LC-101 User Manual

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

Warning!

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademarks

Names are used for identification purposes only and may be registered trademarks of their respective companies.

Date: 2013/05/23

Table of Contents

Introduction

The LC-101 is an easy-to-use lighting control module that requires no specialist skills to install and operate, and no software is needed in order to control the Digital Output channel.

The LC-101 provides 1 channel for digital input (photocouple isolation) and 1 channel for relay output. The output channel is a Form A type relay, while the input channel is based on a sink-type using a wire connection. The input channel can be used to directly control a 1-channel relay ON and OFF sequence without requiring a remote host controller. 4 kV ESD protection and 5000 Vrms intra-module isolation are also provided.

When required, communication with the LC-101 is programmable based on either the DCON or the Modbus RTU protocol, and an added benefit is that different addresses can be set for DCON or Modbus RTU communication via hardware or software configuration.

1 Hardware Information

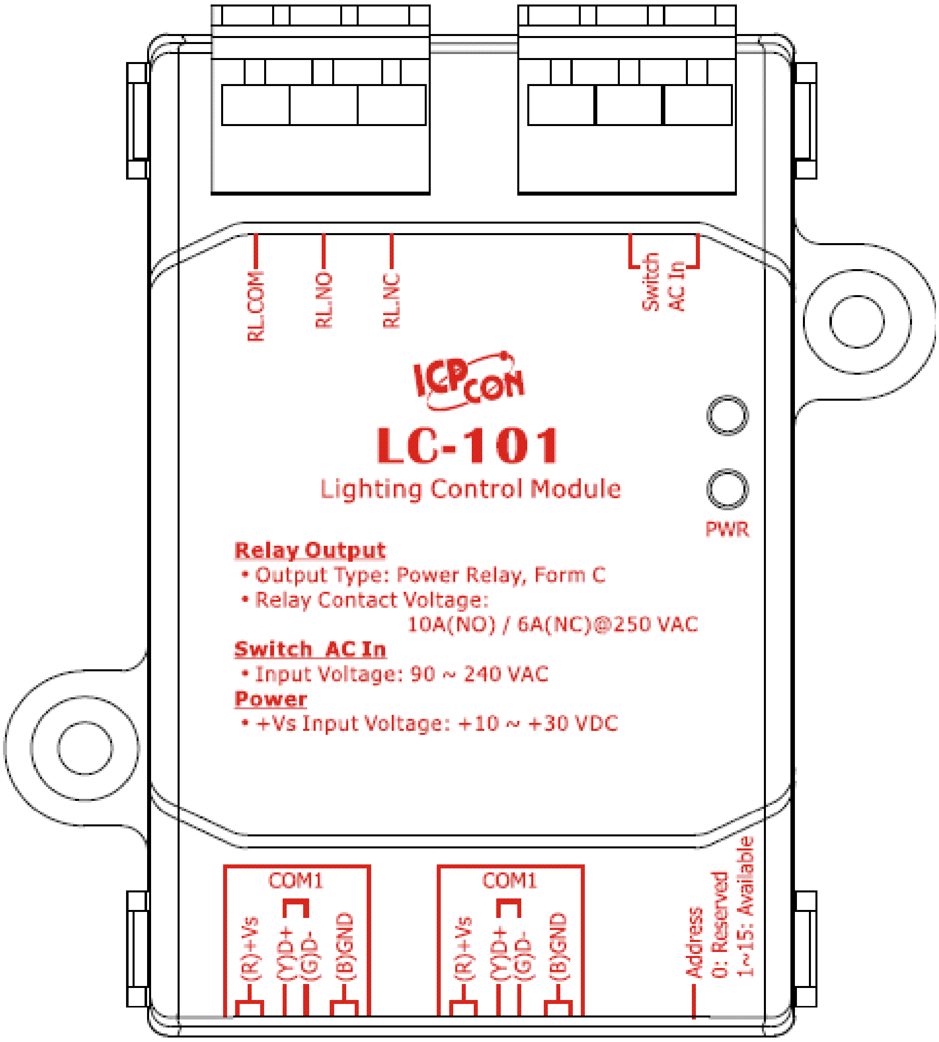
1.1 IO Specifications

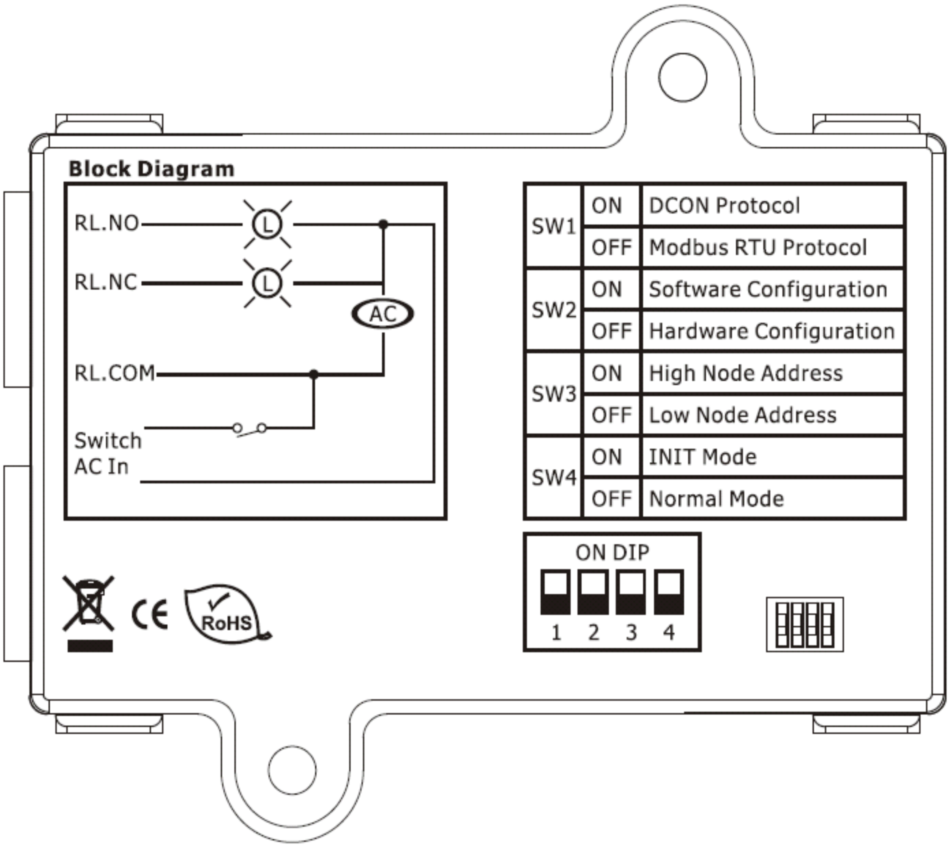
Digital Input	
Input Channels	1
Type	90-240 VAC
On Voltage Level	85 VAC
Off Voltage Level	60 VAC
Input Impedance	68 KΩ, 1 W
Isolation	5000 Vrms
Function	Local and Remote Direct Control Relay ON/OFF and Remote Status Monitoring
Relay Output	
Output Channels	1
Type	Power Relay, Form C
Operating Voltage	250 VAC or 30 VDC
Max. Load Current	10 A (NO) / 6A (NC) @ 250 VAC
Operating Time	15 ms Max.
Release Time	5 ms Max.
Electrical Life (Resistive load)	50,000 ops
Mechanical Life	1,000,000 ops at no load (300 ops/minute)
Safety Approval	UL/CUL, TÜV
Power-on Value	Yes

