Record*, enumerated <ALL-Link Record Flags> in IM Command **0x57** *ALL-Link Record Response*. Renamed Powerline Modem as PowerLinc Modem.
10-12-07	PVD	Added *PowerLinc Modem (PLM) Quick Start Guide* section

Legal Information

Terms of Use

This INSTEON Modem Developer's Guide is supplied to you by SmartLabs, Inc. (SmartLabs) in consideration of your agreement to the following terms. Your use or installation of this INSTEON Modem Developer's Guide constitutes acceptance of these terms. If you do not agree with these terms, please do not use or install this INSTEON Modem Developer's Guide.

In consideration of your agreement to abide by the following terms, and subject to these terms, SmartLabs grants you a personal, non-exclusive license, under SmartLabs' intellectual property rights in this INSTEON Modem Developer's Guide, to use this INSTEON Modem Developer's Guide; provided that no license is granted herein under any patents that may be infringed by your works, modifications of works, derivative works or by other works in which the information in this INSTEON Modem Developer's Guide may be incorporated. No names, trademarks, service marks or logos of SmartLabs, Inc. or INSTEON may be used to endorse or promote products derived from the INSTEON Modem Developer's Guide without specific prior written permission from SmartLabs, Inc. Except as expressly stated herein, no other rights or licenses, express or implied, are granted by SmartLabs and nothing herein grants any license under any patents except claims of SmartLabs patents that cover this INSTEON Modem Developer's Guide as originally provided by SmartLabs, and only to the extent necessary to use this INSTEON Modem Developer's Guide as originally provided by SmartLabs. SmartLabs provides this INSTEON Modem Developer's Guide on an "AS IS" basis.

SMARTLABS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, REGARDING THIS INSTEON MODEM DEVELOPER'S GUIDE OR ITS USE, ALONE OR IN COMBINATION WITH ANY PRODUCT.

IN NO EVENT SHALL SMARTLABS BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) ARISING IN ANY WAY OUT OF THE USE, REPRODUCTION, MODIFICATION AND/OR DISTRIBUTION OF THIS INSTEON MODEM DEVELOPER'S GUIDE, HOWEVER CAUSED AND WHETHER UNDER THEORY OF CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT LIABILITY OR OTHERWISE, EVEN IF SMARTLABS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Trademarks and Patents

SmartLabs, Smarthome, INSTEON, Dual Mesh, BiPHY, ALL-Link, PowerLinc, ControLinc, LampLinc, SwitchLinc, RemoteLinc, Electronic Home Improvement, SmartLabs Device Manager, Home Network Language, and Plug-n-Tap are trademarks of SmartLabs, Inc.

INSTEON networking technology is covered by pending U.S. and foreign patents.

Copyright

INTRODUCTION

This *INSTEON Modem Developer's Guide* is for users of INSTEON Modem chips, such as the IN2680A Powerline Modem Interface or the IN2682A RF Modem Interface, and also for purchasers of the SmartLabs PowerLinc™ Modem (PLM) module.

The information in this document is excepted from the *INSTEON Developer's Guide*[2], which purchasers of an INSTEON Software Development Kit may download from http://code.insteon.net.

INSTEON Modem Chips

INSTEON Modems (IMs) are single chips available from SmartLabs that use simple ASCII commands over a serial port to interface to an INSTEON network. The *IN2680A INSTEON Direct Powerline Modem Interface*[3] chip connects to an INSTEON network via the house wiring and the *IN2682A INSTEON Direct RF Modem Interface*[3] connects via radio. A BiPHY™ Modem that interfaces to *both* the powerline and radio is under development.

Developers can create INSTEON applications that run on whatever host device they choose, as long as the host can communicate serially with the IM using the RS232 serial protocol at TTL levels. A microcontroller chip is the most common choice for a host device in standalone INSTEON modules, although virtually any hardware capable of executing applications and communicating serially can use an IM to interface with an INSTEON network.

Perhaps the greatest advantage of using an IM is that you can create applications in a development environment that you are already comfortable with. The ASCII *IM Serial Commands*[9] are relatively few in number and easy to learn, so development cycles can be short.

As an added bonus, the easiest way to achieve INSTEON conformance for your product is to build it around an INSTEON modem, because an IM automatically handles most of the details of the INSTEON protocol for you. See the *INSTEON Conformance Specification*[2] document for the full conformance requirements.

The main functions of an INSTEON Modem are:

- Interfacing to a host via an RS232 serial port at TTL levels.
- Interfacing to the powerline or an FSK 915 MHz radio.
- Sending and receiving INSTEON messages.
- Sending and receiving X10 messages.
- ALL-Linking to other INSTEON devices and managing an ALL-Link Database.
- Sending ALL-Link Commands and transparently handling ALL-Link Cleanups.
- Managing a SET Button and LED.

The SmartLabs PowerLinc Modem (PLM)

The SmartLabs PowerLinc Modem (PLM) is an INSTEON-to-Serial Bridge module that plugs into a power outlet and also has a serial port that you connect to your PC (an Ethernet interface is under development). It uses an IN2680A Powerline Modem chip that offers a simple set of ASCII *IM Serial Commands*[9] for interacting with INSTEON devices.

The PLM uses a daughter board to implement serial communications with the host. Daughter boards interface to the PLM's main board via an 8-pin connector using TTL-level serial communications. PLMs with RS232 daughter boards are currently available, with USB and Ethernet versions under development.

You may communicate to an RS232 PLM via USB by using a USB-to-Serial adapter. SmartLabs has found that Keyspan brand adapters, models USA-49WLC and USA-19HS, provide excellent protocol translation and PLM compatibility.

If you wish, you may create a custom daughter board that fits within a PLM module. You can find hardware reference designs for such custom devices in the *Hardware Reference*[51] section below. To support custom daughter boards, SmartLabs offers a special version of the PLM with the following features:

- Uses the same case as the current PLM/PLC modules.

- Has no labeling on the front cover or rear UL label.

- Does not have UL approval.

- Does not include a daughter board.

- Includes the plastic insert for a RJ-45 jack or a blank cover.

- Uses PLM firmware with auto EEPROM detection. When no external EEPROM is detected, the PLM is limited to 31 ALL-Links.

Comparing the PowerLinc Modem (PLM) to the PowerLinc Controller (PLC)

The SmartLabs PowerLinc controller (PLC) module runs a downloadable SALad application that implements an interface between a host device and an INSTEON network on the powerline. SALad is a language developed specially for the PLC and other SALad-enabled INSTEON devices. If desired, the SALad application can be designed to run in standalone mode without any connection to a host.

Unlike the PLC, a PLM cannot operate in standalone mode because it cannot run application programs by itself. External applications designed to work with a PLC, such as SmartLabs Device Manager (SDM), will not work with a PLM.

In summary, these are the main differences between the PLC and the PLM:

- The PLM has a simplified command set compared to the PLC.

- The PLM does not support SmartLabs Device Manager (SDM) running on a host computer.

- The PLC runs a downloadable SALad application, such as the SALad *coreApp* program, but the PLM cannot run applications of any kind. An embedded host on a daughter card or else an always-on external host must be available full time to run applications and manage the PLM.

- The PLM does not have an internal realtime clock.

- If fewer than 32 ALL-Links need to be supported, the PLM can run without external EEPROM. The PLC must have external EEPROM to store a downloadable SALad program.

Other Documents Included by Reference

This *INSTEON Modem Developer's Guide* contains information abstracted from the comprehensive *INSTEON Developer's Guide, 2nd Edition*.

Although the full *INSTEON Developer's Guide* is largely self-contained, there are aspects of INSTEON technology, such as listings of INSTEON Commands, INSTEON Device Categories, and INSTEON Product Keys, that require continuous updating as developers create new INSTEON products. Accordingly, SmartLabs maintains separate documents for that kind of information.

All of the documents listed in this section are available for downloading at http://code.insteon.net.

INSTEON Developer's Guide

The book-length *INSTEON Developer's Guide, 2nd Edition* is the primary source for the information contained in this (much shorter) *INSTEON Modem Developer's Guide*. Some links in this document refer to information found there. Developers who purchase an INSTEON Software Developer's Kit may download the *INSTEON Developer's Guide* from http://code.insteon.net.

INSTEON Conformance Specification

The *INSTEON Conformance Specification* identifies those aspects of INSTEON that assure interoperability with other INSTEON products. The Conformance Spec assumes that readers have already gained familiarity with INSTEON technology by reading the *INSTEON Developer's Guide*.

INSTEON Command Tables Document

The current tables of INSTEON Commands are contained in a separate document titled *INSTEON Command Tables*, which is integral to both the *INSTEON Conformance Specification* and the *INSTEON Developer's Guide*.

The filename for that document is *INSTEON Command Tables yyyymmddx.doc*, where *yyyy* is the year, *mm* is the month, *dd* is the day, and *x* is a daily version letter beginning with *a*. Be sure to refer to the document with the latest date.

INSTEON Device Categories and Product Keys Document

The current table of INSTEON Device Categories (DevCats), Subcategories (SubCats), and INSTEON Product Keys (IPKs) is contained in a separate document titled *INSTEON Device Categories and Product Keys*, which is also integral to both the *INSTEON Conformance Specification* and the *INSTEON Developer's Guide*.

The filename for that document is *INSTEON DevCats and Product Keys yyyymmddx.doc*, where *yyyy* is the year, *mm* is the month, *dd* is the day, and *x* is a daily version letter beginning with *a*. Be sure to refer to the document with the latest date.

INSTEON Modem Spec Sheets

Developers will find the latest specifications for INSTEON modem ICs at
www.insteon.net.

IN2680A INSTEON Direct Powerline Modem Interface

The IN2680A is a one-chip solution that uses the simple ASCII serial interface
documented here (see *IM Serial Commands*[9]) to connect a host device or system to
an INSTEON network via the powerline.

IN2682A INSTEON Direct RF Modem Interface

The IN2682A is similar to the IN2680A Powerline Modem except that it connects to
an INSTEON network via radio.

INSTEON MODEM REFERENCE

Software Reference

INSTEON Modem (IM) chips and the SmartLabs PowerLinc™ Modem (PLM) module offer developers a simple, robust interface to an INSTEON network. There are currently two kinds of IM chip, the _IN2680A INSTEON Direct Powerline Modem Interface_[3] and the _IN2682A INSTEON Direct RF Modem Interface_[3]. A BiPHY™ Modem that interfaces to _both_ the powerline and radio is under development.

INSTEON Modems provide a simpler interface to many of the low-level _IBIOS Serial Commands_ implemented in the SmartLabs PowerLinc Controller™ (PLC) described in the _INSTEON Developer's Guide_[2], but they also handle ALL-Linking, ALL-Link Database management, ALL-Link Cleanup messages, X10 powerline interfacing, and message acknowledgement. The RS232 serial interface to the host is similar to that of the PLC.

In This Section

IM Serial Communication Protocol and Settings

In This Section

IM Serial Communication Protocol

All INSTEON Modem (IM) Serial Commands start with ASCII `0x02` (STX, Start-of-Text) followed by the Serial Command Number (see *IM Serial Commands*9). What data follows the Command depends on the Command syntax (see *IM Serial Command Summary Table*10 and *IM Serial Command Charts*14).

When you send a message to the IM, it will respond with an echo of the `0x02` and the IM Command Number followed by any data that the Command returns (often just an echo of what you sent to it). The last byte it sends back will be ASCII 0x06 (ACK, Acknowledge).

(**S:** and **R:** denote serial data you **Send to** or **Receive from** the IM, respectively.)

S:	0x02 <Command Number> <parameters>
R:	0x02 <Command Number> <any returned data> 0x06 (ACK)

If the IM is not ready, it will respond with an echo of the `0x02` and the IM Command Number followed by ASCII 0x15 (NAK, Negative Acknowledge).

S:	0x02 <Command Number> <parameters>
R:	0x15 (NAK)

If you receive 0x15 (NAK), resend your Serial Command.

IM RS232 Port Settings

To communicate to an RS232 IM, set your PC's COM port as follows:

Setting	Value
Baud Rate	19,200
Data Bits	8
Parity	N
Stop Bits	1
Hardware Flow Control	None
Software Flow Control	IM echoes bytes received from host

The IM buffers IM Commands as it receives them, so you can send a complete IM Command without pause. To maintain compatibility with earlier IM versions, the IM will echo each byte that it receives (earlier versions of the IM used byte echoing for flow control). You can now ignore the byte echos, but in order to avoid overrunning the IM's receive buffer, you must wait for the IM to send its response to your current IM Command before sending a new one.

Note that there is a *maximum* time between IM Command bytes that you send to the IM. If you do not send the next expected byte of an IM Command within 240 milliseconds after sending the previous one, the IM will reset its message parser and you will have to resend the message from the beginning. You can disable this *Deadman* feature by setting a configuration bit (see *Set IM Configuration*43 below).

There is no flow control when the IM sends data *to* the host—the IM will transfer data to the host as fast as it can send it.

How to Quickly Start Communicating with an IM

No matter how your application intends to use the IM, it is important to gain a basic understanding of how it operates. SmartLabs suggests that developers use a terminal communications program and a serial connection to an IM to get started.

While there are many terminal programs for computers, SmartLabs has found good results with Docklight Scripting. An evaluation copy may be downloaded from http://www.docklight.de/.

Docklight Scripting allows you to set up test macros and label received *IM Serial Commands*[9] for easy identification, as suggested in the following screenshot:
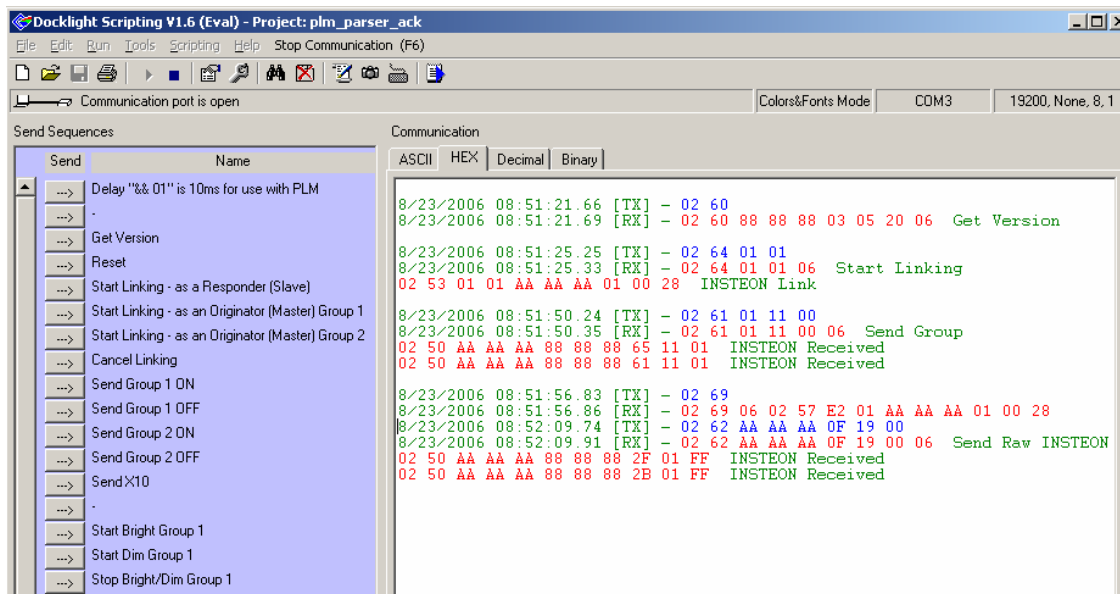
For a step-by-step tutorial on using Docklight Scripting with a PowerLinc Modem (PLM) to control a LampLinc dimmer supplied in a PLM Developer Kit, see the section *POWERLINC MODEM (PLM) QUICK START GUIDE*[62] below.

IM Power-up and Reset States

This section describes the *IM Power-up Behavior*[8] and the *IM Factory Reset State*[8].

IM Power-up Behavior

The table below shows the state of the IM when it powers up. Holding down the SET Button while powering up will cause a factory reset.

LED Indication	Meaning
LED on steadily	The IM detected an external EEPROM (up to 32 KB) for storage of database links.
LED blinks six times	The IM did not detect an external EEPROM, so it will use the internal EEPROM in the processor chip. A maximum of 31 ALL-Links are permitted. An attempt to add a 32nd ALL-Link will result in the 31st being erased.
LED off	The user pressed and held the IM's SET button for 10 seconds while powering up, causing the IM to perform a factory reset and go into the *IM Factory Reset State*[8]. At the conclusion of the reset, the IM's LED will give one of the two indications above. You will also receive a *User Reset Detected*[41] message from the IM.

IM Factory Reset State

Resetting the IM to its factory default condition by holding down the SET Button for ten seconds while powering it up or by sending it a *Reset the IM*[40] Command puts it into the following state:

IM Resource	Factory Reset State
ALL-Link Database	Erased (set to all zeros).
Host Device Category, Device Subcategory, Firmware Version	Set to the original DevCat (0x03), SubCat (0x05), and firmware version hard-coded into the IM's firmware at the factory.
IM Configuration Flags	Cleared (set to all zeros).