1.2 System Specifications

Communication	
Interface	RS-485
Data Format	N,8,1/O,8,1/E,8,1/N,8,2
Baud Rate	Hardware Configuration: Fixed 9600 bps
	Software Configuration: 1200-115200 bps
Protocol	Modbus RTU or DCON
Node Addresses	32-63 for hardware configuration or 0-255 for software configuration
Connector	RJ-11
LED Indicators	
Power	1 LED as Power Indicator
EMS Protection	
ESD (IEC 61000-4-2)	±2 kV Contact for Each Terminal
	±4 kV Air for Random Point
EFT (IEC 61000-4-4)	±2 kV for Power
Power Requirements	
Input Voltage Range	+10 - +30 VDC
Consumption	0.5 W Max.
Connector	RJ-11
Mechanical	
Dimensions (W x L x H)	52 mm x 98 mm x 27 mm
Installation	Screw Mounting
Environment	
Operating Temperature	-25°C - +75°C
Storage Temperature	-30°C - +75°C
Humidity	10 - 95% RH, Non-condensing

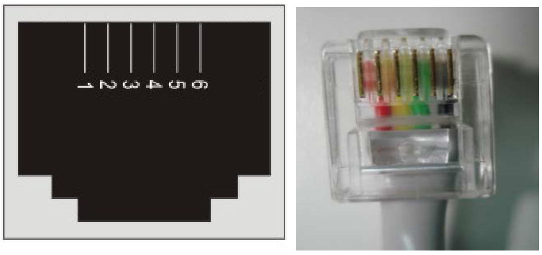
1.3 Pin Assignments





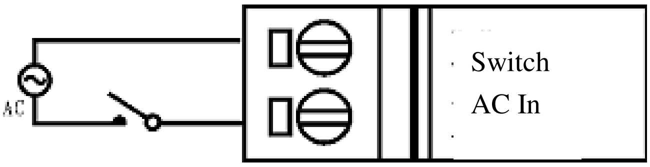
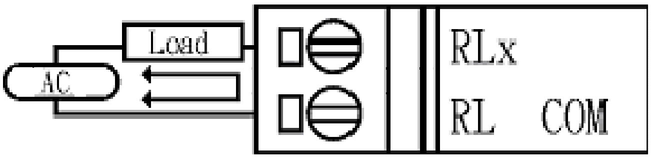
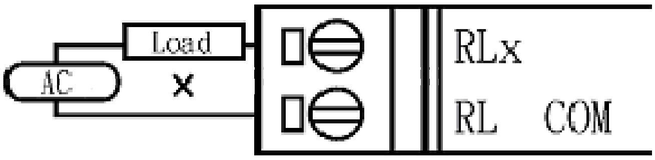
SW1	ON	DCON Protocol
	OFF	Modbus RTU Protocol
SW2	ON	Software Configuration
	OFF	Hardware Configuration
SW3	ON	High Node Address
	OFF	Low Node Address
SW4	ON	INIT Mode
	OFF	Normal Mode

RJ-11 Connector

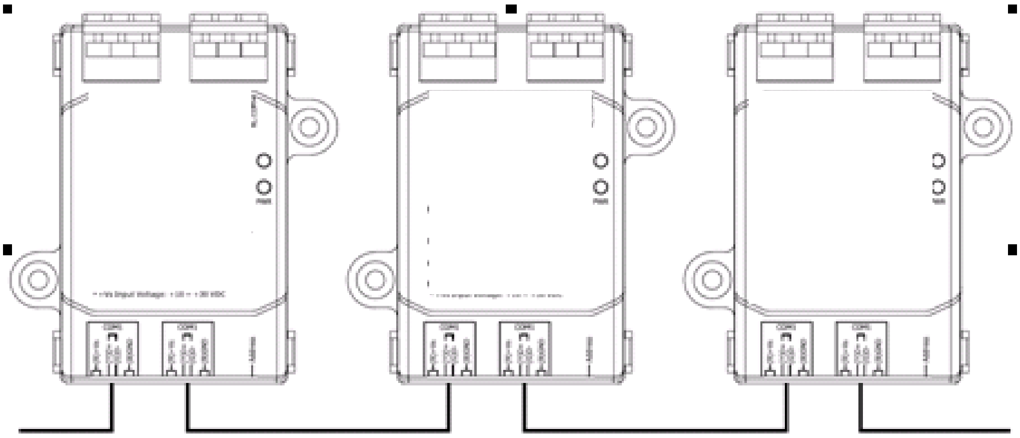


1.4 Wire Connections

DIO Wire Connections

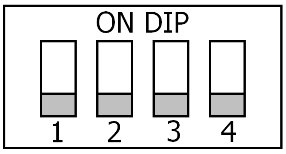
DI

Relay Output ON

Relay Output OFF


Power and Communication

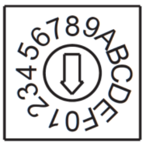
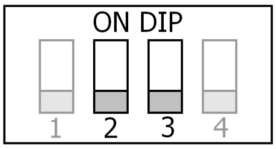
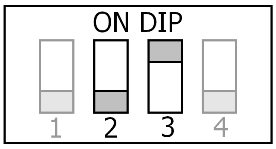


RS-485 and power input daisy chain using RJ-11 connectors

1.5 DIP Switch Configuration

	SW1	ON	DCON Protocol
		OFF	Modbus RTU Protocol
	SW2	ON	Software Configuration
		OFF	Hardware Configuration
	SW3	ON	High Node Address
		OFF	Low Node Address
	SW4	ON	INIT Mode
		OFF	Normal Mode

Address Settings via Hardware Configuration

		0 - F for Address 32 - 47 (Low Node Address)
		0 - F for Address 48 - 63 (High Node Address)

1.5.1 INIT Mode

When the LC-101 is powered on with SW4 in the ON position, the module will be set to INIT Mode. In this mode, the position of the SW1-SW3 and Address switches will be ignored and the LC-101 will use the fixed configuration listed below.

Protocol:	DCON
Address:	32 (0x20)
Baud Rate:	9600 bps
Data Format:	N,8,1

In this mode, the relevant commands can be used to change the configuration, and the new settings will be saved to the EEPROM.

1.5.2 Hardware Configuration Mode

When the LC-101 is powered on with both SW4 and SW2 in the OFF position, the module will be set to Hardware Configuration Mode. In this mode, the following configuration is used.

Protocol:	Dependent on the position of SW1
Address:	Refer to the “Address Settings via Hardware Configuration” table above
Baud Rate:	Fixed at 9600 bps
Data Format:	Fixed to N,8,1

In this mode, any software command related to configuration will be ignored when using the Modbus RTU protocol, or will return an error when using the DCON protocol.

1.5.3 Software Configuration Mode

When the LC-101 is powered on with SW4 in the OFF position and SW2 in the ON position, the module will be set to Software Configuration Mode. In this mode, the configurations will be retrieved from the EEPROM. The default configuration stored in the EEPROM is:

Protocol:	Modbus RTU
Address:	32 (0x20)
Baud Rate:	9600 bps
Data Format:	N,8,1

In this mode, the relevant commands can be used to change the configuration, and the new settings will be saved to the EEPROM.

1.6 Software Configuration Tables

Baud Rate Settings (CC)

Code	03	04	05	06	07	08	09	0A
Baud Rate	1200	2400	4800	9600	19200	38400	57600	115200

Type Settings (TT)

For the LC-101, the type code is fixed to 40.

Data Format Settings (FF)

7	6	5	4	3	2	1	0
CU	CS	Reserved					

Key	Description
CS	Checksum Settings 0: Disabled 1: Enabled
CU	Counter Update: 0: The counter is updated when there is a falling edge in the input signal. 1: The counter is updated when

Note: All Reserved bits should be zero.

1.7 Digital Input/Output Data Format for the DCON Protocol

The data format for the response to the **$AA4**, **$AA6** and **$AALS** commands is:

(First Value)(Second Value)00

The data format for the response to the **@AA** command is:

(First Value)(Second Value)

Note: Both the First Value and the Second Value are in the form of two hexadecimal digits.

Module	First Value		Second Value	
LC-101	DO0	00 - 01	DI0	00 - 01

2 DCON Protocol

All communication with the LC-101 consists of commands generated by the Host and responses transmitted by the LC-101 module. Each module has a unique ID number that is used for addressing purposes and is stored in non-volatile memory. The module ID number is set to 01 by default and can be changed by sending a user command. All commands to the modules contain the ID number as the address, meaning that only the addressed module will respond.

Command Format:

Delimiter Character	Module Address	Command	Checksum	CR

Response Format:

Delimiter Character	Module Address	Data	Checksum	CR

CR = End of command character, carriage return (0x0D), used to end a frame.

Note: All characters should be in upper case.

An Overview of the DCON Command Set

General Command Sets			
Command	**Response**	**Description**	**Section**
%AANNTTCCFF	!AA	Sets the Configuration of the Module	2.1
#**	No Response	Sends the Synchronized Sampling Command	2.2
#AA00(Data)	>	Sets the Value for all Digital Output Channels	2.3
#AA0A(Data)	>	Sets the Value for all Digital Output Channels	2.4
#AA10DD	>	Sets the Digital Output for a Single Channel	2.5
#AAA0DD	>	Sets the Digital Output for a Single Channel	2.6
#AAN	!AA(Data)	Reads the Digital Input Counter Value for a Specific Channel	2.7
$AA2	!AANNTTCCFF	Reads the Configuration of the Module	2.8
$AA4	!S(Data)	Reads the Synchronized Data	2.9
$AA5	!AAS	Reads the Reset Status of the Module	2.10
$AA6	!(Data)	Reads the Status of the Digital Input/Output Channels	2.11
$AAC	!AA	Clears the Status of the Latched Digital Input Channels	2.12
$AACN	!AA	Clears the Digital Input Counter	2.13
$AAF	!AA(Data)	Reads the Firmware Version of the Module	2.14
$AALS	!(Data)	Reads the Latched Digital Input Status	2.15
$AAM	!AA(Data)	Reads the Name of the Module	2.16
$AAP	!AASC	Reads the Communication Protocol	2.17
$AAPN	!AA	Sets the Communication Protocol	2.18
@AA	>(Data)	Reads the Status of the Digital Input/Output Channels	2.19
@AA(Data)	>	Sets the Value for all Digital Output Channels	2.20

2.1 %AANNTTCCFF

Description:

This command is used to set the configuration of a specified module.

Syntax:

%AANNTTCCFF[CHKSUM](CR)

%	Delimiter character
AA	The address of the module to be configured in hexadecimal format (00 to FF)
NN	The new address of the module in hexadecimal format (00 to FF)
TT	Type code, which should be set to 40 for DIO modules
CC	The new Baud Rate, see Section 1.6 for details. The INIT* pin must be connected to the ground pin in order to change Baud Rates. For modules using frame ground, this is achieved by moving the rear slide switch to the INIT position.
FF	The command used to set the counter update direction and the checksum (see Section 1.6). The INIT* pin must be connected to the ground pin in order to change the checksum settings. For modules using frame ground, this is achieved by moving the rear slide switch to the INIT position.

Response:

Valid Command:	**!AA[CHKSUM](CR)**
Invalid Command:	**?AA[CHKSUM](CR)**

!	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid. If an attempt is made to change the **Baud Rate** or **Checksum** settings without first connecting the INIT* pin to the ground pin or without switching the rear slide switch to the INIT position, the module will return a response indicating that the command was invalid.
AA	The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: %0102400600 Response: !02

Changes the address of module 01 to 02. The module returns a response indicating that the command was valid and includes the new address of the module.

Command: %0101200A00 Response: ?01

Changes the Baud Rate of module 01 to 115200bps. The module returns an invalid command, because it is not in INIT* mode.

Command: %0101200A00 Response: !01

Changes the Baud Rate of module 01 to 115200bps and the module is in INIT* mode. The module returns a valid response.

Related Commands:

Section 2.8 $AA2

Related Topics:

Section 1.5 Software Configuration Tables

Notes:

Changes to the address and counter update direction settings take effect immediately after a valid command is received. Changes to the Baud Rate and checksum settings take effect on the next power-on reset.

2.2 #**

Description:

This command is used to instruct all modules to sample their input values and store the data for later retrieval.

Syntax:

#[CHKSUM](CR)**

#	Delimiter character
**	The Synchronized sampling command

Response:

There is no response to this command. To access the data, another command, $AA4, must be sent. See Section 2.9 for details.

Examples:

Command: #** Response: There is no response to this command.

Sends the synchronized sampling command to all modules.