IM Serial Commands

The IM Serial Command set is a simple but complete interface between a host application and an INSTEON network. For example, a microcontroller in a thermostat could use an INSTEON Powerline Modem chip to send and receive messages to other INSTEON or X10 devices on the home's powerline.

In this section, the IM Serial Commands are presented twice, once as a summary table, and again as a series of charts grouped by functionality.

In This Section

Describes all of the IM Serial Commands in table form ordered by Command Number.

Describes all of the IM Serial Commands using individual charts for each Command, grouped by functionality.

IM Serial Command Summary Table

This table lists all of the Modem Serial Commands supported by INSTEON powerline or RF modem chips.

Code

Gives the hexadecimal number of the IM Serial Command. Note that IM Commands sent by an IM to the host begin at **0x50** and IM Commands sent by the host to an IM begin at **0x60**.

Command

Gives the name of the IM Serial Command as a link to the complete explanation of the Command in the *IM Serial Command Charts*[14].

Format

Gives the syntax of the IM Serial Command, including any parameters.

S: and **R**: denote serial data you **Send to** or **Receive from** the IM, respectively. See *IM Serial Communication Protocol*[6] for more information.

All IM Serial Commands start with ASCII 0x02 (STX, Start-of-Text) followed by the Serial Command Number.

All fields in this table contain only one byte, except as noted.

INSTEON Modem Serial Commands		
Commands Sent from an IM to the Host		
Code	**Command**	**Format**
0x50	INSTEON Standard Message Received[18]	**R:** 0x02 **0x50** <INSTEON Standard message (9 bytes)>
0x51	INSTEON Extended Message Received[19]	**R:** 0x02 **0x51** <INSTEON Extended message (23 bytes)>
0x52	X10 Received[25]	**R:** 0x02 **0x52** <Raw X10> <X10 Flag>
0x53	ALL-Linking Completed[32]	**R:** 0x02 **0x53** <0x00 (IM is Responder) | 0x01 (IM is Controller | 0xFF Link Deleted)> <ALL-Link Group> <ID high byte> <ID middle byte> <ID low byte> <Device Category> <Device Subcategory> <0xFF | Firmware Revision>
0x54	Button Event Report[48]	**R:** 0x02 **0x54** <0x02> IM's SET Button tapped
		R: 0x02 **0x54** <0x03> IM's SET Button held
		R: 0x02 **0x54** <0x04> IM's SET Button released after hold
		R: 0x02 **0x54** <0x12> IM's Button 2 tapped
		R: 0x02 **0x54** <0x13> IM's Button 2 held
		R: 0x02 **0x54** <0x14> IM's Button 2 released after hold
		R: 0x02 **0x54** <0x22> IM's Button 3 tapped

	INSTEON Modem Serial Commands	
	Commands Sent from an IM to the Host	
Code	**Command**	**Format**
		R: 0x02 **0x54** <0x23> IM's Button 3 held
		R: 0x02 **0x54** <0x24> IM's Button 3 released after hold
0x55	User Reset Detected[41]	**R:** 0x02 **0x55** User pushed and held IM's SET Button on power up
0x56	ALL-Link Cleanup Failure Report[28]	**R:** 0x02 **0x56** <0x01> <ALL-Link Group> <ID high byte> <ID middle byte> <ID low byte>
0x57	ALL-Link Record Response[36]	**R:** 0x02 **0x57** <ALL-Link Record Flags> <ALL-Link Group> <ID high byte> <ID middle byte> <ID low byte> <Link Data 1> <Link Data 2> <Link Data 3>
0x58	ALL-Link Cleanup Status Report[29]	**R:** 0x02 **0x58** <0x06> ALL-Link Cleanup sequence completed
		R: 0x02 **0x58** <0x15> ALL-Link Cleanup sequence aborted due to INSTEON traffic
	Commands Sent from the Host to an IM	
0x60	Get IM Info[45]	**S:** 0x02 **0x60**
		R: 0x02 **0x60** <ID high byte> <ID middle byte> <ID low byte> <Device Category> <Device Subcategory> < Firmware Revision> <0x06>
0x61	Send ALL-Link Command[26]	**S:** 0x02 **0x61** <ALL-Link Group> <ALL-Link Command> <0xFF | 0x00>
		R: 0x02 **0x61** <ALL-Link Group> <ALL-Link Command> <0xFF | 0x00> <0x06>
0x62	Send INSTEON Standard or Extended Message[15]	**S:** 0x02 **0x62** <INSTEON Standard message (6 bytes, excludes *From Address*) | INSTEON Extended message (20 bytes, excludes *From Address*)>
		R: 0x02 **0x62** <INSTEON Standard message (6 bytes, excludes *From Address*) | INSTEON Extended message (20 bytes, excludes *From Address*)> <0x06>
0x63	Send X10[24]	**S:** 0x02 **0x63** <Raw X10> <X10 Flag>
		R: 0x02 **0x63** <Raw X10> <X10 Flag> <0x06>
0x64	Start ALL-Linking[30]	**S:** 0x02 **0x64** <0x00 (IM is Responder) | 0x01 (IM is Controller) | 0x03 (IM is either) | 0xFF (Link Deleted)> <ALL-Link Group>

	INSTEON Modem Serial Commands	
	Commands Sent from an IM to the Host	
Code	**Command**	**Format**
		R: 0x02 **0x64** <0x00 (IM is Responder) | 0x01 (IM is Controller) | 0x03 (IM is either) | 0xFF (Link Deleted)> <ALL-Link Group> <0x06>
0x65	Cancel ALL-Linking[31]	**S:** 0x02 **0x65**
		R: 0x02 **0x65** <0x06>
0x66	Set Host Device Category[46]	**S:** 0x02 **0x66** <Device Category> <Device Subcategory> <0xFF | Firmware Revision>
		R: 0x02 **0x66** <Device Category> <Device Subcategory> <0xFF | Firmware Revision> <0x06>
0x67	Reset the IM[40]	**S:** 0x02 **0x67**
		R: 0x02 **0x67** <0x06>
0x68	Set INSTEON ACK Message Byte[21]	**S:** 0x02 **0x68** <Command 2 Data>
		R: 0x02 **0x68** <Command 2 Data> <0x06>
0x69	Get First ALL-Link Record[33]	**S:** 0x02 **0x69**
		R: 0x02 **0x69** <0x06>
0x6A	Get Next ALL-Link Record[34]	**S:** 0x02 **0x6A**
		R: 0x02 **0x6A** <0x06>
0x6B	Set IM Configuration[43]	**S:** 0x02 **0x6B** <IM Configuration Flags>
		R: 0x02 **0x6B** <IM Configuration Flags> <0x06>
0x6C	Get ALL-Link Record for Sender[35]	**S:** 0x02 **0x6C**
		R: 0x02 **0x6C** <0x06>
0x6D	LED On[49]	**S:** 0x02 **0x6D**
		R: 0x02 **0x6D** <0x06>
0x6E	LED Off[50]	**S:** 0x02 **0x6E**
		R: 0x02 **0x6E** <0x06>
0x6F	Manage ALL-Link Record[37]	**S:** 0x02 **0x6F** <Control Flags> <ALL-Link Record Flags> <ALL-Link Group> <ID high byte> <ID middle byte> <ID low byte> <Link Data 1> <Link Data 2> <Link Data 3>

	INSTEON Modem Serial Commands	
	Commands Sent from an IM to the Host	
Code	**Command**	**Format**
		R: 0x02 **0x6F** <Control Flags> <ALL-Link Record Flags> <ALL-Link Group> <ID high byte> <ID middle byte> <ID low byte> <Link Data 1> <Link Data 2> <Link Data 3> <0x06>
0x70	Set INSTEON NAK Message Byte[23]	**S:** 0x02 **0x70** <Command 2 Data>
		R: 0x02 **0x70** <Command 2 Data> <0x06>
0x71	Set INSTEON ACK Message Two Bytes[22]	**S:** 0x02 **0x71** <Command 1 Data> <Command 2 Data>
		R: 0x02 **0x71** <Command 1 Data> <Command 2 Data> <0x06>
0x72	RF Sleep[47]	**S:** 0x02 **0x72**
		R: 0x02 **0x72** <0x06>
0x73	Get IM Configuration[42]	**S:** 0x02 **0x73**
		R: 0x02 **0x73** <IM Configuration Flags> <Spare 1> <Spare 2> <0x06>

IM Serial Command Charts

The following charts describe the IM Commands individually in a chart format, grouped by functionality. These are the same IM Commands as in the *IM Serial Command Summary Table*[10], which is ordered by Command Number.

Note that IM Commands sent by an IM to the host begin at **0x50** and IM Commands sent by the host to an IM begin at **0x60**. When the host sends an IM Command to an IM, the IM will respond with a message according to the *IM Serial Communication Protocol*[6].

In This Section

INSTEON Message Handling

Send INSTEON Standard or Extended Message

This Command lets you send either a Standard-length or an Extended-length INSTEON message, depending only on what kind of INSTEON message you include in the body of the Command.

Send INSTEON Standard-length Message

Send INSTEON Standard-length Message (0x62)	
What it does	Allows you to send a raw Standard-length INSTEON message.
What you send	8 bytes.
What you'll get	9 bytes.
LED indication	None.
Related Commands	IM 0x50 INSTEON Standard Message Received[18] IM 0x51 INSTEON Extended Message Received[19]

Command Sent from Host to IM		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x62**	IM Command Number
3	<To Address high>	The high byte of the INSTEON ID of the message addressee.
4	<To Address middle>	The middle byte of the INSTEON ID of the message addressee.
5	<To Address low>	The low byte of the INSTEON ID of the message addressee.
6	<Message Flags>	The INSTEON message flags indicating message type and hops. Extended Message Flag (bit 4) is 0
7	<Command 1>	INSTEON Command 1 for the addressee to execute
8	<Command 2>	INSTEON Command 2 for the addressee to execute

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x62**	Echoed IM Command Number
3	<To Address high>	Echoed <To Address high>
4	<To Address middle>	Echoed <To Address middle>
5	<To Address low>	Echoed <To Address low>
6	<Message Flags>	Echoed <Message Flags> Extended Message Flag (bit 4) is 0
7	<Command 1>	Echoed <Command 1>
8	<Command 2>	Echoed <Command 2>
9	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly 0x15 (NAK) if an error occurred

Notes
The *From Address* is not required because the IM will automatically insert its own INSTEON ID into the message.
For more information on INSTEON Commands and the latest Command set, please download the current INSTEON Command Tables Document[2] from http://code.insteon.net.

Send INSTEON Extended-length Message

Send INSTEON Extended-length Message (0x62)	
What it does	Allows you to send a raw Extended-length INSTEON message.
What you send	22 bytes.
What you'll get	23 bytes.
LED indication	None.
Related Commands	IM 0x50 INSTEON Standard Message Received[18] IM 0x51 INSTEON Extended Message Received[19]

Command Sent from Host to IM		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x62**	IM Command Number
3	\<To Address high\>	The high byte of the INSTEON ID of the message addressee.
4	\<To Address middle\>	The middle byte of the INSTEON ID of the message addressee.
5	\<To Address low\>	The low byte of the INSTEON ID of the message addressee.
6	\<Message Flags\>	The INSTEON message flags indicating message type and hops. Extended Message Flag (bit 4) is 1
7	\<Command 1\>	INSTEON Command 1 for the addressee to execute
8	\<Command 2\>	INSTEON Command 2 for the addressee to execute
9	\<User Data 1\>	Extended message data
10	\<User Data 2\>	Extended message data
11	\<User Data 3\>	Extended message data
12	\<User Data 4\>	Extended message data
13	\<User Data 5\>	Extended message data
14	\<User Data 6\>	Extended message data
15	\<User Data 7\>	Extended message data
16	\<User Data 8\>	Extended message data
17	\<User Data 9\>	Extended message data
18	\<User Data 10\>	Extended message data
19	\<User Data 11\>	Extended message data
20	\<User Data 12\>	Extended message data
21	\<User Data 13\>	Extended message data
22	\<User Data 14\>	Extended message data

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x62**	Echoed IM Command Number
3	\<To Address high\>	Echoed \<To Address high\>
4	\<To Address middle\>	Echoed \<To Address middle\>
5	\<To Address low\>	Echoed \<To Address low\>
6	\<Message Flags\>	Echoed \<Message Flags\> Extended Message Flag (bit 4) is 1
7	\<Command 1\>	Echoed \<Command 1\>
8	\<Command 2\>	Echoed \<Command 2\>
9	\<User Data 1\>	Echoed Extended message data
10	\<User Data 2\>	Echoed Extended message data
11	\<User Data 3\>	Echoed Extended message data
12	\<User Data 4\>	Echoed Extended message data
13	\<User Data 5\>	Echoed Extended message data
14	\<User Data 6\>	Echoed Extended message data

Send INSTEON Extended-length Message (0x62)		
15	<User Data 7>	Echoed Extended message data
16	<User Data 8>	Echoed Extended message data
17	<User Data 9>	Echoed Extended message data
18	<User Data 10>	Echoed Extended message data
19	<User Data 11>	Echoed Extended message data
20	<User Data 12>	Echoed Extended message data
21	<User Data 13>	Echoed Extended message data
22	<User Data 14>	Echoed Extended message data
23	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly 0x15 (NAK) if an error occurred
Notes		
The *From Address* is not required because the IM will automatically insert its own INSTEON ID into the message.		
For more information on INSTEON Commands and the latest Command set, please download the current *INSTEON Command Tables Document* from http://code.insteon.net.		

INSTEON Standard Message Received

INSTEON Standard Message Received (0x50)	
What it does	Informs you of an incoming Standard-length INSTEON message.
When you'll get this	A Standard-length INSTEON message is received from either a Controller or Responder that you are ALL-Linked to.
What you'll get	11 bytes.
LED indication	The LED will blink during INSTEON reception.
Related Commands	IM 0x51 INSTEON Extended Message Received[19] IM 0x52 X10 Received[25]

Message Sent from IM to Host		
Byte	Value	Meaning
1	0x02	Start of IM Command
2	**0x50**	IM Command Number
3	<From Address high>	The high byte of the INSTEON ID of the message originator.
4	<From Address middle>	The middle byte of the INSTEON ID of the message originator.
5	<From Address low>	The low byte of the INSTEON ID of the message originator.
6	<To Address high>	The high byte of the INSTEON ID of the message addressee. If the message is an ALL-Link Broadcast (bits 7 and 6 of the <Message Flags> byte are set) then this will be 0.
7	<To Address middle>	The middle byte of the INSTEON ID of the message addressee. If the message is an ALL-Link Broadcast (bits 7 and 6 of the <Message Flags> byte are set) then this will be 0.
8	<To Address low>	The low byte of the INSTEON ID of the message addressee. If the message is an ALL-Link Broadcast (bits 7 and 6 of the <Message Flags> byte are set) then this will indicate the ALL-Link Group Number.
9	<Message Flags>	The INSTEON message flags indicating message type and hops.
10	<Command 1>	INSTEON *Command 1* field of the message.
11	<Command 2>	INSTEON *Command 2* field of the message. This byte contains the ALL-Link Group Number of the ALL-Link Broadcast when either bit 6 of the <Message Flags> byte is set (ALL-Link Cleanup) or bits 6 and 5 of the <Message Flags> byte are set (ALL-Link Cleanup ACK).

Notes
This is the same as IM 0x51 *INSTEON Extended Message Received*[19], except that there is no <User Data>.
Normally, the IM will only send the host INSTEON messages that are explicitly addressed to the IM or that are from devices that the IM is ALL-Linked to. This behavior can be modified—see the *About Monitor Mode*[44] note in the *Set IM Configuration*[43] chart for more information.
For more information on INSTEON Commands and the latest Command set, please download the current *INSTEON Command Tables Document*[2] from http://code.insteon.net.