Command: $014 Response: !10F0000

Reads the synchronized sampling data and the module returns a response indicating that the command was valid. The status byte of the response is 1, which means that it is the first time the synchronized sampling data has been read since the previous #** command was received.

Command: $014 Response: !00F0000

Reads the synchronized sampling data and the module returns a response indicating that the command was valid. The status byte of the response is 0, which means that it is NOT the first time the synchronized sampling data has been read since the previous #** command was received.

Related Commands:

Section 2.9 $AA4

2.3 #AA00(Data)

Description:

This command is used to set the digital output value for Channel RL1 of a specified module.

Syntax:

#AA00(Data)[CHKSUM](CR)

Delimiter character

AA The address of the module to be set in hexadecimal format (00 to FF)

00 The command to set the digital output value for Channel RL1

(Data) A two-digit hexadecimal value, where bit 0 corresponds to channel RL1. When the bit is 0, it denotes that the digital output channel is set to OFF, and 1 denotes that the digital output channel is set to ON.

Response:

Valid Command: **>[CHKSUM](CR)**

Invalid Command: **?AA [CHKSUM](CR)**

Ignored Command: **![CHKSUM](CR)**

> Delimiter character to indicate that the command was valid

? Delimiter character to indicate that the command was invalid

! Delimiter character to indicate that the command was ignored. The command will be ignored if a Host Watchdog timeout has occurred.

AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: #020001 Response: >

Sets Channel RL1 of module 02 to ON, and the module returns a response indicating that the command was valid.

Command: #020002 Response: ?AA

Attempts to set Channel RL2 of module 02 to ON, but the module returns a response indicating that the command was invalid because Channel RL2 does not exist.

Command: #020A02 Response: !

Attempts to set Channel RL2 of module 02 to ON, but the module returns a response indicating that the command was ignored because a Host Watchdog timeout has occurred.

Related Commands:

Section 2.4 #AA0A(Data), Section 2.5 #AA10DD, Section 2.6 #AAA0DD

Related Topics:

Section 1.6 Software Configuration Tables

2.4 #AA0A(Data)

Description:

This command is used to set the digital output value for Channel RL1 of a specified module.

Syntax:

#AA0A(Data)[CHKSUM](CR)

Delimiter character

AA The address of the module to be set in hexadecimal format (00 to FF)

0A The command to set the digital output value for Channel RL1

(Data) A two-digit hexadecimal value, where bit 0 corresponds to Channel RL1. When the bit is 0, it denotes that the digital output channel is set to OFF, and 1 denotes that the digital output channel is set to ON.

Response:

Valid Command: **>[CHKSUM](CR)**

Invalid Command: **?AA [CHKSUM](CR)**

Ignored Command: **![CHKSUM](CR)**

> Delimiter character to indicate that the command was valid

? Delimiter character to indicate that the command was invalid

! Delimiter character to indicate the command was ignored. The command will be ignored if a Host Watchdog timeout has occurred.

AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: #020A01 Response: >

Sets Channel RL1 of module 02 to ON, and the module returns a response indicating that the command was valid.

Command: #020A02 Response: ?AA

Attempts to set Channel RL2 of module 02 to ON, but the module returns a response indicating that the command was invalid because Channel RL2 does not exist.

Command: #020A01 Response: !

Attempts to set Channel RL1 of module 02 to ON, but the module returns a response indicating that the command was ignored because a Host Watchdog timeout has occurred.

Related Commands:

Section 2.3 #AA00(Data), Section 2.5 #AA10DD, Section 2.6 #AAA0DD

Related Topics:

Section 1.6 Software Configuration Tables

2.5 #AA10DD

Description:

This command is used to set the digital output value for a single channel of a specified module.

Syntax:

#AA10DD[CHKSUM](CR)

#	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
1	The command to set the digital output for a single channel
0	Specifies the relay output channel to be set, zero based
DD	The command to set the relay output channel: 00: Sets the relay output channel to OFF 01: Sets the relay output channel to ON

Response:

Valid Command:	**>[CHKSUM](CR)**
Invalid Command:	**?AA [CHKSUM](CR)**
Ignored Command:	**![CHKSUM](CR)**

>	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid
!	Delimiter character to indicate the command was ignored. The command will be ignored if a Host Watchdog timeout has occurred.
AA	The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: #021001 Response: >

Sets Channel RL1 of module 02 to ON, and the module returns a response indicating that the command was valid.

Command: #021101 Response: ?AA

Attempts to set Channel RL2 of module 02 to ON, but the module returns a response indicating that the command was invalid because Channel RL2 does not exist.

Command: #021001 Response: !

Attempts to set Channel RL1 of module 02 to ON, but the module returns a response indicating that the command was ignored because a Host Watchdog timeout has occurred.

Related Commands:

Section 2.3 #AA00(Data), Section 2.4 #AA0A(Data), Section 2.6 #AAA0DD

Related Topics:

Section 1.6 Software Configuration Tables

2.6 #AAA0DD

Description:

This command is used to set the digital output value for a single channel of a specified module.

Syntax:

#AAA0DD[CHKSUM](CR)

#	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
A	The command to set the digital output value for a single channel
0	Specifies the relay output channel to be set, zero based
DD	The command to set the relay output channel: 00: Sets the relay output channel to OFF. 01: Sets the relay output channel to ON.

Response:

Valid Command:	**>[CHKSUM](CR)**
Invalid Command:	**?AA [CHKSUM](CR)**
Ignored Command:	**![CHKSUM](CR)**

>	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid
!	Delimiter character to indicate the command was ignored. The command will be ignored if a Host Watchdog timeout has occurred.
AA	The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: #02A001 Response: >

Sets Channel RL1 of module 02 to ON, and the module returns a response indicating that the command was valid.

Command: #02A101 Response: ?AA

Attempts to set Channel RL2 of module 02 to ON, but the module returns a response indicating that the command was invalid because Channel RL2 does not exist.

Command: #02A001 Response: !

Attempts to set Channel RL1 of module 02 to ON, but the module returns a response indicating that the command was ignored because a Host Watchdog timeout has occurred.

Related Commands:

Section 2.3 #AA00(Data), Section 2.4 #AA0A(Data), Section 2.5 #AA10DD

Related Topics:

Section 1.6 Software Configuration Tables

2.7 #AAN

Description:

This command is used to read the digital input counter value from a specific channel of a specified module.

Syntax:

#AAN[CHKSUM](CR)

#	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
N	The channel to be read, zero based

Response:

Valid Command: **!AA(Data)[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!	Delimiter character to indicate that the command was valid.
?	Delimiter character to indicate that the command was invalid.
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	Five decimal digits representing the digital input counter data from the specified channel (00000 to 65535)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: #030 Response: !0300103

Reads the digital input counter value from Channel 0 of module 03 and returns a response indicating that the command was valid, with a counter value of 103.

Command: #029 Response: ?02

Attempts to read the digital input counter value from Channel 9 of module 02, but the module returns a response indicating that the command was invalid because Channel

9 does not exist.

Related Commands:

Section 2.13 #AACN

2.8 $AA2

Description:

This command is used to read the configuration of a specified module.

Syntax:

$AA2[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
2	The command to read the configuration of the module

Response:

Valid Command: **!AATTCCFF[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid
AA	The address of the responding module in hexadecimal format (00 to FF)
TT	The Type code for the module, which should be 40 for DIO modules
CC	The Baud Rate for the module. See Section 1.6 for details.
FF	The checksum and counter update direction settings of the module. See Section 1.6 for details.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $012 Response: !01400600

Reads the configuration of module 01 and returns a response indicating that the command was valid, and showing that the Type code is set to 40, the Baud Rate is 9600 bps, the Checksum is Disabled and the counter update direction is Falling Edge.

Command: $032 Response: ?03

Attempts to read the configuration of module 03, but returns a response indicating that the command was invalid because module 03 does not exist.

註解 [DK1]: There would be no response if the address is incorrect?

Related Commands:

Section 2.1 %AANNTTCCFF

Related Topics:

Section 1.5 Dip Switch Configuration, Section 1.6 Software Configuration Tables

2.9 $AA4

Description:

This command is used to read the synchronized sampling data that was stored when the last #** command was sent.

Syntax:

$AA4[CHKSUM](CR)

$ Delimiter character

AA The address of the module to be read in hexadecimal format (00 to FF)

4 The command to read the synchronized sampling data

Response:

Valid Command: **!S(Data)[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter character to indicate that the command was valid

? Delimiter character to indicate that the command was invalid

AA The address of the responding module in hexadecimal format (00 to FF)

S The status of the synchronized sampling data:

0: This is **NOT** the first time that the data has been read

1: This is the first time that the data has been read

(Data) The synchronized sampling data. See Section 1.6 for details of the data format.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $014 Response: ?01

Attempts to read the synchronized sampling data for module 01, but returns a response indicating that the command was invalid because the Synchronized Sampling Command, #**, was not sent in advance.

Command: #**　　Response: There is no response to this command.

Sends the synchronized sampling command to all modules.

Command: $014　　Response: !1000F00

Reads the synchronized sampling data for module 01. The module returns a response indicating that the command was valid containing the synchronized sampling data, and sets the status byte to 1 to signify that this is the first time that the synchronized sampling data has been read.

Command: $014　　Response: !0000F00

Reads the synchronized sampling data for module 01. The module returns a response indicating that the command was valid containing the synchronized sampling data, and sets the status byte to 0 to signify that the synchronized sampling data has been read.

Command: $034　　Response: ?03

Attempts to read the synchronized sampling data for module 03, but returns a response indicating that the command was invalid because module 03 does not exist.

註解 [DK2]: There would be no response if the address is incorrect?

Related Commands:

Section 2.2 #**

Related Topics:

Section 1.6 Software Configuration Tables

2.10 $AA5

Description:

This command is used to read the reset status for a specified module.

Syntax:

$AA5[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
5	The command to read the reset status of the module

Response:

Valid Command:	**!AAS[CHKSUM](CR)**
Invalid Command:	**?AA[CHKSUM](CR)**

! Delimiter character to indicate that the command was valid

? Delimiter character to indicate that the command was invalid

AA The address of the responding module in hexadecimal format (00 to FF)

S The reset status of the module:

0: This is **NOT** the first time the command has been sent since the module was powered on, which denotes that there has been no module reset since the last $AA5 command was sent.

1: This is the first time the $AA5 command has been sent since the module was powered on.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $015 Response: !011

Reads the reset status for module 01 and returns a response indicating that the command was valid, and that it is the first time the $AA5 command has been sent since the module was powered on.

Command: $015 Response: !010

Reads the reset status for module 01 and returns a response indicating that the command was valid, and that there has been no module reset since the last $AA5 command was sent.

Command: $035 Response: ?03

Attempts to read the reset status for module 03, but returns a response indicating that the command was invalid because module 03 does not exist.

2.11 $AA6

Description:

This command is used to read the status of both the digital input and digital output channels of a specified module.

Syntax:

$AA6[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
6	The command to read the status of the digital input and digital output channels

Response:

Valid Command:	**!(Data)[CHKSUM](CR)**
Invalid Command:	**?AA[CHKSUM](CR)**

!	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The status of the digital output and digital input channels represented by a four-digit hexadecimal value followed by 00. The first two digits represent the status of the digital output channels and the second two represent the status of the digital input channels. See Section 1.7 for more details.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $026 Response: !010100

Reads the status of the digital output and digital input channels for module 02 and returns a response indicating that the command was valid and that the current digital output value is 01 and the current digital input value is 01 denoting that both the digital output and digital input channels are ON.

Command: $036 Response: ?03

Attempts to read the status of the digital output and digital input channels for module 03 and returns a response indicating that the command was invalid because module 03 does not exist.

註解 [DK3]: There would be no response if the address is incorrect?

Related Commands:

Section 2.19 @AA

Related Topics:

Section 1.6 Software Configuration Tables, Section 1.7 Digital Input/Output Data Format Settings

2.12 $AAC

Description:

This command is used to clear the status of the latched digital output and digital input channels of a specified module.

Syntax:

$AAC[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be cleared in hexadecimal format (00 to FF)
C	The command to clear the status of the latched digital output and digital input channels

Response:

Valid Command:	**!AA[CHKSUM](CR)**
Invalid Command:	**?AA[CHKSUM](CR)**

!	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid
AA	The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $01L0 Response: !010100

Reads the status of the low latched digital output and digital input channels of module 01 and returns a response indicating that the command was valid, with a value of 0101, which denotes that the latched values for the both the digital output and digital input channels have recently been set to ON.

Command: $01C Response: !01

Clears the status of the latched digital input channels of module 01 and returns a response indicating that the command was valid.

Command: $01L0 Response: !000000

Reads the status of the low latched digital output and digital input channels of module 01 and returns a response indicating that the command was valid, with a value of 0000, which denotes that the status of all low latched digital output and digital input channels has been cleared.

Command: $01C Response: ?03

Attempts to clear the status of the latched digital input channels of module 03 and returns a response indicating that the command was invalid because module 03 does not exist.

註解 [DK4]: There would be no response if the address is incorrect?

Related Commands:

Section 2.15 $AALS

Related Topics:

None

Notes:

The status of both the low and the high latched digital output and digital input channels will be cleared when using this command.

2.13 $AAC0

Description:

This command is used to clear the digital input counter for a specific channel of a specified module.