INSTEON Extended Message Received

INSTEON Extended Message Received (0x51)	
What it does	Informs you of an incoming Extended-length INSTEON message.
When you'll get this	An Extended-length INSTEON message is received from either a Controller or Responder that you are ALL-Linked to.
What you'll get	25 bytes.
LED indication	The LED will blink during INSTEON reception.
Related Commands	IM 0x50 INSTEON Standard Message Received[18] IM 0x52 X10 Received[25]

Message Sent from IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x51**	IM Command Number
3	<From Address high>	The high byte of the INSTEON ID of the message originator.
4	<From Address middle>	The middle byte of the INSTEON ID of the message originator.
5	<From Address low>	The low byte of the INSTEON ID of the message originator.
6	<To Address high>	The high byte of the INSTEON ID of the message addressee. If the message is an ALL-Link Broadcast (bits 7 and 6 of the <Message Flags> byte are set) then this will be 0.
7	<To Address middle>	The middle byte of the INSTEON ID of the message addressee. If the message is an ALL-Link Broadcast (bits 7 and 6 of the <Message Flags> byte are set) then this will be 0.
8	<To Address low>	The low byte of the INSTEON ID of the message addressee. If the message is an ALL-Link Broadcast (bits 7 and 6 of the <Message Flags> byte are set) then this will indicate the ALL-Link Group Number.
9	<Message Flags>	The INSTEON message flags indicating message type and hops.
10	<Command 1>	INSTEON *Command 1* field of the message.
11	<Command 2>	INSTEON *Command 2* field of the message. This byte contains the ALL-Link Group Number of the ALL-Link Broadcast when either bit 6 of the <Message Flags> byte is set (ALL-Link Cleanup) or bits 6 and 5 of the <Message Flags> byte are set (ALL-Link Cleanup ACK).
12	<User Data 1>	Extended message data
13	<User Data 2>	Extended message data
14	<User Data 3>	Extended message data
15	<User Data 4>	Extended message data
16	<User Data 5>	Extended message data
17	<User Data 6>	Extended message data
18	<User Data 7>	Extended message data
19	<User Data 8>	Extended message data
20	<User Data 9>	Extended message data
21	<User Data 10>	Extended message data
22	<User Data 11>	Extended message data
23	<User Data 12>	Extended message data
24	<User Data 13>	Extended message data
25	<User Data 14>	Extended message data

INSTEON Extended Message Received (0x51)
Notes
This is the same as IM 0x50 <u>INSTEON Standard Message Received</u>[18], except that there are 14 bytes of <User Data>.
Normally, the IM will only send the host INSTEON messages that are explicitly addressed to the IM or that are from devices that the IM is ALL-Linked to. This behavior can be modified—see the *About Monitor Mode*[44] note in the *Set IM Configuration*[43] chart for more information.
For more information on INSTEON Commands and the latest Command set, please download the current *INSTEON Command Tables Document*[2] from <u>http://code.insteon.net</u>.

Set INSTEON ACK Message Byte

Set INSTEON ACK Message Byte (0x68)	
What it does	Allows you to put one byte of data into the *Command 2* field of the INSTEON ACK message that the INSTEON Engine automatically sends after it receives an INSTEON Direct message.
What you send	3 bytes.
What you'll get	4 bytes.
LED indication	None.
Related Commands	IM 0x50 INSTEON Standard Message Received[18] IM 0x51 INSTEON Extended Message Received[19] IM 0x71 Set INSTEON ACK Message Two Bytes[22] IM 0x70 Set INSTEON NAK Message Byte[23]

Command Sent from Host to IM		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x68**	IM Command Number
3	<Command 2 Data>	Data byte to place into the *Command 2* field of the ACK response.

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x68**	Echoed IM Command Number
3	<Command 2 Data>	Echoed <Command 2 Data>
4	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly. 0x15 (NAK) if an error occurred.

Notes
You have only about 15 milliseconds after the receipt of an INSTEON Direct message from the IM to send this Command to the IM. The reason is that the INSTEON Engine in the IM automatically sends Acknowledgement messages in assigned timeslots.
Use *Set INSTEON ACK Message Two Bytes*[22] when you need to return two bytes of data in an ACK message.
Use *Set INSTEON NAK Message Byte*[23] when you need to return one byte of data in a NAK message.
Certain INSTEON Direct Commands require returned data in the Acknowledgement message. For more information on INSTEON Commands and the latest Command set, please download the current *INSTEON Command Tables Document*[2] from http://code.insteon.net.

Set INSTEON ACK Message Two Bytes

Set INSTEON ACK Message Two Bytes (0x71)	
What it does	Allows you to put two bytes of data into the combined *Command 1* and *Command 2* fields of the INSTEON ACK message that the INSTEON Engine automatically sends after it receives an INSTEON Direct message.
What you send	4 bytes.
What you'll get	5 bytes.
LED indication	None.
Related Commands	IM 0x50 INSTEON Standard Message Received[18] IM 0x51 INSTEON Extended Message Received[19] IM 0x68 Set INSTEON ACK Message Byte[21] IM 0x70 Set INSTEON NAK Message Byte[23]

Command Sent from Host to IM		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x71**	IM Command Number
3	<Command 1 Data>	Data byte to place into the *Command 1* field 2 of the ACK response.
4	<Command 2 Data>	Data byte to place into the *Command 2* field 2 of the ACK response.

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x71**	Echoed IM Command Number
3	<Command 1 Data>	Echoed <Command 1 Data>
4	<Command 2 Data>	Echoed <Command 2 Data>
5	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly. 0x15 (NAK) if an error occurred.

Notes
You have only about 15 milliseconds after the receipt of an INSTEON Direct message from the IM to send this Command to the IM. The reason is that the INSTEON Engine in the IM automatically sends Acknowledgement messages in assigned timeslots.
Use *Set INSTEON ACK Message Byte*[21] when you only need to return one byte of data in an ACK message.
Use *Set INSTEON NAK Message Byte*[23] when you need to return one byte of data in a NAK message.
Certain INSTEON Direct Commands require returned data in the Acknowledgement message. For more information on INSTEON Commands and the latest Command set, please download the current *INSTEON Command Tables Document*[2] from http://code.insteon.net.

Set INSTEON NAK Message Byte

Set INSTEON NAK Message Byte (0x70)	
What it does	Allows you to change the INSTEON ACK message that the INSTEON Engine automatically sends after it receives an INSTEON Direct message into a NAK message, and to put one byte of data into the *Command 2* field of that message.
What you send	3 bytes.
What you'll get	4 bytes.
LED indication	None.
Related Commands	IM 0x50 INSTEON Standard Message Received[18] IM 0x51 INSTEON Extended Message Received[19] IM 0x68 Set INSTEON ACK Message Byte[21] IM 0x70 Set INSTEON ACK Message Two Bytes[22]

Command Sent from Host to IM		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x70**	IM Command Number
3	<Command 2 Data>	Data byte to place into the *Command 2* field of the ACK response.

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x70**	Echoed IM Command Number
3	<Command 2 Data>	Echoed <Command 2 Data>
4	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly. 0x15 (NAK) if an error occurred.

Notes
You have only about 15 milliseconds after the receipt of an INSTEON Direct message from the IM to send this Command to the IM. The reason is that the INSTEON Engine in the IM automatically sends Acknowledgement messages in assigned timeslots.
Use *Set INSTEON ACK Message Byte*[21] or *Set INSTEON ACK Message Two Bytes*[22] when you need to return one or two bytes of data in an ACK message.
NAK messages report certain error conditions in a receiving device. See *NAK Error Codes* in the *INSTEON Developer's Guide*[2] for more information.

X10 Message Handling

Send X10

Send X10 (0x63)	
What it does	Allows you to send a raw X10 Address or X10 Command.
What you send	4 bytes.
What you'll get	5 bytes.
LED indication	None.
Related Commands	IM 0x52 X10 Received₂₅

Command Sent from Host to IM		
Byte	Value	Meaning
1	0x02	Start of IM Command
2	**0x63**	IM Command Number
3	<Raw X10>	The four most significant bits contain the X10 House Code. The four least significant bits contain the X10 Key Code.
4	<X10 Flag>	0x00 indicates that the X10 Key Code is an X10 Unit Code. 0x80 indicates that the X10 Key Code is an X10 Command.

Message Returned by IM to Host		
Byte	Value	Meaning
1	0x02	Echoed Start of IM Command
2	**0x63**	Echoed IM Command Number
3	<Raw X10>	Echoed <Raw X10>
4	<X10 Flag>	Echoed <X10 Flag>
5	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly 0x15 (NAK) if an error occurred

X10 Translation Table		
	4 MSBs of <Raw X10>	4 LSBs of <Raw X10>	
4-bit Code	X10 House Code	X10 Unit Code <X10 Flag> = 0x00	X10 Command <X10 Flag> = 0x80
0x6	A	1	All Lights Off
0xE	B	2	Status = Off
0x2	C	3	On
0xA	D	4	Preset Dim
0x1	E	5	All Lights On
0x9	F	6	Hail Acknowledge
0x5	G	7	Bright
0xD	H	8	Status = On
0x7	I	9	Extended Code
0xF	J	10	Status Request
0x3	K	11	Off
0xB	L	12	Preset Dim
0x0	M	13	All Units Off
0x8	N	14	Hail Request
0x4	O	15	Dim
0xC	P	16	Extended Data (analog)

X10 Received

<table>
<tr><td colspan="2" align="center">X10 Received (0x52)</td></tr>
<tr><td>What it does</td><td>Informs you of an X10 byte detected on the powerline.</td></tr>
<tr><td>When you'll get this</td><td>Any X10 traffic is detected on the powerline.</td></tr>
<tr><td>What you'll get</td><td>4 bytes.</td></tr>
<tr><td>LED indication</td><td>The LED will blink during X10 reception.</td></tr>
<tr><td>Related Commands</td><td>IM 0x63 Send X10[24]
IM 0x50 INSTEON Standard Message Received[18]
IM 0x51 INSTEON Extended Message Received[19]</td></tr>
</table>

Message Sent from IM to Host

Byte	Value	Meaning
1	0x02	Start of IM Command
2	**0x52**	IM Command Number
3	<Raw X10>	The four most significant bits contain the X10 House Code. The four least significant bits contain the X10 Key Code.
4	<X10 Flag>	0x00 indicates that the X10 Key Code is an X10 Unit Code. 0x80 indicates that the X10 Key Code is an X10 Command.

X10 Translation Table

	4 MSBs of <Raw X10>	4 LSBs of <Raw X10>	
4-bit Code	X10 House Code	X10 Unit Code <X10 Flag> = 0x00	X10 Command <X10 Flag> = 0x80
0x6	A	1	All Lights Off
0xE	B	2	Status = Off
0x2	C	3	On
0xA	D	4	Preset Dim
0x1	E	5	All Lights On
0x9	F	6	Hail Acknowledge
0x5	G	7	Bright
0xD	H	8	Status = On
0x7	I	9	Extended Code
0xF	J	10	Status Request
0x3	K	11	Off
0xB	L	12	Preset Dim
0x0	M	13	All Units Off
0x8	N	14	Hail Request
0x4	O	15	Dim
0xC	P	16	Extended Data (analog)

INSTE ⏻ N

INSTEON ALL-Link Commands

Send ALL-Link Command

Send ALL-Link Command (0x61)	
What it does	Sends an ALL-Link Command to an ALL-Link Group of one or more Responders that the IM is ALL-Linked to.
What you send	5 bytes.
What you'll get	6 bytes for the echo of the Command and then an additional 11 bytes in an _INSTEON Standard Message Received_18 message for each device in the group that acknowledges ALL-Link Cleanup, or 7 bytes in an _ALL-Link Cleanup Failure Report_28 message for each device in the group that does not acknowledge ALL-Link Cleanup.
LED indication	None.
Related Commands	IM 0x50 _INSTEON Standard Message Received_18 IM 0x56 _ALL-Link Cleanup Failure Report_28 IM 0x58 _ALL-Link Cleanup Status Report_29

Command Sent from Host to IM		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x61**	IM Command Number
3	<ALL-Link Group>	ALL-Link Group Number that the ALL-Link Command is sent to
4	<ALL-Link Command>	ALL-Link Command
5	<Broadcast Command 2>	Sent in the _Command 2_ field of the ALL-Link Broadcast message only. _Command 2_ will always contain the ALL-Link Group Number for the ALL-Link Cleanup messages that follow.

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x61**	Echoed IM Command Number
3	<ALL-Link Group>	Echoed <ALL-Link Group>
4	<ALL-Link Command>	Echoed <ALL-Link Command>
5	<Broadcast Command 2>	Echoed <Broadcast Command 2>
6	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly 0x15 (NAK) if an error occurred or the group does not exist

Notes
The IM automatically sends ALL-Link Cleanup messages to each member of an ALL-Link Group following an ALL-Link Broadcast message. If the IM detects other INSTEON traffic during this process, it will abort the ALL-Link Cleanup sequence and send you an _ALL-Link Cleanup Status Report_29 with a _Status Byte_ of 0x15 (NAK). The Cleanup sequence proceeds in the order in which the devices in the ALL-Link Group were added to the ALL-Link Database. If the IM finishes sending _all_ of the Cleanup messages, it will send you an _ALL-Link Cleanup Status Report_29 with a _Status Byte of_ 0x06 (ACK).
For _each_ ALL-Link Cleanup message that the IM sends, you will either receive an _INSTEON Standard Message Received_18 when the Responder answers with a Cleanup acknowledgement message, or else you will receive an _ALL-Link Cleanup Failure Report_28 if the Responder fails to answer with a Cleanup acknowledgement message. The IM will send you an _ALL-Link Cleanup Status Report_29 whether or not every ALL-Link Group member acknowledges the Cleanup Command that the IM sends to it.
You can cause the IM to cancel its own Cleanup sequence by sending it a new _Send ALL-Link Command_26 or _Send INSTEON Standard or Extended Message_15 during the time that it is sending a Cleanup sequence (i.e. _after_ it has finished sending an ALL-Link Broadcast message). The IM _will_ send you an _ALL-Link Cleanup Status Report_29 in those cases.
The IM first sends an ALL-Link Broadcast message with _Max Hops_ set to 3. When it sends the ensuing ALL-Link Cleanup messages, it sets _Max Hops_ to 1. If the IM's INSTEON Engine needs to retry a Cleanup message, it will automatically increment _Max Hops_ for each retry, up to a maximum of value of 3.

Send ALL-Link Command (0x61)
The IM sends the ALL-Link Broadcast message immediately if there is no other INSTEON traffic. If there is other INSTEON traffic, the IM will wait for one silent powerline zero crossing following a completed INSTEON message. The IM will send the first ALL-Link Cleanup message after a delay of 7 zero crossings. Subsequent Cleanups will go out with a delay of 2 zero crossings.
Do not use this command to control light levels with the *Light Start Manual Change* INSTEON Command **SA 0x17**. Use *Send INSTEON Standard-length Message*[15] to send INSTEON Command **SD 0x17** instead.
For more information on INSTEON Commands and the latest Command set, please download the current *INSTEON Command Tables Document*[2] from http://code.insteon.net.

ALL-Link Cleanup Failure Report

<table>
<tr><td colspan="3" align="center">**ALL-Link Cleanup Failure Report (0x56)**</td></tr>
<tr><td colspan="2">**What it does**</td><td>Reports that an ALL-Link Group member did not acknowledge an ALL-Link Cleanup Command.</td></tr>
<tr><td colspan="2">**When you'll get this**</td><td>An ALL-Link Group member that you are trying to control did not acknowledge the ALL-Link Cleanup Command sent by the IM.</td></tr>
<tr><td colspan="2">**What you'll get**</td><td>7 bytes.</td></tr>
<tr><td colspan="2">**LED indication**</td><td>None.</td></tr>
<tr><td colspan="2">**Related Commands**</td><td>IM 0x58 ALL-Link Cleanup Status Report[29]</td></tr>
<tr><td colspan="3" align="center">**Message Sent from IM to Host**</td></tr>
<tr><td>**Byte**</td><td>**Value**</td><td>**Meaning**</td></tr>
<tr><td>1</td><td>0x02</td><td>Start of IM Command</td></tr>
<tr><td>2</td><td>**0x56**</td><td>IM Command Number</td></tr>
<tr><td>3</td><td>0x01</td><td>Indicates that this ALL-Link Group member did not acknowledge an ALL-Link Cleanup Command.</td></tr>
<tr><td>4</td><td><ALL-Link Group></td><td>Indicates the ALL-Link Group Number that was sent in the ALL-Link Cleanup Command.</td></tr>
<tr><td>5</td><td><ID high byte></td><td>The high byte of the INSTEON ID of the device that did not respond.</td></tr>
<tr><td>6</td><td><ID middle byte></td><td>The middle byte of the INSTEON ID of the device that did not respond.</td></tr>
<tr><td>7</td><td><ID low byte></td><td>The low byte of the INSTEON ID of the device that did not respond.</td></tr>
<tr><td colspan="3" align="center">**Notes**</td></tr>
<tr><td colspan="3">The IM automatically sends ALL-Link Cleanup messages to each member of an ALL-Link Group following an ALL-Link Broadcast message. If the IM detects other INSTEON traffic during this process, it will abort the ALL-Link Cleanup sequence. If the Cleanup sequence is aborted, you will not receive this message nor will you receive a Cleanup acknowldgement message for any subsequent devices in the ALL-Link Group. The Cleanup sequence proceeds in the order in which the devices in the ALL-Link Group were added to the ALL-Link Database.</td></tr>
<tr><td colspan="3">For each ALL-Link Cleanup message the IM sends, you will either receive an *INSTEON Standard Message Received*[18] when the Responder sends you an ACK, or you will receive this message. However, it can take awhile before you receive this message. Worst case, if the IM has to wait for a clear line and then retries the Cleanup message for the maximum of five times, the wait will be 2.150 seconds after sending the ALL-Link Broadcast message, or 1.550 seconds after receiving the first Cleanup acknowledgement or this message. If the Cleanup sequence was aborted due to other INSTEON traffic, you will not get this message even then. However, you will receive *ALL-Link Cleanup Status Report*[29] with a *Status Byte* of 0x15 (NAK) indicating that the Cleanup sequence was aborted.</td></tr>
<tr><td colspan="3">It is possible that this ALL-Link Group member did in fact properly receive the ALL-Link Broadcast message that preceded the ALL-Link Cleanup message.</td></tr>
</table>

ALL-Link Cleanup Status Report

ALL-Link Cleanup Status Report (0x58)	
What it does	Notifies you if a _Send ALL-Link Command_26 completed with all Cleanup messages sent, or else if Cleanups were interrupted due to other INSTEON traffic.
When you'll get this	After you issue a _Send ALL-Link Command_26 and the IM finishes sending Cleanups to all members of the ALL-Link Group, or else when the Cleanup sequence is aborted due to other INSTEON traffic.
What you'll get	3 bytes.
LED indication	None.
Related Commands	IM 0x61 Send ALL-Link Command26 IM 0x56 ALL-Link Cleanup Failure Report28

Message Sent from IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x58**	IM Command Number
3	<Status Byte>	<0x06> (ASCII ACK) The ALL-Link Command sequence initiated previously using _Send ALL-Link Command_26 completed. The IM first sent an ALL-Link Broadcast message, followed by ALL-Link Cleanup messages sent to all members of the specified ALL-Link Group. If any member of the ALL-Link Group does not return a Cleanup acknowledgement, you will receive an _ALL-Link Cleanup Failure Report_28 from that member.
		<0x15> (ASCII NAK) The ALL-Link Command sequence initiated previously using _Send ALL-Link Command_26 terminated before the IM sent ALL-Link Cleanup messages to all members of the specified ALL-Link Group. This is normal behavior when the IM detects INSTEON traffic from other devices.