Syntax:

$AACN[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be cleared in hexadecimal format (00 to FF)
C	The command to clear the digital input counter
0	The channel to be cleared, zero based

Response:

Valid Command:	**!AA[CHKSUM](CR)**
Invalid Command:	**?AA[CHKSUM](CR)**

!	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid
AA	The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: #030 Response: !0300103

Reads the counter data from channel 1 of module 03 and returns a response indicating that the command was valid, and that the counter value is 103.

Command: $03C0 Response: !03

Clears the counter value for channel 1 of module 03 and returns a response indicating that the command was valid.

Command: #032 Response: !0300003

Reads the counter data from channel 2 of module 03 and returns a response indicating that the command was valid, and that the counter value is 3.

Command: #039 Response: ?03

Attempts to read the counter data from channel 9 of module 03 and returns a response indicating that the command was invalid because channel 9 does not exist.

Related Commands:

Section 2.7 #AAN

2.14 $AAF

Description:

This command is used to read the firmware version of a specified module.

Syntax:

$AAF[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
F	The command to read the firmware version

Response:

Valid Command:	**!AA(Data)[CHKSUM](CR)**
Invalid Command:	**?AA[CHKSUM](CR)**

!	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	A string indicating the firmware version of the module

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $01F Response: !0101.00

Reads the firmware version of module 01, and returns a response indicating that the command was valid, and that the firmware is version 01.00.

Command: $03F Response: ?03

Attempts to read the firmware version of module 03 and returns a response indicating that the command was invalid because module 03 does not exist.

2.15 $AALS

Description:

This command is used to read the status of the latched digital output and digital input channels of a specified module.

Syntax:

$AALS[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
L	The command to read the status of the latched channels
S	The status to be read: 0: Reads the status of the low latched channels 1: Reads the status of the high latched channels

Response:

Valid Command: **!(Data)[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

!	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The status of the latched digital output and digital input channels, represented by a four-digit hexadecimal value followed by 00. See Section 1.6 for details.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $01L0 Response: !010100

Reads the status of the low latched digital output and digital input channels of module 01 and returns a response indicating that the command was valid, with a value of 0101 denoting that the latched values for the both the digital output and digital input channels have recently been set to ON.

Command: $01C　　Response: !01

Clears the status of the latched digital output and digital input channels of module 01 and returns a response indicating that the command was valid.

Command: $01L0　　Response: !000000

Reads the status of the low latched digital output and digital input channels of module 01 and returns a response indicating that the command was valid, with a value of 0000 denoting that the latched value has recently been set to ON .

Command: #03C　　Response: ?03

Attempts to clear the status of the latched digital input channels of module 03 and returns a response indicating that the command was invalid because module 03 does not exist.

註解 [DK5]: There would be no response if the address is incorrect?

Related Commands:

Section 2.12 $AAC

Related Topics:

Section 1.6 Software Configuration Tables

2.16 $AAM

Description:

This command is used to read the name of a specified module.

Syntax:

$AAM[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
M	The command to read the name of the module

Response:

Valid Command:	**!AA(Data)[CHKSUM](CR)**
Invalid Command:	**?AA[CHKSUM](CR)**

!	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	A string indicating the name of the module

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $02M Response: !02LC101

Reads the name of module 02 and returns a response indicating that the command was valid, and that the name of the module is "LC-101".

Command: $03M Response: ?03

Attempts to read the name of module 03 and returns a response indicating that the command was invalid because module 03 does not exist.

註解 [DK6]: There would be no response if the address is incorrect?

2.17 $AAP

Description:

This command is used to read the communication protocol information for a specified module.

Syntax:

$AAP[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
P	The command to read the communication protocol information

Response:

Valid Command: **!AASC[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

! Delimiter character to indicate that the command was valid

? Delimiter character to indicate that the command was invalid

AA The address of the responding module in hexadecimal format (00 to FF)

S The protocols supported by the module:

0: Only the DCON protocol is supported

1: Both the DCON and Modbus RTU protocols are supported

C The current protocol that is saved in the EEPROM that will be used at the next power-on reset:

0: The protocol saved in the EEPROM is DCON

1: The protocol saved in the EEPROM is Modbus RTU

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $01P Response: !0110

Reads the communication protocol information for module 01 returns a response indicating that the command was valid, with a value of 10, which denotes that the module supports both the DCON and Modbus RTU protocols and that the protocol that will be used at the next power-on reset is DCON.

Command: $03P   Response: ?03

Attempts to read the communication protocol information for module 03 and returns a response indicating that the command was invalid because module 03 does not exist.

註解 [DK7]: There would be no response if the address is incorrect?

Related Commands:

Section 2.18 $AAPN

Related Topics:

Section 1.5 Dip Switch Configuration

2.18 $AAPN

Description:

This command is used to set the communication protocol for a specified module.

Syntax:

$AAPN[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
P	The command to set the communication protocol
N	The protocol to be used: 0: DCON Protocol 1: Modbus RTU Protocol

Note that before using this command, the rear slide switch must be in the INIT position. The new protocol information will be saved in the EEPROM and will become effective after the next power-on reset.

Response:

Valid Command:	**!AA[CHKSUM](CR)**
Invalid Command:	**?AA[CHKSUM](CR)**

!	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid
AA	The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: $01P1 Response: ?01

Attempts to set the communication protocol to be used for module 01 to Modbus RTU, but returns a response indicating that the command was invalid because the module is not in INIT mode.

Command: $01P1 Response: !01

Sets the communication protocol to be used for module 01 to Modbus RTU and returns a response indicating that the command was valid.

Related Commands:

Section 2.17 $AAP

Related Topics:

Section 1.5 Dip Switch Configuration

2.19 @AA

Description:

This command is used to read the status of both the digital output and digital input channels of a specified module.

Syntax:

@AA[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)

Response:

Valid Command:	**>(Data)[CHKSUM](CR)**
Invalid Command:	**?AA[CHKSUM](CR)**

>	Delimiter character to indicate that the command was valid
?	Delimiter character to indicate that the command was invalid
AA	The address of the responding module in hexadecimal format (00 to FF)
(Data)	The status of the digital output and digital input channels represented by a four-digit hexadecimal value. The first two digits represent the status of the digital output channels and the second two represent the status of the digital input channels. See Section 1.7 for more details.

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: @02 Response: >0101

Reads the status of the digital output and digital input channels for module 02 and returns a response indicating that the command was valid, and that the current digital output value is 01 and the current digital input value is 01 denoting that both the digital output and digital input channels are ON..

Command: @03 Response: ?03

Attempts to read the status of the digital output and digital input channels for module 03 and returns a response indicating that the command was invalid because module 03 does not exist.

註解 [DK8]: There would be no response if the address is incorrect?

Related Commands:

Section 2.11 $AA6, Section 2.20 @AA(Data)

Related Topics:

Section 1.7 Digital Input/Output Data Format Settings

2.20 @AA(Data)

Description:

This command is used to set the value for all digital output channels of a specified module.

註解 [DK9]: And input?

Syntax:

@AA(Data)[CHKSUM](CR)

@ Delimiter character

AA The address of the module to be set in hexadecimal format (00 to FF)

(Data) A single-digit hexadecimal value representing the data to be written to the digital output channels, where bit 0 of the value corresponds to channel RL1. When the bit is 0, it denotes that the digital output channel is set to OFF, and 1 denotes that the digital output channel is set to ON.

註解 [DK10]: And input?

Response:

Valid Command: **>[CHKSUM](CR)**

Invalid Command: **?AA[CHKSUM](CR)**

Ignored Command: **![CHKSUM](CR)**

> Delimiter character to indicate that the command was valid

? Delimiter character to indicate that the command was invalid

! Delimiter character to indicate the command was ignored. The command will be ignored if a Host Watchdog timeout has occurred.

AA The address of the responding module in hexadecimal format (00 to FF)

There will be no response if the command syntax is incorrect, there is a communication error, or there is no module with the specified address.

Examples:

Command: @021 Response: **>**

Sets Channel RL1 of module 02 to ON, and the module returns a response indicating that the command was valid.

Command: #020A02 Response: ?02

Attempts to set Channel RL2 of module 02 to ON, but the module returns a response indicating that the command was invalid because Channel RL2 does not exist.

Command: #020A01 Response: **!**

Attempts to set Channel RL1 of module 02 to ON, but the module returns a response indicating that the command was ignored because a Host Watchdog timeout has occurred.

Related Commands:

Section 2.3 #AA00(Data), Section 2.4 #AA0A(Data), Section 2.5 #AA10DD, Section 2.6 #AAA0DD, Section 2.19 @AA

Related Topics:

Section 1.7 Digital Input/Output Data Format Settings

Notes:

This command is only applicable to modules that contain digital output channels.

註解 [DK11]: Why do we need this note?

3 Modbus RTU Protocol

The Modbus protocol was originally developed for Modicon controllers by Modicon Inc. Detailed information related to the Modbus RTU protocol can be found at http://www.modicon.com/techpubs/toc7.html. Visit http://www.modbus.org for more valuable information.

The LC-101 module supports the Modbus RTU protocol, with communication Baud Rates ranging from 1200 bps to 115200 bps. The parity, data bits and stop bits are fixed as no parity, 8 data bits and 1 stop bit. The following Modbus functions are supported.

Function Code	Description	Section
0x01	Reads the Coils	2.1
0x02	Reads the Discrete Inputs	2.2
0x03	Reads Multiple Registers	2.3
0x04	Reads Multiple Input Registers	2.4
0x05	Writes a Single Coil	2.5
0x0F	Writes Multiple Coils	2.6
0x46	Reads/writes the Module Settings	2.7

If the function specified in the message is not supported, then the module responds as below. Note that the address mapping for the Modbus protocol is Base 0.

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	Function Code + 0x80
02	Exception Code	1	01

Note: If a CRC mismatch occurs, the module will not respond.

3.1 01 (0x01) Read Coils

This function code is used to read the current digital output values from the LC-101 DIO module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x01
02 - 03	Starting Channel Number	2	0x0000 - 0x001F for the Digital Output Value 0x0020 - 0x003F for the Digital Input Value 0x0040 - 0x005F for the Digital Output and Digital Input Latch High Value 0x0060 - 0x007F for the Digital Output and Digital Input Latch Low Value
04 - 05	Output Channel Number	2	0x0001 - 0x001F

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x01
02	Byte Count	1	1
03	Output Channel Value	1	Refer to the Remarks section below.

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x81
02	Exception Code	1	Refer to the Modbus standard for more details.

Remarks

LC-101:

Valid Starting Channel	0x0000 for the Digital Output Value
	0x0020 for the Digital Input Value
	0x0040 for the Digital Input Latch High Value 0x0044 for the Digital Output Latch High Value
	0x0060 for the Digital Input Latch Low Value 0x0064 for the Digital Output Latch Low Value

Examples:

註解 [DK12]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.2 02 (0x02) Read Discrete Inputs

This function code is used to read the current digital input value from the LC-101 DIO module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x02
02 - 03	Starting Channel Number	2	0x0020 - 0x003F
04 - 05	Input Channel Number	2	0x0001 - 0x0020

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x02
02	Byte Count	1	1
03	Input Channel Data	1	Refer to the Remarks section below.

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x82
02	Exception Code	1	Refer to the Modbus standard for more details.

Remarks

LC-101:

Valid Starting Channel	0x0020 for the Digital Input value

Examples:

註解 [DK13]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.3 03 (0x03) Read Multiple Registers

This function code is used to read the current digital input count value from the LC-101 DIO module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x03
02 - 03	Starting Channel Number	2	0x0000 - 0x001F
04 - 05	Input Channel Number	2	0x0001 - 0x0020

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x03
02	Byte Count	1	1
03 -	Input Channel Count Value	*N x 2	Each channel can record the count value up to a maximum of 65535 (0xFFFF) bytes.

*N = Number of Input Channels

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x83
02	Exception Code	1	Refer to the Modbus standard for more details.

Remarks

LC-101:

Valid Starting Channel	0x0000 for the Digital Input Count Value

Examples:

註解 [DK14]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.4 04 (0x04) Read Multiple Input Registers

This function code is used to read the current digital input count values from the LC-101 DIO module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x04
02 - 03	Starting Channel Number	2	0x0000 - 0x001F
04 - 05	Input channel number	2	0x0001 - 0x0020

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x04
02	Byte Count	1	1
03 -	Input Channel Count Value	*N x 2	Each channel can record the count value up to a maximum of 65535 (0xFFFF) bytes.

*N = Number of Input Channels

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x84
02	Exception Code	1	Refer to the Modbus standard for more details.

Remarks

LC-101:

Valid Starting Channel	0x0000 for the Digital Input Count Value

Examples:

註解 [DK15]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.5 05 (0x05) Write Single Coil

This function code is used to write the digital output value for the LC-101 DIO module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x05
02 - 03	Starting Channel Number	2	0x0000 - 0x001F 0x0100 to clear the Latch Value 0x0200-0x0220 to clear the Digital Input Count Value
04 - 05	Output Value	2	A value of 0xFF00 will set the output to ON. A value of 0x0000 will set it to OFF. All other values are invalid and will not affect the coil.

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x05
02 - 03	Output Channel Number	2	The value is the same as bytes 02 and 03 of the Request
04 - 05	Output Value	2	The value is the same as bytes 04 and 05 of the Request

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x85
02	Exception Code	1	Refer to the Modbus standard for more details.