Notes
The IM automatically sends ALL-Link Cleanup messages to each member of an ALL-Link Group following an ALL-Link Broadcast message. If the IM detects other INSTEON traffic during this process, it will abort the ALL-Link Cleanup sequence and send you this message with a _Status Byte_ of 0x15 (NAK). The Cleanup sequence proceeds in the order in which the devices in the ALL-Link Group were added to the ALL-Link Database. If the IM finishes sending _all_ of the Cleanup messages, it will send you this message with a _Status Byte of_ 0x06 (ACK).
For _each_ ALL-Link Cleanup message that the IM sends, you will either receive an _INSTEON Standard Message Received_18 when the Responder answers with a Cleanup acknowledgement message, or else you will receive an _ALL-Link Cleanup Failure Report_28 if the Responder fails to answer with a Cleanup acknowledgement message. The IM will send you _this_ message whether or not every ALL-Link Group member acknowledges the Cleanup Command that the IM sends to it.
You can cause the IM to cancel its own Cleanup sequence by sending it a new _Send ALL-Link Command_26 or _Send INSTEON Standard or Extended Message_15 during the time that it is sending a Cleanup sequence (i.e. _after_ it has finished sending an ALL-Link Broadcast message). The IM _will_ send you this message in those cases.

ALL-Linking Session Management

Start ALL-Linking

Start ALL-Linking (0x64)	
What it does	Puts the IM into ALL-Linking mode without using the SET Button.
What you send	4 bytes.
What you'll get	5 bytes for this Command response and then an additional 10 bytes in an *ALL-Linking Completed*$_{32}$ message once a successful ALL-Link has been established.
LED indication	The LED will blink continuously at a rate of ½ second on and ½ second off until the ALL-Link is completed or canceled.
Related Commands	IM 0x53 ALL-Linking Completed$_{32}$ IM 0x65 Cancel ALL-Linking$_{31}$

Command Sent from Host to IM			
Byte	**Value**	**Meaning**	
1	0x02	Start of IM Command	
2	**0x64**	IM Command Number	
3	<Link Code>	The type of ALL-Link to establish.	
		0x00	ALL-Links the IM as a Responder (slave).
		0x01	ALL-Links the IM as a Controller (master).
		0x03	ALL-Links the IM as a Controller when the IM initiates ALL-Linking, or as a Responder when another device initiates ALL-Linking.
		0xFF	Deletes the ALL-Link.
4	<ALL-Link Group>	The ALL-Link Group Number to be linked to or deleted.	

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x64**	Echoed IM Command Number
3	<Code>	Echoed <Code>
4	<ALL-Link Group>	Echoed <ALL-Link Group>
5	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly 0x15 (NAK) if an error occurred

Cancel ALL-Linking

Cancel ALL-Linking (0x65)	
What it does	Cancels the ALL-Linking process that was started either by holding down the IM's SET Button or by sending a *Start ALL-Linking*[30] Command to the IM.
What you send	2 bytes.
What you'll get	3 bytes.
LED indication	The LED will stop blinking.
Related Commands	IM 0x64 Start ALL-Linking[30] IM 0x54 Button Event Report[48]

Command Sent from Host to IM		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x65**	IM Command Number

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x65**	Echoed IM Command Number
3	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly 0x15 (NAK) if an error occurred

ALL-Linking Completed

<table>
<tr><th colspan="2">ALL-Linking Completed (0x53)</th></tr>
<tr><td>What it does</td><td>Informs you of a successful ALL-Linking procedure.</td></tr>
<tr><td>When you'll get this</td><td>An ALL-Linking procedure has been completed between the IM and either a Controller or Responder.</td></tr>
<tr><td>What you'll get</td><td>10 bytes.</td></tr>
<tr><td>LED indication</td><td>None.</td></tr>
<tr><td>Related Commands</td><td>IM 0x64 Start ALL-Linking[30]
IM 0x65 Cancel ALL-Linking[31]</td></tr>
</table>

Message Sent from IM to Host		
Byte	Value	Meaning
1	0x02	Start of IM Command
2	**0x53**	IM Command Number
3	<Link Code>	Indicates the type of link made. 　0x00 means the IM is a Responder (slave) to this device 　0x01 means the IM is a Controller (master) of this device 　0xFF means the ALL-Link to the device was deleted If done manually (by pushing the SET Button) the Controller / Responder relationship between the IM and the device is determined automatically. You can assign the Controller / Responder relationship unconditionally by using the *Start ALL-Linking*[30] Command.
4	<ALL-Link Group>	Indicates the ALL-Link Group Number that was assigned to this link. If done manually (by pushing the SET Button) the ALL-Link Group Number is automatically assigned by the IM. You can assign ALL-Link Group Numbers unconditionally by using the *Start ALL-Linking*[30] Command.
5	<ID high byte>	The high byte of the INSTEON ID of the device that was ALL-Linked.
6	<ID middle byte>	The middle byte of the INSTEON ID of the device that was ALL-Linked.
7	<ID low byte>	The low byte of the INSTEON ID of the device that was ALL-Linked.
8	<Device Category>	The Device Category (DevCat) of the Responder device that was ALL-Linked. 　(Only valid when the IM is a Controller)
9	<Device Subcategory>	The Device Subcategory (SubCat) of the Responder device that was ALL-Linked. 　(Only valid when the IM is a Controller)
10	<0xFF | Firmware Version>	0xFF for newer devices. For legacy devices this is the firmware version of the Responder device that was ALL-Linked. 　(Only valid when the IM is a Controller)

ALL-Link Database Management

Get First ALL-Link Record

Get First ALL-Link Record (0x69)	
What it does	Returns the first record in the IM's ALL-Link Database. The data will follow in an *ALL-Link Record Response*36 message.
What you send	2 bytes.
What you'll get	3 bytes.
LED indication	None.
Related Commands	IM 0x57 ALL-Link Record Response36 IM 0x6A Get Next ALL-Link Record34 IM 0x6C Get ALL-Link Record for Sender35

Command Sent from Host to IM		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x69**	IM Command Number

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x69**	Echoed IM Command Number
3	<ACK/NAK>	0x06 (ACK) if an *ALL-Link Record Response*36 follows 0x15 (NAK) if the database is empty.

Note
Use this to begin scanning the IM's ALL-Link Database. Follow up with *Get Next ALL-Link Record*34 Commands until you receive a NAK.
In the *IM Factory Reset State*8 the ALL-Link Database will be cleared, so you will receive a NAK.

Get Next ALL-Link Record

<table>
<tr><th colspan="3">Get Next ALL-Link Record (0x6A)</th></tr>
<tr><td>What it does</td><td colspan="2">Returns the next record in the IM's ALL-Link Database. The data will follow in an ALL-Link Record Response36 message.</td></tr>
<tr><td>What you send</td><td colspan="2">2 bytes.</td></tr>
<tr><td>What you'll get</td><td colspan="2">3 bytes.</td></tr>
<tr><td>LED indication</td><td colspan="2">None.</td></tr>
<tr><td>Related Commands</td><td colspan="2">IM 0x57 ALL-Link Record Response36
IM 0x69 Get First ALL-Link Record33
IM 0x6C Get ALL-Link Record for Sender35</td></tr>
<tr><th colspan="3">Command Sent from Host to IM</th></tr>
<tr><td>Byte</td><td>Value</td><td>Meaning</td></tr>
<tr><td>1</td><td>0x02</td><td>Start of IM Command</td></tr>
<tr><td>2</td><td>0x6A</td><td>IM Command Number</td></tr>
<tr><th colspan="3">Message Returned by IM to Host</th></tr>
<tr><td>Byte</td><td>Value</td><td>Meaning</td></tr>
<tr><td>1</td><td>0x02</td><td>Echoed Start of IM Command</td></tr>
<tr><td>2</td><td>0x6A</td><td>Echoed IM Command Number</td></tr>
<tr><td>3</td><td><ACK/NAK></td><td>0x06 (ACK) if an ALL-Link Record Response36 follows
0x15 (NAK) if there are no more records.</td></tr>
<tr><th colspan="3">Note</th></tr>
<tr><td colspan="3">Use this to continue scanning the IM's ALL-Link Database until you receive a NAK. Begin the scan up with a Get First ALL-Link Record33 Command.</td></tr>
<tr><td colspan="3">In the IM Factory Reset State8 the ALL-Link Database will be cleared, so you will receive a NAK.</td></tr>
</table>

Get ALL-Link Record for Sender

Get ALL-Link Record for Sender (0x6C)	
What it does	This gets the record from the IM's ALL-Link Database for the last INSTEON message received from an INSTEON device that is in the IM's ALL-Link Database. The data will follow in an _ALL-Link Record Response_36 message.
What you send	2 bytes.
What you'll get	3 bytes.
LED indication	None.
Related Commands	IM 0x57 ALL-Link Record Response_36 IM 0x69 Get First ALL-Link Record_33 IM 0x6A Get Next ALL-Link Record_34

Command Sent from Host to IM		
Byte	Value	Meaning
1	0x02	Start of IM Command
2	**0x6C**	IM Command Number

Message Returned by IM to Host		
Byte	Value	Meaning
1	0x02	Echoed Start of IM Command
2	**0x6C**	Echoed IM Command Number
3	<ACK/NAK>	0x06 (ACK) if an _ALL-Link Record Response_36 follows 0x15 (NAK) if the last INSTEON message received had a _From Address_ not in the IM's ALL-Link Database.

Note
If you send this after receiving an INSTEON message from an INSTEON device that is not in the IM's ALL-Link Database, you will receive a NAK in response.
Sending a _Get Next ALL-Link Record_34 Command after this will return the ALL-Link Record that follows this one, but your actual position within the ALL-Link Database will be unknown (unless you are at the end).
In the _IM Factory Reset State_8 the ALL-Link Database will be cleared, so you will receive a NAK.

ALL-Link Record Response

ALL-Link Record Response (0x57)	
What it does	Provides a record from the IM's ALL-Link Database.
When you'll get this	You get this when you have requested it, in response to a *Get First ALL-Link Record*[33] a *Get Next ALL-Link Record*[34] , or a *Get ALL-Link Record for Sender*[35] Command.
What you'll get	10 bytes.
LED indication	None.
Related Commands	IM 0x69 Get First ALL-Link Record[33] IM 0x6A Get Next ALL-Link Record[34] IM 0x6C Get ALL-Link Record for Sender[35]

Message Sent from IM to Host			
Byte	**Value**	**Meaning**	
1	0x02	Start of IM Command	
2	**0x57**	IM Command Number	
3	<ALL-Link Record Flags>	ALL-Link Database control flags for this ALL-Link Record	
		Bit 7	1 = Record is in use 0 = Record is available Note: This bit will always be set to 1.
		Bit 6	1 = IM is a Controller (master) of the device with <ID> given in bytes 5 – 7 below, 0 = IM is a Responder (slave) to the device with <ID> given in bytes 5 – 7 below
		Bit 5	Product dependent
		Bit 4	Product dependent
		Bit 3	Reserved (set to 0)
		Bit 2	Reserved (set to 0)
		Bit 1	1 = Record has been used before 0 = 'High-water Mark' Note: This bit will always be set to 1.
		Bit 0	Reserved (set to 0)
4	<ALL-Link Group>	ALL-Link Group Number for this ALL-Link Record	
5	<ID high byte>	INSTEON ID high byte for device ALL-Linked to	
6	<ID middle byte>	INSTEON ID middle byte for device ALL-Linked to	
7	<ID low byte>	INSTEON ID low byte for device ALL-Linked to	
8	<Link Data 1>	Link Information (varies by device ALL-Linked to)	
9	<Link Data 2>	Link Information (varies by device ALL-Linked to)	
10	<Link Data 3>	Link Information (varies by device ALL-Linked to)	
Note			
See the section *INSTEON All-Link Database* in the *INSTEON Developer's Guide*[2] for details about the contents of an ALL-Link Record.			

Manage ALL-Link Record

<table>
<tr><td colspan="3" align="center">Manage ALL-Link Record (0x6F)</td></tr>
<tr><td>What it does</td><td colspan="2">Updates the IM's ALL-Link Database (ALDB) with the ALL-Link Record information you send. Use caution with this Command—the IM does not check the validity of the data.</td></tr>
<tr><td>What you send</td><td colspan="2">11 bytes.</td></tr>
<tr><td>What you'll get</td><td colspan="2">12 bytes.</td></tr>
<tr><td>LED indication</td><td colspan="2">None.</td></tr>
<tr><td>Related Commands</td><td colspan="2">IM 0x57 <u>ALL-Link Record Response</u>₃₆</td></tr>
<tr><td colspan="3" align="center">Command Sent from Host to IM</td></tr>
<tr><td>Byte</td><td>Value</td><td>Meaning</td></tr>
<tr><td>1</td><td>0x02</td><td>Start of IM Command</td></tr>
<tr><td>2</td><td>0x6F</td><td>IM Command Number</td></tr>
<tr><td>3</td><td><Control Code></td><td>What to do with the ALL-Link Record</td></tr>
</table>

	0x00	**Find First**
		Starting at the top of the ALDB, search for the first ALL-Link Record matching the <ALL-Link Group> and <ID> in bytes 5 – 8. The search ignores byte 4, <ALL-Link Record Flags>.
		You will receive an ACK at the end of the returned message if such an ALL-Link Record exists, or else a NAK if it doesn't. If the record exists, the IM will return it in an _ALL-Link Record Response_$_{36}$ message.
	0x01	**Find Next**
		Search for the next ALL-Link Record following the one found using <Control Code> 0x00 above. This allows you to find both Controller and Responder records for a given <ALL-Link Group> and <ID>. Be sure to use the same <ALL-Link Group> and <ID> (bytes 5 – 8) as you used for <Control Code> 0x00.
		You will receive an ACK at the end of the returned message if another matching ALL-Link Record exists, or else a NAK if it doesn't. If the record exists, the IM will return it in an _ALL-Link Record Response_$_{36}$ message.
	0x20	**Modify First Found or Add**
		Modify an existing or else add a new ALL-Link Record for either a Controller or Responder.
		Starting at the top of the ALDB, search for the first ALL-Link Record matching the <ALL-Link Group> and <ID> in bytes 5 – 8. The search ignores byte 4, <ALL-Link Record Flags>.
		If such an ALL-Link Record exists, overwrite it with the data in bytes 4 – 11; otherwise, create a _new_ ALL-Link Record using bytes 4 – 11.
		Note that the IM will copy <ALL-Link Record Flags> you supplied in byte 4 below directly into the <ALL-Link Record Flags> byte of the ALL-Link Record in an ALDB-L (linear) database. Use caution, because you can damage an ALDB-L if you misuse this Command. For instance, if you zero the <ALL-Link Record Flags> byte in the first ALL-Link Record, the IM's ALDB-L database will then appear empty.