Remarks

LC-101:

Valid Output Channel	0x0000 for the Digital Output Channel RL1.
	0x0100 to clear the Digital Output and Digital Input latch value. If setting this channel to ON, the latch value will be set to 0. 0x0200 to clear the Digital Input count value.

Examples:

註解 [DK16]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.6 15 (0x0F) Write Multiple Coils

This function code is used to write the digital output values for the LC-101 DIO module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x0F
02 - 03	Starting Channel Number	2	0x0000 - 0x001F for the Digital Output 0x0200 - 0x0220 to clear the Digital Input Count Value
04 - 05	Output Channel Number	2	0x0001 - 0x0020
06	Byte Count	1	1
07	Output Value	1	A bit corresponds to a channel. If the bit is 1, it denotes that the channel that was set is ON. If the bit is 0, it denotes that the channel that was set is OFF.

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x0F
02 - 03	Starting Channel Number	2	The value is the same as byte 02 and 03 of the Request
04 - 05	Input Channel Number	2	The value is the same as byte 04 and 05 of the Request

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x8F
02	Exception Code	1	Refer to the Modbus standard for more details.

Remarks

LC-101:

Valid Starting Channel	Write 0x0000 for the Digital Output
	Write 0x0200 to clear the Digital Input Count Value

Examples:

註解 [DK17]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.7 70 (0x46) Read/Write Module Settings

This function code is used to read the configuration settings from the module or to change the settings for the module. The following sub-function codes are supported.

Sub-function Code	Description	Section
00 (0x00)	Reads the Module Name	3.7.1
04 (0x04)	Sets the Module Address	3.7.2
05 (0x05)	Reads the Communication Settings	3.7.3
06 (0x06)	Sets the Communication Settings	3.7.4
32 (0x20)	Reads the Firmware Version	3.7.5
33 (0x21)	Sets the Digital Input Counter Edge	3.7.6
34 (0x22)	Reads the Digital Input Counter Edge Settings Value	3.7.7
39 (0x27)	Sets the Digital Output Power-on Value	3.7.8
40 (0x28)	Reads the Digital Output Power-on Value	3.7.9

If the module does not support the sub-function code specified in the message, then it will respond as follows:

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details.

3.7.1 Sub-function 00 (0x00) Read Module Name

This sub-function code is used to read the name of the LC-101 module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x00

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x00
03 - 06	Module Name	4	0x00 0x01 0x01 0x00

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details.

Examples:

註解 [DK18]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.7.2 Sub-function 04 (0x04) Write Module Address

This sub-function code is used to set the address of the LC-101 module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x04
03	New Address	1	1 - 247
04 - 06	Reserved	3	0x00 0x00 0x00

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x04
03	New Address	1	0: OK Others: Error
04 - 06	Reserved	3	0x00 0x00 0x00

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details.

Examples:

註解 [DK19]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.7.3 Sub-function 05 (0x05) Read Communication Settings

This sub-function code is used to read the communication protocol settings for the LC-101 module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x05
03	Reserved	1	0x00

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x05
03	Reserved	1	0x00
04	Baud Rate	1	Refer to the Baud Rate Settings table **錯誤! 尚未指定書籤名稱。**below for details.
05 - 07	Reserved	3	0x00 0x00 0x00
08	Mode	1	0: DCON Protocol 1: Modbus RTU Protocol
09 - 10	Reserved	2	0x00 0x00

Note: This information is the data saved in the EEPROM and will be used for the next power-on reset. It is not the currently used settings.

Baud Rate Settings:

Value	03	04	05	06	07	08	09	0A
Baud Rate	1200	2400	4800	9600	19200	38400	57600	115200

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details.

Examples:

註解 [DK20]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.7.4 Sub-function 06 (0x06) Write Communication Settings

This sub-function code is used to configure the communication protocol for the LC-101 module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x06
03	Reserved	1	0x00
04	Baud Rate	1	Refer to the Baud Rate Settings table above for details.
05 - 07	Reserved	3	0x00 0x00 0x00
08	Mode	1	0: DCON Protocol 1: Modbus RTU Protocol
09 - 10	Reserved	2	0x00 0x00

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x06
03	Reserved	1	0x00
04	Baud Rate	1	0: OK Others: Error
05 - 07	Reserved	3	0x00 0x00 0x00
08	Mode	1	0: OK Others: Error
09 - 10	Reserved	2	0x00 0x00

Note: The new Baud Rate and Protocol settings will only become effective after the next power-on reset.

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details.

Examples:

註解 [DK21]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.7.5 Sub-function 32 (0x20) Read Firmware Version

This sub-function code is used to read the firmware version information for the LC-101 module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x20

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x20
03	Major Version	1	0x00 - 0xFF
04	Minor Version	1	0x00 - 0xFF
05	Build Version	1	0x00 - 0xFF

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details.

Examples:

註解 [DK22]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.7.6 Sub-function 33 (0x21) Write Digital Input Counter Edge Settings

This sub-function code is used to set the digital input counter edge value for the LC-101 module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x21
03	Edge Setting Value	1	*0x00 - 0x0F

* 0 = Falling Edge, 1 = Rising Edge. For example, 0x03 denotes that the counters for channels 0 and 1 are set to rising edge and those for channels 2 and 3 are set to falling edge.

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x21
03	Edge Setting Value	1	0: OK Others: Error

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details.

Examples:

註解 [DK23]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.7.7 Sub-function 34 (0x22) Read Digital Input Counter Edge Settings

This sub-function code is used to read the digital input counter edge value for the LC-101 module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x22

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x22
03	Edge Setting Value	1	*0x00 - 0x0F

*0 = Falling Edge, 1 = Rising Edge. For example, 0x03 denotes that the counters for channels 0 and 1 are set to rising edge and those for channels 2 and 3 are set to falling edge.

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details.

Examples:

註解 [DK24]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.7.8 Sub-function 39 (0x27) Write Power-on Value

This sub-function code is used to set the power-on value for the LC-101 module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x27
03	Power-on Value	1	*0x00-0xFF

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x28
03	Power-on Value	1	0: OK Others: Error

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details.

Examples:

註解 [DK25]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

3.7.9 Sub-function 40 (0x28) Read Power-on Value

This sub-function code is used to read the power-on value for the LC-101 module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x28

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x28
03	Power-on Value	1	*0x00 - 0xFF

Error Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 - 247
01	Function Code	1	0xC6
02	Exception Code	1	Refer to the Modbus standard for more details.

Examples:

註解 [DK26]: In other manuals, i.e., the ZT-2060, there were examples of the usage of these commands. Should we include similar things in this manual?

註解 [DK27]: Should we include a Troubleshooting section? Or a Watchdog section? Appendixes?