Manage ALL-Link Record (0x6F)			
		0x40	**Modify First Controller Found or Add** Modify an existing or else add a new Controller (master) ALL-Link Record. Starting at the top of the ALDB, search for the first ALL-Link Controller Record matching the <ALL-Link Group> and <ID> in bytes 5 – 8. An ALL-Link Controller Record has bit 6 of its <ALL-Link Record Flags> byte set to 1. If such a Controller ALL-Link Record exists, overwrite it with the data in bytes 5 – 11; otherwise, create a *new* ALL-Link Record using bytes 5 – 11. In either case, the IM will set bit 6 of the <ALL-Link Record Flags> byte in the ALL-Link Record to 1 to indicate that the record is for a Controller.
		0x41	**Modify First Responder Found or Add** Modify an existing or else add a new Responder (slave) ALL-Link Record. Starting at the top of the ALDB, search for the first ALL-Link Responder Record matching the <ALL-Link Group> and <ID> in bytes 5 – 8. An ALL-Link Responder Record has bit 6 of its <ALL-Link Record Flags> byte cleared to 0. If such a Responder ALL-Link Record exists, overwrite it with the data in bytes 5 – 11; otherwise, create a *new* ALL-Link Record using bytes 5 – 11. In either case, The IM will clear bit 6 of the <ALL-Link Record Flags> byte in the ALL-Link Record to 0 to indicate that the record is for a Responder.
		0x80	**Delete First Found** Delete an ALL-Link Record. Starting at the top of the ALDB, search for the first ALL-Link Record matching the <ALL-Link Group> and <ID> in bytes 5 – 8. The search ignores byte 4, <ALL-Link Record Flags>. You will receive an ACK at the end of the returned message if such an ALL-Link Record existed and was deleted, or else a NAK no such record exists.
4	<ALL-Link Record Flags>	ALL-Link Database control flags for this ALL-Link Record	
		Bit 7	1 = Record is in use 0 = Record is available Note: This bit is only settable for ALDB-L (linear) databases using <Control Code> 0x20. It is handled automatically by ALDB-T (threaded) databases.
		Bit 6	1 = IM is a Controller (master) of the device with <ID> given in bytes 6 – 8 below, 0 = IM is a Responder (slave) to the device with <ID> given in bytes 6 – 8 below
		Bit 5	Product dependent
		Bit 4	Product dependent
		Bit 3	Reserved (set to 0)
		Bit 2	Reserved (set to 0)
		Bit 1	1 = Record has been used before 0 = 'High-water Mark' Note: This bit is only settable for ALDB-L (linear) databases using <Control Code> 0x20. It is handled automatically by ALDB-T (threaded) databases.
		Bit 0	Reserved (set to 0)
5	<ALL-Link Group>	ALL-Link Group Number for this ALL-Link Record	
6	<ID high byte>	INSTEON ID high byte for device ALL-Linked to	
7	<ID middle byte>	INSTEON ID middle byte for device ALL-Linked to	
8	<ID low byte>	INSTEON ID low byte for device ALL-Linked to	
9	<Link Data 1>	Link Information: varies by device ALL-Linked to	

Manage ALL-Link Record (0x6F)		
10	<Link Data 2>	Link Information: varies by device ALL-Linked to
11	<Link Data 3>	Link Information: varies by device ALL-Linked to
Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x6F**	Echoed IM Command Number
3	<Control Code>	Echoed <Control Code>
4	<ALL-Link Record Flags>	Echoed <ALL-Link Record Flags>
5	<ALL-Link Group>	Echoed <ALL-Link group>
6	<ID high byte>	Echoed <ID high byte>
7	<ID middle byte>	Echoed <ID middle byte>
8	<ID low byte>	Echoed <ID low byte>
9	<Link Data 1>	Echoed <Link Data 1>
10	<Link Data 2>	Echoed <Link Data 2>
11	<Link Data 3>	Echoed <Link Data 3>
12	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly. 0x15 (NAK) if an error occurred or the ALL-Link Record doesn't exist.
Notes		
See the section *INSTEON All-Link Database* in the <u>*INSTEON Developer's Guide₂*</u> for details about the contents of an ALL-Link Record.		
Note that except for **Find Next**, all ALDB searches begin at the top of the ALDB, which means that the IM will *always* perform modify or delete operations on the *first* matching record that it finds in its ALDB. You must therefore use caution when deleting records because an ALDB may contain multiple Controller and Responder records matching a given <ALL-Link Group> and <ID>. If you want to delete only one of the matching records, then you should: 1. Use **Find First** and **Find Next** to find and buffer *all* of the records matching the <ALL-Link Group> and <ID>, 2. Use **Delete First Found** enough times to delete *all* of the matching records, 3. Put back the records you did *not* want to delete using **Modify First Controller Found or Add** or **Modify First Responder Found or Add**.		
Please be aware that you can damage an ALDB-L (linear) database if you misuse the **Modify First Found or Add** operation, <Control Code> 0x20. For instance, if you zero bit 1 of the <ALL-Link Record Flags> byte in the first record in an ALDB-L, the ALDB-L will then appear *empty*. Or, if you zero bit 7 of the <ALL-Link Record Flags> byte in an ALDB-L record, then that record will appear *deleted*. In an ALDB-T (threaded) database, the IM handles those bits for you automatically, so the **Modify First Found or Add** operation is not so risky. Nevertheless, it is always preferable to use **Modify First Controller Found or Add** or **Modify First Responder Found or Add** instead, because with **Modify First Found or Add** you cannot be sure if you are modifying a record for a Controller or Responder.		

IM Status Management

Reset the IM

Reset the IM (0x67)	
What it does	Puts the IM into the _IM Factory Reset State_8, which clears the entire ALL-Link Database.
What you send	2 bytes.
What you'll get	3 bytes.
LED indication	While the reset procedure is being processed, the Status LED will turn off. At the conclusion of the reset procedure, the Status LED will illuminate steadily.
Related Commands	IM 0x55 User Reset Detected41

Command Sent from Host to IM		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x67**	IM Command Number

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x67**	Echoed IM Command Number
3	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly 0x15 (NAK) if an error occurred

Notes

The IM will send the <ACK/NAK> byte after it erases the EEPROM.
    ~20 seconds for models with external EEPROM
    ~2 seconds for models with no external EEPROM

See the _IM Factory Reset State_8 section for complete information on the state of the IM after sending this Command.

User Reset Detected

User Reset Detected (0x55)	
What it does	Reports that the user manually put the IM into the *IM Factory Reset State*$_8$.
When you'll get this	The user held down the IM's SET Button for at least 10 seconds when power was first applied.
What you'll get	2 bytes (not until about 20 seconds after applying power to the IM with the SET Button held down).
LED indication	The LED will turn off for about 20 seconds. Once the LED turns back on the reset is complete.
Related Commands	IM 0x67 Reset the IM$_{40}$

Message Sent from IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x55**	IM Command Number

Notes

The IM will send this message after it erases the EEPROM.
    ~20 seconds for models with external EEPROM
    ~2 seconds for models with no external EEPROM

See the *IM Factory Reset State*$_8$ section for complete information on the state of the IM after receiving this message.

Get IM Configuration

Get IM Configuration (0x73)	
What it does	Returns the IM's Configuration Flags byte. Also returns two spare bytes of data reserved for future use.
What you send	2 bytes.
What you'll get	6 bytes.
LED indication	None.
Related Commands	IM 0x6B Set IM Configuration$_{43}$

Command Sent from Host to IM		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x73**	IM Command Number

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x73**	Echoed IM Command Number
3	<IM Configuration Flags>	IM's Configuration Flags. See _Set IM Configuration_$_{43}$ for bit definitions.
4	<Spare 1>	0x00, reserved for future use
5	<Spare 2>	0x00, reserved for future use
6	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly 0x15 (NAK) if an error occurred

Note
Because _Set IM Configuration_$_{43}$ sets all of the <IM Configuration Flags> at once, to change an individual bit, first use this Command to determine the current state of all of the <IM Configuration Flags>.

Set IM Configuration

Set IM Configuration (0x6B)	
What it does	Allows you change operating parameters of the IM.
What you send	3 bytes.
What you'll get	4 bytes.
LED indication	None.
Related Commands	IM 0x73 Get IM Configuration$_{42}$ IM 0x54 Button Event Report$_{48}$ IM 0x50 INSTEON Standard Message Received$_{18}$ IM 0x51 INSTEON Extended Message Received$_{19}$ IM 0x6D LED On$_{49}$ IM 0x6E LED Off$_{50}$

Command Sent from Host to IM			
Byte	**Value**	**Meaning**	
1	0x02	Start of IM Command	
2	**0x6B**	IM Command Number	
3	<IM Configuration Flags>	Flag byte containing Configuration Flags that affect IM operation. These all default to 0.	
		Bit 7 = 1	Disables automatic linking when the user pushes and holds the SET Button (see *Button Event Report*$_{48}$).
		Bit 6 = 1	Puts the IM into *Monitor Mode* (see *About Monitor Mode*$_{44}$ in the **Notes** below).
		Bit 5 = 1	Disables automatic LED operation by the IM. The host must now control the IM's LED using *LED On*$_{49}$ and *LED Off*$_{50}$.
		Bit 4 = 1	Disable host communications *Deadman* feature (i.e. allow host to delay more than 240 milliseconds between sending bytes to the IM). See *IM RS232 Port Settings*$_6$.
		Bits 3 - 0	Reserved for internal use. **Set these bits to 0.**

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x6B**	Echoed IM Command Number
3	<IM Configuration Flags>	Echoed <IM Configuration Flags>
4	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly. 0x15 (NAK) if an error occurred.

Notes
When the IM is in the *IM Factory Reset State*$_8$, the <IM Configuration Flags> will all be set to zero.
This Command sets all of the <IM Configuration Flags> at once. To change an individual bit, first use *Get IM Configuration*$_{42}$ to determine the current state of all of the <IM Configuration Flags>.

Set IM Configuration (0x6B)

About Monitor Mode

Normally, the IM will only send the host an *INSTEON Standard Message Received*[18] or *INSTEON Extended Message Received*[19] notification when it receives an INSTEON messages directed specifically to the IM. There are three possibilities:

1. The IM received a Direct message with a *To Address* matching the IM's INSTEON ID,
2. The IM received an ALL-Link Broadcast message sent to an ALL-Link Group that the IM belongs to as a Responder (i.e. the message's *From Address* and ALL-Link Group Number match a Responder entry in the IM's ALL-Link Database), or
3. The IM received an ALL-Link Cleanup message with a *To Address* matching the IM's INSTEON ID and the message's *From Address* and ALL-Link Group Number match a Responder entry in the IM's ALL-Link Database.

In *Monitor Mode*, the IM will also notify the host of received INSTEON messages that contain a *From Address* matching *any* INSTEON ID in the IM's ALL-Link Database, even if the *To Address* does not match the IM's INSTEON ID or the IM does not belong to an ALL-Link Group associated with the message. In other words, if the message originator is in the IM's ALL-Link Database as either a Controller or Responder, the IM will pass the message to the host even if it is not specifically directed to the IM. In this way you can monitor messages between other INSTEON devices as long as the sender is in the IM's ALL-Link Database.

Please be aware that the IM may not always detect this traffic. If the message originator and addressee are close to one another and the IM is farther away, the message originator may not cause the message to hop enough times for the IM to hear it. To know for sure what an INSTEON device's status is, you can usually query it directly using an appropriate INSTEON Direct Command. For more information on INSTEON Commands and the latest Command set, please download the current *INSTEON Command Tables Document*[2] from http://code.insteon.net.

Get IM Info

<table>
<tr><th colspan="2">Get IM Info (0x60)</th></tr>
<tr><td>What it does</td><td>Identifies the IM's 3 byte INSTEON ID, Device Category (DevCat), Device Subcategory (SubCat), and firmware version.</td></tr>
<tr><td>What you send</td><td>2 bytes.</td></tr>
<tr><td>What you'll get</td><td>9 bytes.</td></tr>
<tr><td>LED indication</td><td>None.</td></tr>
<tr><td>Related Commands</td><td>IM 0x66 Set Host Device Category[46]
IM 0x73 Get IM Configuration[42]
IM 0x6B Set IM Configuration[43]</td></tr>
<tr><td colspan="2" align="center">Command Sent from Host to IM</td></tr>
</table>

Byte	Value	Meaning
1	0x02	Start of IM Command
2	**0x60**	IM Command Number

<table><tr><td colspan="3" align="center">Message Returned by IM to Host</td></tr></table>

Byte	Value	Meaning
1	0x02	Echoed Start of IM Command
2	**0x60**	Echoed IM Command Number
3	<ID high byte>	IM's INSTEON ID high byte
4	<ID middle byte>	IM's INSTEON ID middle byte
5	<ID low byte>	IM's INSTEON ID low byte
6	<Device Category>	IM's Device Category
7	<Device Subcategory>	IM's Device Subcategory
8	<Firmware Version>	IM's Firmware Version
9	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly 0x15 (NAK) if an error occurred

<table>
<tr><td align="center">Note</td></tr>
<tr><td>Using the Set Host Device Category[46] Command to change the host's DevCat and SubCat will only affect the data transmitted by the IM to other INSTEON devices during ALL-Linking.
When the host sends this Command to the IM, the IM will return the original DevCat, SubCat and firmware version hard-coded into the IM's firmware at the factory.</td></tr>
</table>

Set Host Device Category

<table>
<tr><td colspan="3" align="center">**Set Host Device Category (0x66)**</td></tr>
<tr><td>**What it does**</td><td colspan="2">Lets you set the Device Category (DevCat) and Device Subcategory (SubCat) of the host device connected to the IM.</td></tr>
<tr><td>**What you send**</td><td colspan="2">5 bytes.</td></tr>
<tr><td>**What you'll get**</td><td colspan="2">6 bytes.</td></tr>
<tr><td>**LED indication**</td><td colspan="2">None.</td></tr>
<tr><td>**Related Commands**</td><td colspan="2">IM 0x60 Get IM Info₄₅</td></tr>
<tr><td colspan="3" align="center">**Command Sent from Host to IM**</td></tr>
<tr><td>**Byte**</td><td>**Value**</td><td>**Meaning**</td></tr>
<tr><td>1</td><td>0x02</td><td>Start of IM Command</td></tr>
<tr><td>2</td><td>**0x66**</td><td>IM Command Number</td></tr>
<tr><td>3</td><td><Device Category></td><td>INSTEON Device Category (DevCat) of the host device connected to the IM.</td></tr>
<tr><td>4</td><td><Device Subcategory></td><td>INSTEON Device Subcategory (SubCat) of the host device connected to the IM.</td></tr>
<tr><td>5</td><td><0xFF | Firmware Version></td><td>0xFF
In legacy devices this byte represented a BCD-encoded firmware version. The high nibble (4 bits) gave the major revision number and the low nibble gave the minor revision.
In current devices use the INSTEON *Product Data Request* and *Product Data Response* Commands to retrieve the firmware version as user-defined data.</td></tr>
<tr><td colspan="3" align="center">**Message Returned by IM to Host**</td></tr>
<tr><td>**Byte**</td><td>**Value**</td><td>**Meaning**</td></tr>
<tr><td>1</td><td>0x02</td><td>Echoed Start of IM Command</td></tr>
<tr><td>2</td><td>**0x66**</td><td>Echoed IM Command Number</td></tr>
<tr><td>3</td><td><Device Category></td><td>Echoed <Device Category></td></tr>
<tr><td>4</td><td><Device Subcategory></td><td>Echoed <Device Subcategory></td></tr>
<tr><td>5</td><td><0xFF | Firmware Version></td><td>Echoed <0xFF> or <Firmware Version></td></tr>
<tr><td>6</td><td><ACK/NAK></td><td>0x06 (ACK) if the IM executed the Command correctly
0x15 (NAK) if an error occurred</td></tr>
<tr><td colspan="3" align="center">**Notes**</td></tr>
<tr><td colspan="3">For INSTEON compliance, you must obtain an approved DevCat and SubCat assignment for your host product from SmartLabs.</td></tr>
<tr><td colspan="3">The IM stores these values in EEPROM so they will not be erased if power is lost.</td></tr>
<tr><td colspan="3">When the IM is in the *IM Factory Reset State*₈, these values will be set to those hard-coded into the IM's firmware at the factory.</td></tr>
<tr><td colspan="3">Using this Command to change the host's DevCat and SubCat will only affect the data transmitted by the IM to other INSTEON devices during ALL-Linking.
When the host sends a *Get IM Info*₄₅ Command to the IM, the IM will return the original DevCat, SubCat and firmware version hard-coded into the IM's firmware at the factory.</td></tr>
<tr><td colspan="3">For the latest list of assigned INSTEON DevCats, please download the *INSTEON Device Categories and Product Keys Document*₂ from http://code.insteon.net.</td></tr>
</table>

RF Sleep

<table>
<tr><td colspan="3" align="center">**RF Sleep (0x72)**</td></tr>
<tr><td>**What it does**</td><td colspan="2">Directs an RF IM to go into power saving sleep mode. To wake up the RF IM, send it one byte of serial data.</td></tr>
<tr><td>**What you send**</td><td colspan="2">2 bytes.</td></tr>
<tr><td>**What you'll get**</td><td colspan="2">3 bytes.</td></tr>
<tr><td>**LED indication**</td><td colspan="2">None.</td></tr>
<tr><td>**Related Commands**</td><td colspan="2">None.</td></tr>
<tr><td colspan="3" align="center">**Command Sent from Host to IM**</td></tr>
<tr><td>**Byte**</td><td>**Value**</td><td>**Meaning**</td></tr>
<tr><td>1</td><td>0x02</td><td>Start of IM Command</td></tr>
<tr><td>2</td><td>**0x72**</td><td>IM Command Number</td></tr>
<tr><td>3</td><td><Command 1 Data></td><td>Data byte to place into the *Command 1* field 2 of the ACK response.</td></tr>
<tr><td>4</td><td><Command 2 Data></td><td>Data byte to place into the *Command 2* field 2 of the ACK response.</td></tr>
<tr><td colspan="3" align="center">**Message Returned by IM to Host**</td></tr>
<tr><td>**Byte**</td><td>**Value**</td><td>**Meaning**</td></tr>
<tr><td>1</td><td>0x02</td><td>Echoed Start of IM Command</td></tr>
<tr><td>2</td><td>**0x72**</td><td>Echoed IM Command Number</td></tr>
<tr><td>3</td><td><Command 1 Data></td><td>Echoed <Command 1 Data></td></tr>
<tr><td>4</td><td><Command 2 Data></td><td>Echoed <Command 2 Data></td></tr>
<tr><td>5</td><td><ACK/NAK></td><td>0x06 (ACK) if the IM executed the Command correctly.
0x15 (NAK) if an error occurred.</td></tr>
<tr><td colspan="3" align="center">**Notes**</td></tr>
<tr><td colspan="3">It does not matter what byte you send serially to wake up the RF IM.</td></tr>
<tr><td colspan="3">When the RF IM wakes up, it will reinitialize, but memory will not be altered as it would be in the *IM Factory Reset State*[8]. Wait a minimum of 40 milliseconds before sending any further IM Serial Commands.</td></tr>
</table>

IM Input/Output

Button Event Report

Button Event Report (0x54)	
What it does	Reports user SET Button events.
When you'll get this	The user operates the SET Button, or if they exist, Button 2 or Button 3.
What you'll get	3 bytes.
LED indication	If the event is *SET Button Press and Hold* the IM will automatically go into ALL-Linking mode which will cause the LED to blink continuously at a rate of ½ second on and ½ second off. Automatic linking may be turned off by setting *IM Configuration Flags* bit 7 (see *Set IM Configuration*[43]).
Related Commands	IM 0x53 ALL-Linking Completed[32] IM 0x64 Start ALL-Linking[30] IM 0x65 Cancel ALL-Linking[31]

Message Sent from IM to Host			
Byte	**Value**	**Meaning**	
1	0x02	Start of IM Command	
2	**0x54**	IM Command Number	
3	<Button Event>	Indicates the type of SET Button event that occurred.	
		0x02	The SET Button was *Tapped*
		0x03	There was a SET Button *Press and Hold* for more than three seconds. This automatically puts the IM into ALL-Linking mode unless *IM Configuration Flags* bit 7 is set.
		0x04	The SET Button was released after a SET Button *Press and Hold* event was recorded.
		0x12	Button 2 was *Tapped*
		0x13	There was a Button 2 *Press and Hold* for more than three seconds.
		0x14	Button 2 was released after a Button 2 *Press and Hold* event was recorded.
		0x22	Button 3 was *Tapped*
		0x23	There was a Button 3 *Press and Hold* for more than three seconds.
		0x24	Button 3 was released after a Button 3 *Press and Hold* event was recorded.

LED On

<table>
<tr><td colspan="3" align="center">**LED On (0x6D)**</td></tr>
<tr><td>**What it does**</td><td colspan="2">Turns on the IM's LED if *IM Configuration Flags* bit 5 = 1.</td></tr>
<tr><td>**What you send**</td><td colspan="2">2 bytes.</td></tr>
<tr><td>**What you'll get**</td><td colspan="2">3 bytes.</td></tr>
<tr><td>**LED indication**</td><td colspan="2">The LED will go on.</td></tr>
<tr><td>**Related Commands**</td><td colspan="2">IM 0x6B Set IM Configuration[43]
IM 0x6E LED Off[50]</td></tr>
<tr><td colspan="3" align="center">**Command Sent from Host to IM**</td></tr>
<tr><td>**Byte**</td><td>**Value**</td><td>**Meaning**</td></tr>
<tr><td>1</td><td>0x02</td><td>Start of IM Command</td></tr>
<tr><td>2</td><td>**0x6D**</td><td>IM Command Number</td></tr>
<tr><td colspan="3" align="center">**Message Returned by IM to Host**</td></tr>
<tr><td>**Byte**</td><td>**Value**</td><td>**Meaning**</td></tr>
<tr><td>1</td><td>0x02</td><td>Echoed Start of IM Command</td></tr>
<tr><td>2</td><td>**0x6D**</td><td>Echoed IM Command Number</td></tr>
<tr><td>3</td><td><ACK/NAK></td><td>0x06 (ACK) if the IM executed the Command correctly.
0x15 (NAK) if an error occurred or *IM Configuration Flags* bit 5 = 0.</td></tr>
</table>

LED Off

LED Off (0x6E)	
What it does	Turns off the IM's LED if *IM Configuration Flags* bit 5 = 1.
What you send	2 bytes.
What you'll get	3 bytes.
LED indication	The LED will go off.
Related Commands	IM 0x6B Set IM Configuration₄₃ IM 0x6D LED On₄₉

Command Sent from Host to IM		
Byte	**Value**	**Meaning**
1	0x02	Start of IM Command
2	**0x6E**	IM Command Number

Message Returned by IM to Host		
Byte	**Value**	**Meaning**
1	0x02	Echoed Start of IM Command
2	**0x6E**	Echoed IM Command Number
3	<ACK/NAK>	0x06 (ACK) if the IM executed the Command correctly. 0x15 (NAK) if an error occurred or *IM Configuration Flags* bit 5 = 0.

Hardware Reference

This section gives a reference design for using the IN2680A Powerline Modem chip in a module connected both to the powerline and to a host device. The design uses a main board for the modem chip, power supply, INSTEON powerline interface, and TTL-level serial communications, and a daughter board for interfacing to a host.

Two different daughter board designs are included. One is for an RS232 interface, and the other is for an IP (Ethernet) interface. A USB interface is under development. Developers may create their own daughter cards to implement custom interfaces.

The reference design presented here is the same one that SmartLabs uses for its PowerLinc Modem (PLM) module.

In This Section

INSTEON PowerLinc Modem (PLM) Main Board

The PowerLinc Modem (PLM) main board includes the IN2680A Powerline Modem chip, a transformer-isolated power supply with a 30-volt charge pump booster, a transformer-coupled powerline signal transponder, an optically-isolated zero crossing detector, and an 8-pin daughter board connector for TTL-level host communications.

In This Section

INSTEON PLM Main Board Schematic

Power Line Modem Module
Mainboard Schematic
Rev. D

J4 Pin layout for Power
Line Modem main board
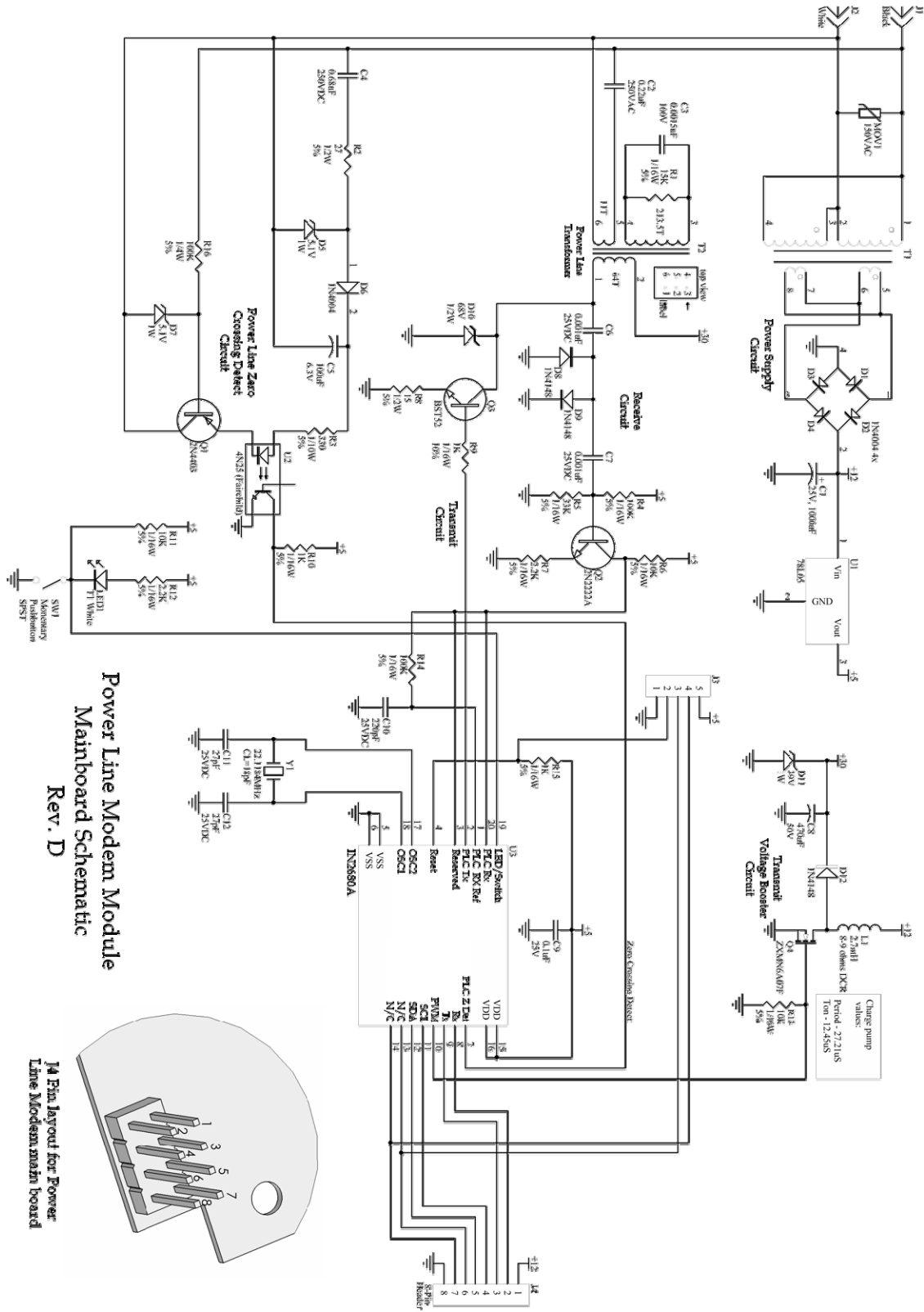
INSTEON PLM Main Board Bill of Materials

Description	Part Type	Designator	Footprint	Remark
Capacitor	Electrolytic, 1000uF, 25V	C1	Through-hole, 0.2"	
Capacitor	Metal Polyester, 0.22uF, 250VDC	C2	Through-hole, 0.3"	
Capacitor	Ceramic, 0.0015uF, 100V	C3	SMT, 0805	
Capacitor	Metal Polyester, 0.68uF, 250VDC	C4	Through-hole, 0.4"	
Capacitor	Electrolytic, 100uF, 6.3V	C5	Through-hole, 0.1"	
Capacitor	Ceramic, 0.001uF, 25V	C6	SMT, 0603	
Capacitor	Ceramic, 0.001uF, 25V	C7	SMT, 0603	
Capacitor	Electrolytic, 470uF, 50V	C8	Through-hole, 0.2"	
Capacitor	Ceramic, 0.1uF,25V	C9	SMT, 0603	
Capacitor	Ceramic, 220pF, 25V	C10	SMT, 0603	
Capacitor	Ceramic, 27pF, 25V	C11	SMT, 0603	
Capacitor	Ceramic, 27pF, 25V	C12	SMT, 0603	
Crystal	22.1184MHz, 18pF Load	Y1	Through-hole	Recommended: Citizen model CMR309T22.1184MABJTR
Diode	DL4004	D1	SMT, MELF	
Diode	DL4004	D2	SMT, MELF	
Diode	DL4004	D3	SMT, MELF	
Diode	DL4004	D4	SMT, MELF	
Diode	Zener, 5.1V, 1W	D5	SMT, MELF	
Diode	DL4004	D6	SMT, MELF	
Diode	Zener, 5.1V, 1W	D7	SMT, MELF	
Diode	1N4148	D8	SMT, Mini-MELF	
Diode	1N4148	D9	SMT, Mini-MELF	
Diode	Zener, 68V, 1/2W	D10	SMT, Mini-MELF	
Diode	Zener, 39V, 1W	D11	SMT, MELF	
Diode	1N4148	D12	SMT, Mini-MELF	
Header	5-Pin male	J3	Through-hole, 0.1" ctr	For in-circuit programming
Header	2X4 male	J4	Through-hole, 0.1" ctr	Used to connect to daughter boards
Inductor	2.7mH, 8-9 ohms DCR, 100mA DCI	L1	Through-hole, 0.2"	
LED	Any single color is acceptable	LED1	Through-hole, T1	
MCU	INSTEON IN2680A	U3	SMT, SSOP20	
MOSFET	N-Channel, Zetex ZXMN6A07F	Q4	SMT, SOT-23	
Optocoupler	Fairchild 4N25SM or 4N25S	U2	SMT	100% Transfer ration @ 8mA If and 5mA Ic
Regulator	78L05 Positive 5V regulator	U1	Through-hole, TO-92	
Resistor	15KW, 1/16W, 5%	R1	SMT, 0603	
Resistor	27W, 1/2W, 5%	R2	SMT, 1210	Recommended: Panasonic ERJ-P14J27OU Anti-Surge
Resistor	330W, 1/10W, 5%	R3	SMT, 0805	

Description	Part Type	Desig-nator	Footprint	Remark
Resistor	100KW, 1/16W, 5%	R4	SMT, 0603	
Resistor	33KW, 1/16W, 5%	R5	SMT, 0603	
Resistor	10KW, 1/16W, 5%	R6	SMT, 0603	
Resistor	2.2KW, 1/16W, 5%	R7	SMT, 0603	
Resistor	15w, 1/2W, 5%	R8	SMT, 2010	
Resistor	1KW, 1/16W, 5%	R9	SMT, 0603	
Resistor	1K, 1/16W, 5%	R10	SMT, 0603	
Resistor	10KW, 1/16W, 5%	R11	SMT, 0603	
Resistor	2.2KW, 1/16W, 5%	R12	SMT, 0603	May be changed to control LED brightness
Resistor	10KW, 1/16W, 5%	R13	SMT, 0603	
Resistor	100KW, 1/16W, 5%	R14	SMT, 0603	
Resistor	1KW, 1/16W, 5%	R15	SMT, 0603	
Resistor	100KW, 1/4W, 5%	R16	SMT, 1206	
Switch	Tact Switch	SW1	Through-hole	
Transformer	Power Transformer, model 710-2000512	T1	Through-hole	Custom made, available from SmartLabs
Transformer	Power line transformer coil	T2	Through-hole	Abracon AIRV-111 PLC
Transistor	2N4403 PNP	Q1	SMT, SOT-23	
Transistor	2N2222A NPN	Q2	SMT, SOT-23	
Transistor	BST-52 Darlington NPN	Q3	SMT, SOT-89	Recommended brand: Zetex
Varistor	150VAC Metal Oxide Varistor	MOV1	Through-hole, 0.2"	
Wire	Hot wire, black, 16AWG, 300V, 105°C, VW-1	J1	Through-hole	In from power prong
Wire	Neutral wire, white, 16AWG, 300V, 105°C, VW-1	J2	Through-hole	In from power prong

INSTEON PLM Serial (RS232) Daughter Board

The Serial Daughter Board attaches to the PowerLinc Modem (PLM) Main Board using an 8-pin connector, and to a host device using an RJ-45 jack. Host communications uses the RS232 protocol at TTL signal levels.

In This Section

INSTEON PLM Serial Daughter Board Schematic
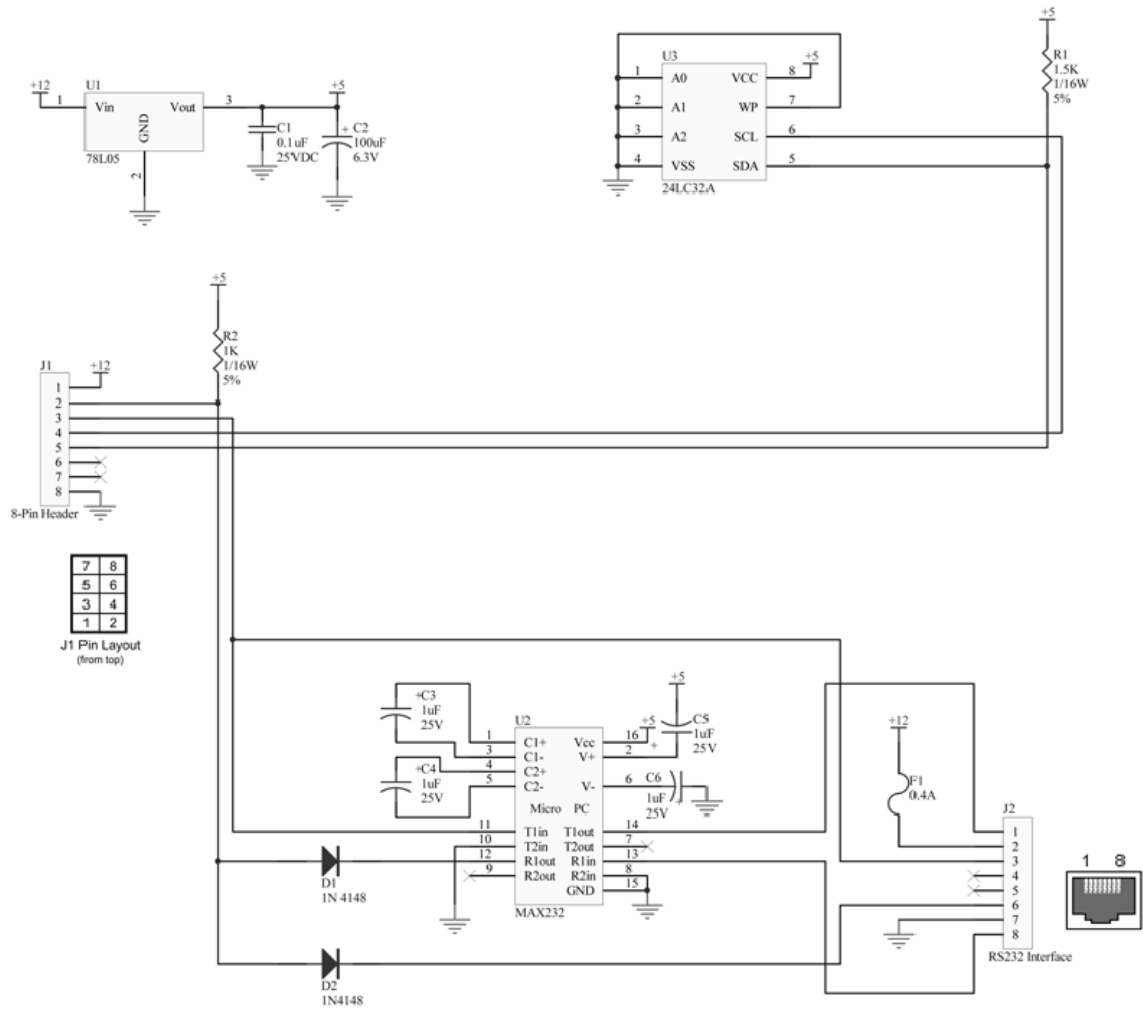
Serial & TTL Daughter
Card Schematic
Rev. A

INSTEON PLM Serial Daughter Board Bill of Materials

Description	Part Type	Desig-nator	Footprint	Remark
Capacitor	Ceramic, 0.1uF, 25V	C1	SMT, 0603	
Capacitor	Electrolytic, 100uF, 6.3V	C2	Through-hole	
Capacitor	Electrolytic, 1uF, 25V	C3	Through-hole	
Capacitor	Electrolytic, 1uF, 25V	C4	Through-hole	
Capacitor	Electrolytic, 1uF, 25V	C5	Through-hole	
Capacitor	Electrolytic, 1uF, 25V	C6	Through-hole	
Diode	1N4148	D1	SMT, Mini-MELF	
Diode	1N4148	D2	SMT, Mini-MELF	
Driver / Receiver	MAX232 Multichannel RS-232 ST232BDR	U2	SMT, SOIC16	
EEPROM	24LC32A	U3	SMT, SOIC8	
Fuse	250V, 0.4A	F1	Through-hole	
Header	Female 2x4, 2x4PIN, 2.54mm, 2185-20	J1	Through-hole, 0.1" ctr	
Jack	Female RJ45	J2	SMT	
Resistor	1.5KΩ, 1/16W, 5%	R1	SMT, 0603	
Resistor	1KΩ, 1/16W, 5%	R2	SMT, 0603	
Voltage Regulator	5V Zetex ZSR500G	U1	SMT, SOT223	

INSTEON PLM Ethernet (IP) Daughter Board

The IP (Ethernet) Daughter Board attaches to the PowerLinc Modem (PLM) Main Board using an 8-pin connector, and to an Ethernet LAN using an RJ-45 jack.

In This Section

INSTEON PLM Ethernet (IP) Daughter Board Schematic
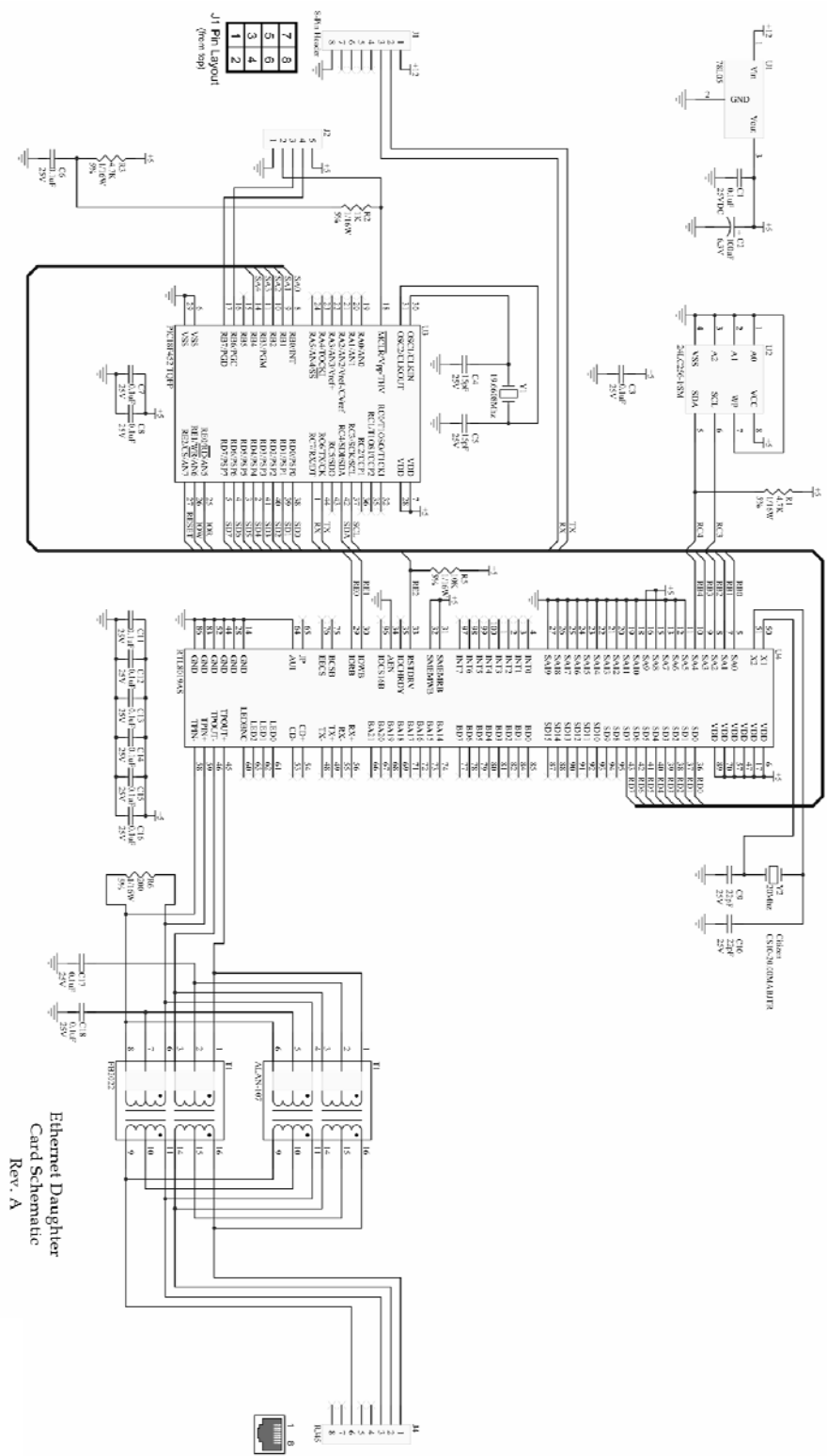
Ethernet Daughter
Card Schematic
Rev. A

INSTEON PLM Ethernet (IP) Daughter Board Bill of Materials

Description	Part Type	Desig-nator	Footprint	Remark
Capacitor	Ceramic, 0.1uF, 25V	C1	SMT, 0603	
Capacitor	Electrolytic, 100uF, 6.3V	C2	Through-hole, 0.1"	
Capacitor	Ceramic, 0.1uF, 25V	C3	SMT, 0603	
Capacitor	Ceramic, 15pF, 25V	C4	SMT, 0603	
Capacitor	Ceramic, 15pF, 25V	C5	SMT, 0603	
Capacitor	Ceramic, 0.1uF, 25V	C6	SMT, 0603	
Capacitor	Ceramic, 0.1uF, 25V	C7	SMT, 0603	
Capacitor	Ceramic, 0.1uF, 25V	C8	SMT, 0603	
Capacitor	Ceramic, 22pF, 25V	C9	SMT, 0603	
Capacitor	Ceramic, 22pF, 25V	C10	SMT, 0603	
Capacitor	Ceramic, 0.1uF, 25V	C11	SMT, 0603	
Capacitor	Ceramic, 0.1uF, 25V	C12	SMT, 0603	
Capacitor	Ceramic, 0.1uF, 25V	C13	SMT, 0603	
Capacitor	Ceramic, 0.1uF, 25V	C14	SMT, 0603	
Capacitor	Ceramic, 0.1uF, 25V	C15	SMT, 0603	
Capacitor	Ceramic, 0.1uF, 25V	C16	SMT, 0603	
Capacitor	Ceramic, 0.1uF, 25V	C17	SMT, 0603	
Capacitor	Ceramic, 0.1uF, 25V	C18	SMT, 0603	
Controller	Ethernet controller, Realtek RTL8019AS	U4	SMT, QFP-100	
Crystal	19.6608MHz Crystal, 18pF Load	Y1	Through-hole	
Crystal	20MHz Crystal, 18pF Load	Y2	SMT	
Header	5-Pin Male	J2	Through-hole, 0.1" ctr	For in-circuit programming
Jack	RJ45 Female jack	J4	SMT	
MCU	PIC18F452-I/PT	U3	SMT, TQFP-44	
Memory	24LC256-I/SN	U2	SMT, SOP-8	
Regulator	78L05 5V Voltage regulator	U1	SMT, SOT-223	
Resistor	4.7KW, 1/16W, 5%	R1	SMT, 0603	
Resistor	1KW, 1/16W, 5%	R2	SMT, 0603	
Resistor	4.7KW, 1/16W, 5%	R3	SMT, 0603	
Resistor	10KW, 1/16W, 5%	R5	SMT, 0603	
Resistor	200W, 1/16W, 5%	R6	SMT, 0603	
Transformer	Ethernet transformer, Abracon ALAN-107	T1	SMT	

POWERLINC MODEM (PLM) QUICK START GUIDE

PowerLinc Modem Developer Kit

This is a reprint of the *PLM Quick Start Guide* included in the SmartLabs 2600S PowerLinc Modem (PLM) Developer Kit.

The purpose of this *Quick Start Guid*e is to help you connect a PowerLinc Modem to your computer and start using it right away to control a light plugged into a Developer LampLinc™. Complete details can be found in the *INSTEON Developer's Guide*located at http://code.insteon.net, which you can access using your INSTEON Developer's username and password.

Included in this kit:

- 1 Developer PowerLinc Modem (PLM) module
- 1 Developer LampLinc module
- 1 Small light bulb to be plugged into the bottom of the Lamplinc.
- 1 RJ45-to-RS232 Serial Cable
- INSTEON Alliance Invitation

What you will need to get started:

- Power strip to plug in both the PowerLinc Modem and the LampLinc
- Computer with internet connection and RS232 serial port
- Trial copy of the software *Docklight Scripting V1.7*, downloaded from http://www.docklight.de/download_en.htm
- The INSTEON sample script for Docklight, downloaded from http://www.insteon.net/includes/scripts/PLM_Basic_Command_Set.zip

Getting Started

Follow the steps below to establish serial communications with the PowerLinc Modem, and then use it to control a light plugged into the Developer LampLinc module.

Step 1

Set up the PLM and Developer LampLinc as shown in the picture below.

- Plug the light bulb into the bottom of the Developer LampLinc.
- Plug the PLM and the Developer LampLinc into a power strip.
- Using the provided cable, connect the PLM to an RS232 serial port on your PC.

Step 2

Unzip *Docklight_Scripting.zip* and run *Setup.exe* to install *Docklight Scripting V1.7*.

Step 3

Launch *DockLight Scripting* and choose *Open Project…* from the *File* menu.
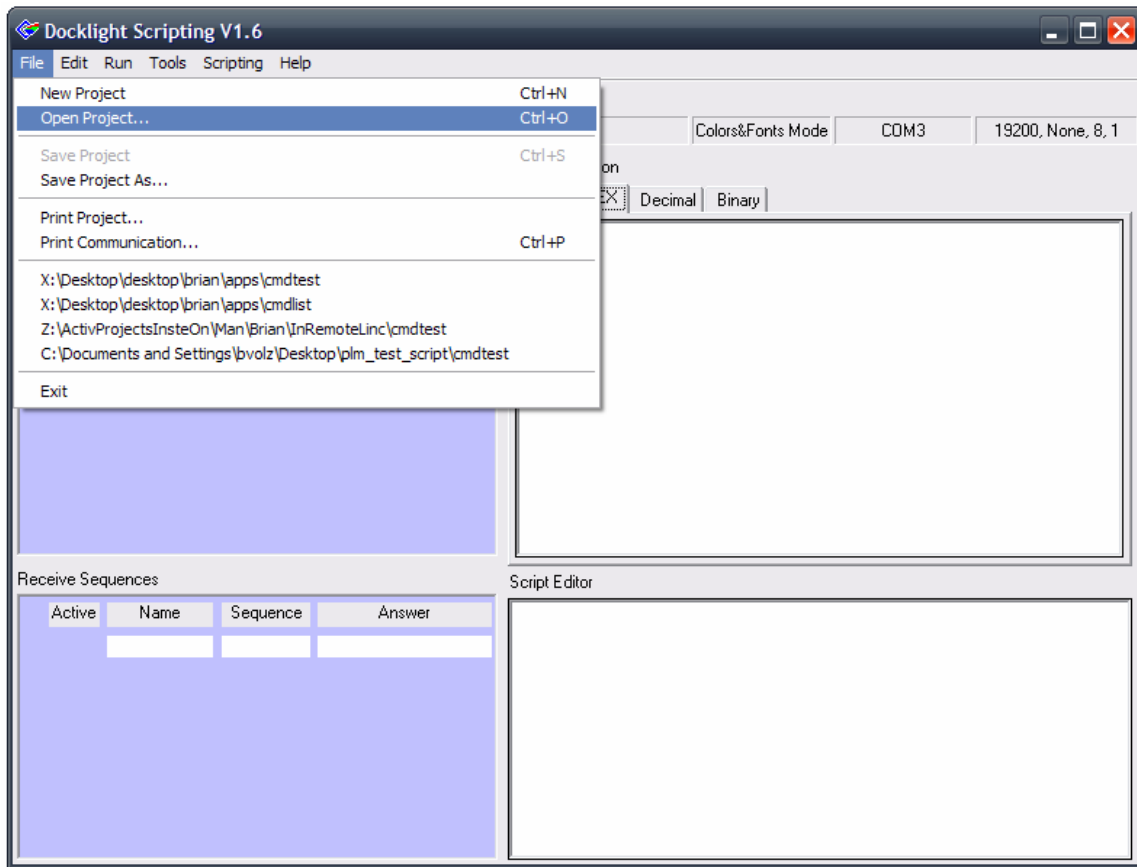
Step 4

Locate and open *PLM_Basic_Command_Set.ptp* that you downloaded from
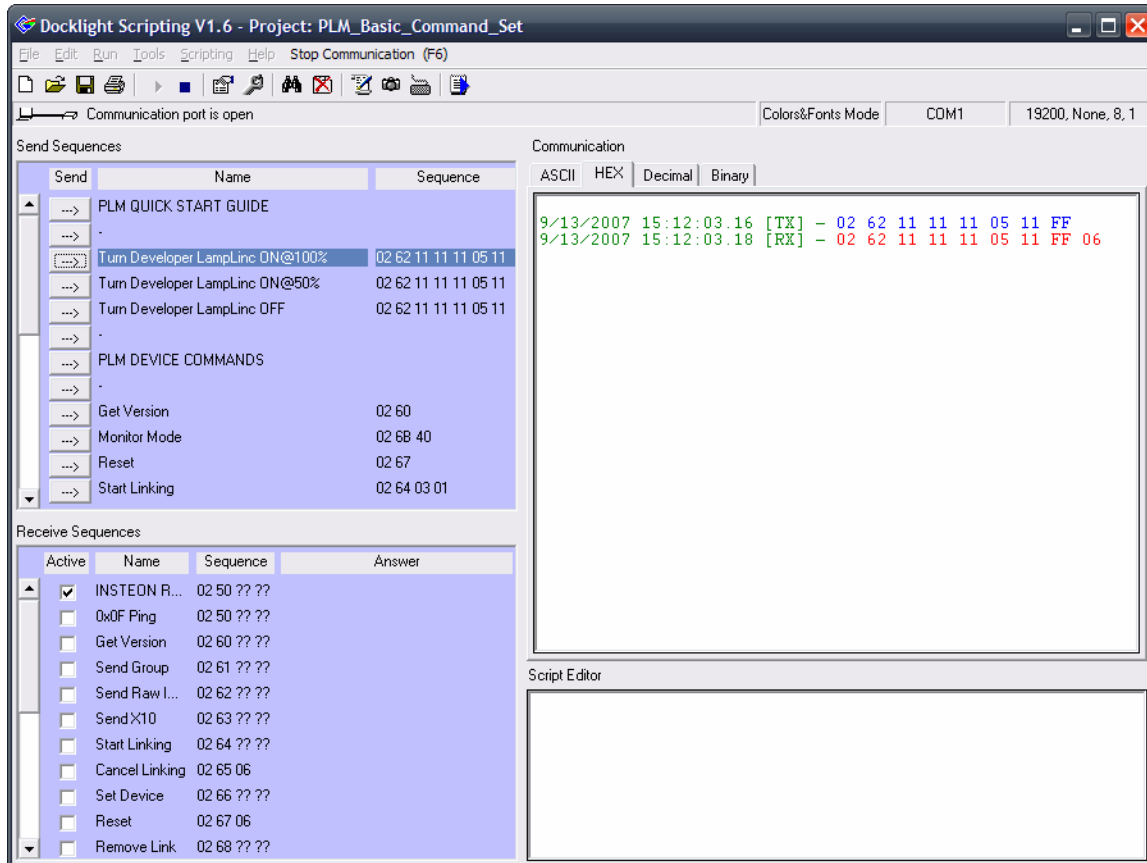http://code.insteon.net.
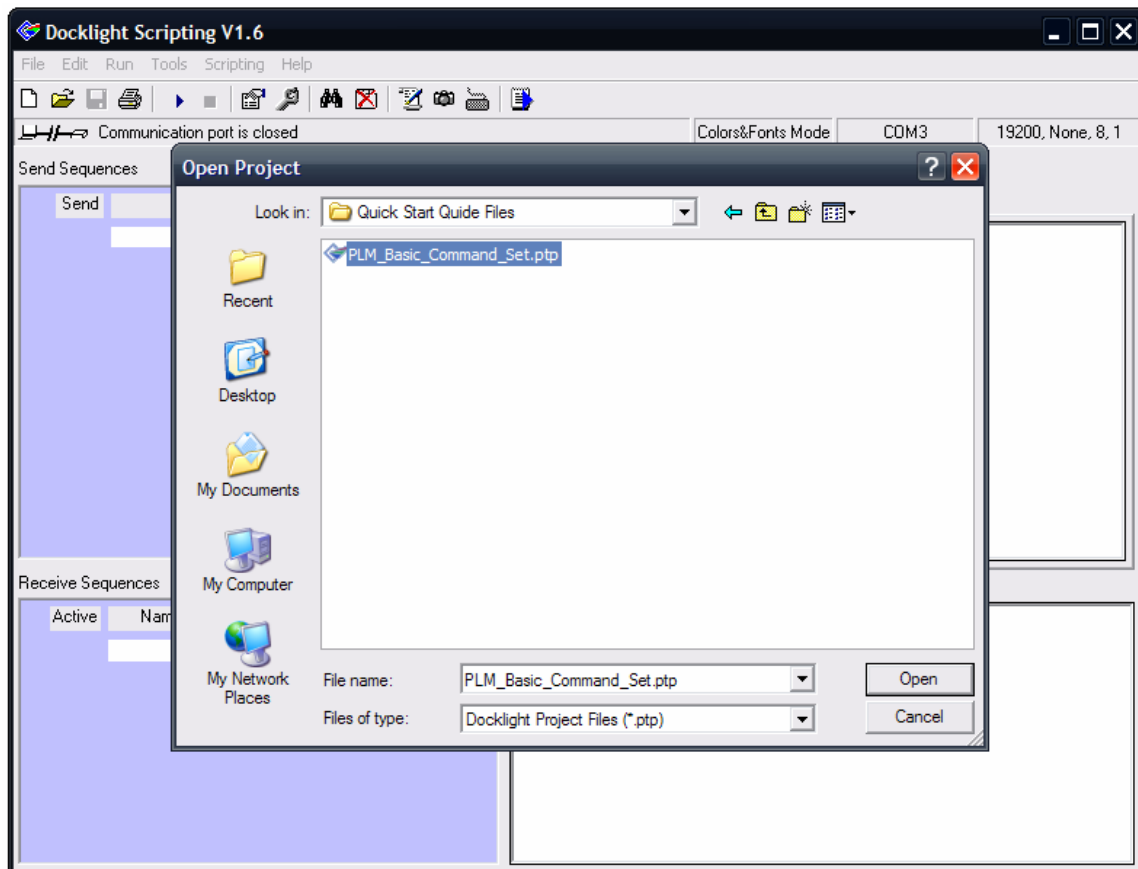
Step 5

Be sure the *HEX* tab is selected for the *Communication* window on the right.

Now, to turn on the light, press the *Send* button labeled *Turn Developer LampLinc ON@100%*.

Press the button labeled *Turn Developer LampLinc OFF* to turn the light off.

If you are able to control the light, you are now ready to experiment with other PLM commands, such as those given at the end of this *Quick Start Guide*. Please consult *IM Serial Commands*[9], for a complete catalog of all PLM commands. In-depth information about all aspects of INSTEON is available in the book-length *INSTEON Developer's Guide*, which you can also find at http://code.insteon.net.

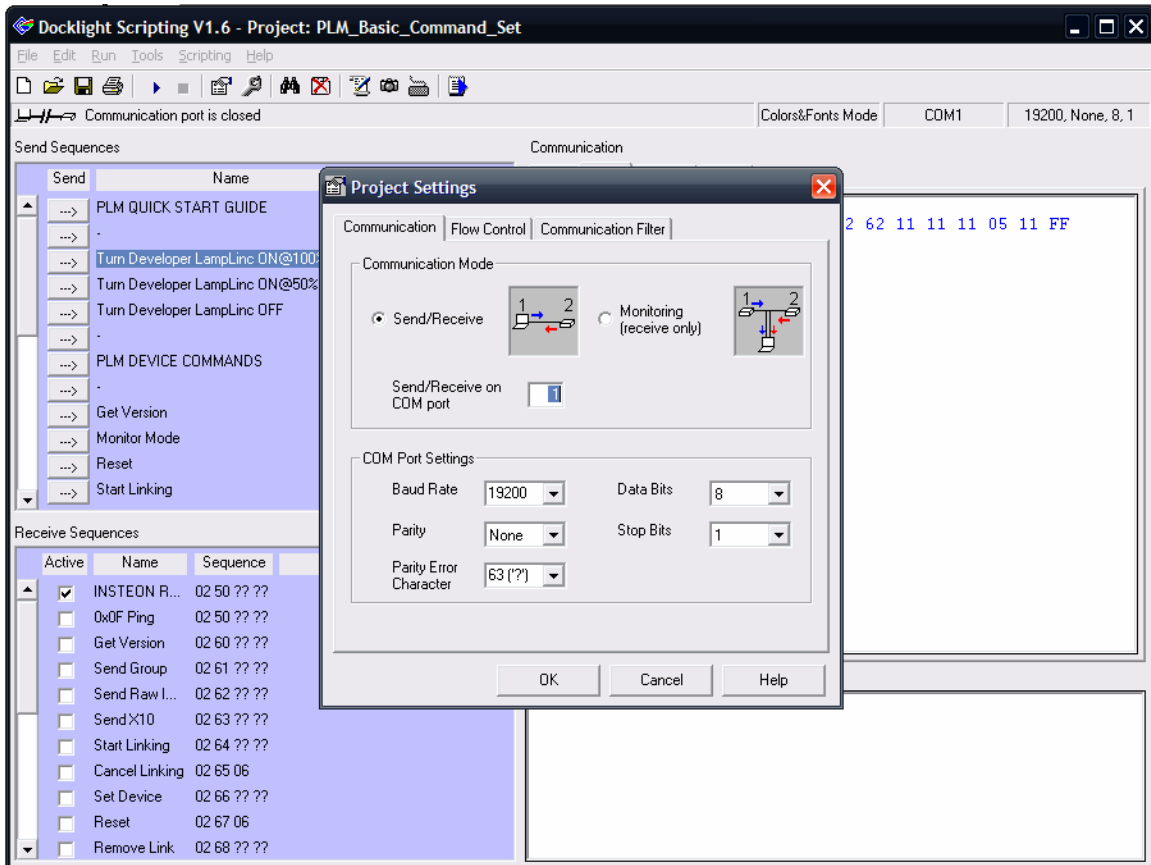If you were *not* successful in controlling the light, please proceed to *Step 6*[67].

Step 6

Try changing the COM Port for the PLM by selecting *Project Settings…* from the *Tools* menu.
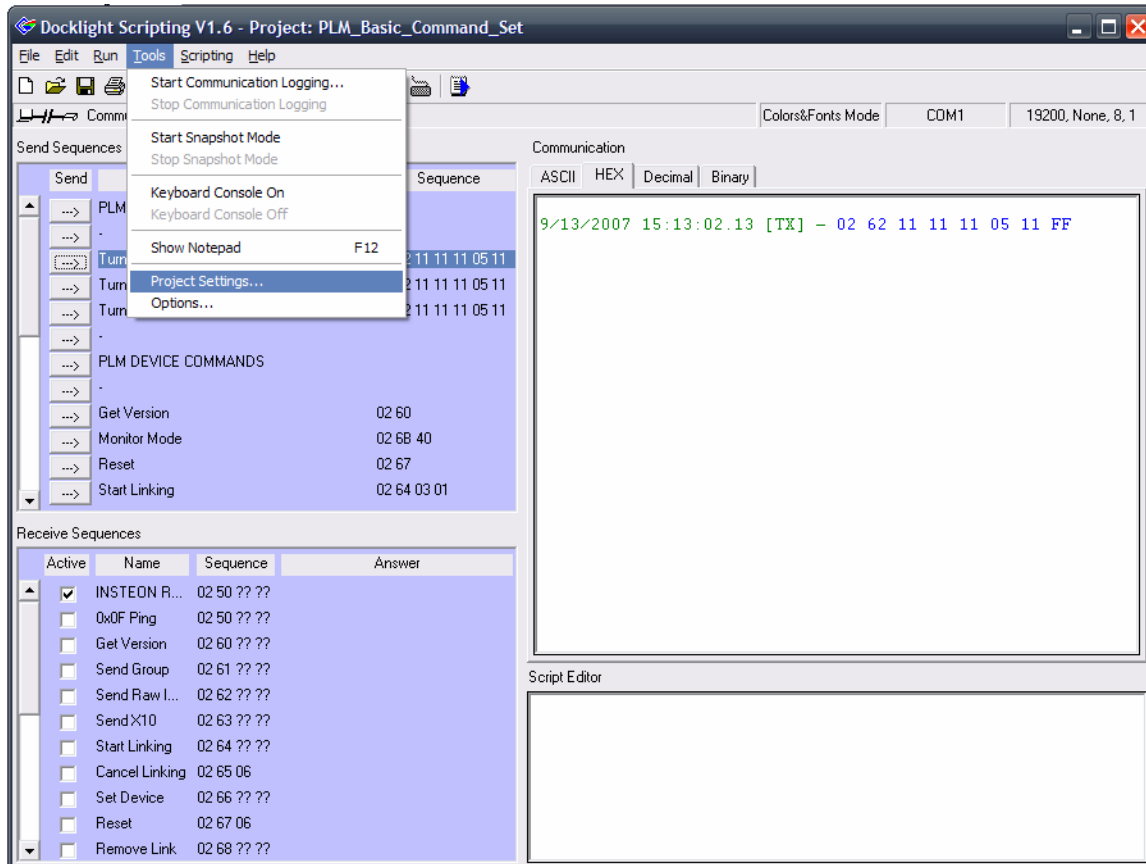
Step 7

Verify that the selected COM Port is connected to the PLM.

Repeat *Step 5*₆₆.

For further troubleshooting, check that the serial cable is connected correctly and that the light bulb is fully screwed in and working.

Examples of Other PLM Commands

For more detailed explanations and additional commands please see *IM Serial Commands*[9], above. In the following examples, square brackets contain the hexadecimal INSTEON Modem (IM) Command number along with the IM Command name as given above.

Get Version

[**0x60**, *Get IM Info*[45]]: Pings the PLM and confirms that communication is working properly. If the PLM is correctly connected to Docklight, the response will be returned right away.

```
[TX] - 02 60
[RX] - 02 60 AA AA AA 03 05 54 06
```

`AA AA AA` is the INSTEON ID of the PLM

`03 05`   is its Device Category

`54`   is its firmware revision

Monitor Mode

[**0x6B**, *Set IM Configuration*[30]]: Allows all messages from an originating device contained in the PLM's ALL-Link Database to be passed up to you.

```
[TX] - 02 6B 40
[RX] - 02 6B 40 00 00 06
```

Reset

[**0x67**, *Reset the IM*[40]]: Clears the PLM's ALL-Link Database

```
[TX] - 02 67
[RX] - 02 67 06
```

Start ALL-Linking

[**0x64**, *Start ALL-Linking*[30]]: Enters ALL-Linking mode for 4 minutes.

Send the *Start ALL-Linking* command. The first **0x01** is a flag denoting that the PLM is a Controller. The second **0x01** is the ALL-Link Group number.

```
[TX] - 02 64 01 01
[RX] - 02 64 01 01 06 02 53 01 01 11 11 11 01 00 22
```

Press and hold the SET button on the unit you want to link to the PLM. You will then be able to find the record of the link in the PLM's ALL-Link Database.

Start Unlinking

[**0x64**, *Start ALL-Linking*[30]]: Enters Unlinking mode for 4 minutes. The **0xFF** is a flag denoting unlinking mode. The **0x01** is the ALL-Link Group number.

```
[TX] - 02 64 FF 01
[RX] - 02 64 FF 01 06
```

Cancel Linking

[**0x65**, *Cancel ALL-Linking*[31]]: Exits ALL-Linking mode.

```
[TX] - 02 65
[RX] - 02 65 06
```

Get First Database Entry

[**0x69**, *Get First ALL-Link Record*33]: Returns the very first record in the PLM's ALL-Link Database in an **0x57** *ALL-Link Record Response* message.

```
[TX] - 02 69
[RX] - 02 69 06 02 57 E2 01 11 11 11 01 00 22
```

Get Next Database Entry

[**0x6A**, *Get Next ALL-Link Record*34]: Returns all the other records in the PLM's ALL-Link Database incrementally in a series of **0x57** *ALL-Link Record Response* messages. When there are no more records, you will receive a NAK (**0x15**).

```
[TX] - 02 6A
[RX] - 02 6A 06 02 57 A2 01 04 F7 EE 01 00 22
[TX] - 02 6A
[RX] - 02 6A 15
```

Group 1 On

[**0x61**, *Send ALL-Link Command*26]: Sends an INSTEON command to activate ALL-Link Group 1.

```
[TX] - 02 61 01 11 00
[RX] - 02 61 01 11 00 06
```

Group 1 Off

[**0x61**, *Send ALL-Link Command*26]: Sends an INSTEON command to deactivate ALL-Link Group 1.

```
[TX] - 02 61 01 13 00
[RX] - 02 61 01 13 00 06
```

Send X10

[**0x63**, *Send X10*24]: Sends an X10 message. For a list of available messages please see *X10 Message Handling*24.

```
[TX] - 02 63 01 00
[RX] - 02 63 01 00 06
```