PIR-130-AC/DC

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: 2014/07/16

Table of Contents

1. Hardware Information

1.1 Introduction

The PIR-130 module includes a 1-channel passive infrared (PIR) sensor that is able to detect infrared waves generated by human within a range of approximately 8 meters in diameter with a 360° coverage area. The PIR-130 is used for indoor motion detection, and can be configured to automatically switch on a light if motion is detected.

The PIR-130 module also includes a 1-channel temperature sensor that can be used for measuring room temperature or can be configured to activate a fire alarm.

1.2 Specifications

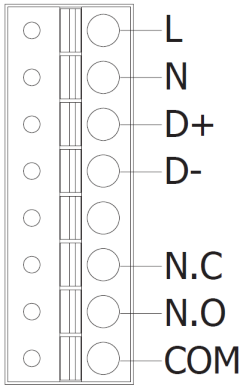
Model	PIR-130-AC	PIR-130-DC
PIR		
Time-delay	Hardware: 8-step Switch-selectable (seconds): 6, 16, 33, 66, 131, 262, 524, 1049 Software: 16-step (seconds): 2, 4, 6, 8, 16, 33, 49, 66, 131, 262, 393, 524, 1049, 2097, 3146, 4194	
LUX Control Level :	Hardware: 2 mode (Dawn and dust) / Software: 5-step	
Detection Range	Distance: 4 meters Max.	
Detection Field of View	360º; Diameter 8 meters Max.	
Temperature Sensor		
Measuring Range	-25 ~ +100 °C	
Fire Alarm	65°C (Programmable)	
Resolution	0.0625 °C	
Accuracy	±2 °C	
Relay Output		
Channels	1	
Type	Power Relay, Form C	
Max. Load Current	NO: 10A@250VAC NC: 6A@250VAC	
Load Wattage	Incandescent Bulb: 1500 W Max.; Fluorescent Lamp 300 W Max.	
RS-485 Interface		
COM Port	RS-485	
Transmission Distance (m)	Dependent on Baud Rate. For example, 1200 m Max. at 9600 bps.	
Baud Rate (bps)	Softeare:1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	
Protocol	DCON, Modbus RTU	
Node Address	Hardware:160 ~ 191 / Software:1 ~ 255	
LED Indicators		
LED Indicators	Yes,1 as Power/Communication Indicator. 1 as Alarm Indicator	
EMS Protection		
ESD (IEC 61000-4-2)	±4 kV Contact for Each Terminal, ±8 kV Air for Random Point	
EFT (IEC 61000-4-4)	±4 kV for Power Line	
Power Requirements		

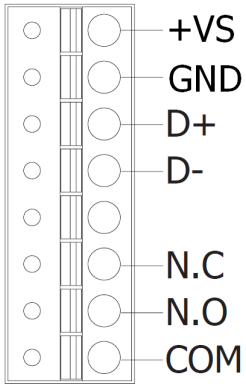
Power supply	100 ~ 240 VAC	10 ~ 30 VDC
Protection	Power reverse polarity protection, Over-voltage brown-out protection	
Power Consumption	2 W	1.3 W
Mechanical		
Installation	Ceiling mounting	
Protection Class	IP20	
Dimensions (D x H)	Ø 121 mm x 52 mm	
Environment		
Operating Temp.	-25 ~ 75 ℃	
Storage Temp.	-30 ~ 80 ℃	
Humidity	10 to 90% RH, non-condensing	

Model	**PIR-130-AC**	**PIR-130-DC**
Time Delay (seconds)	Hardware: 8-step Switch-selectable: 6, 16, 33, 66, 131, 262, 524, and 1049 Software: 16-step: 2, 4, 6, 8, 16, 33, 49, 66, 131, 262, 393, 524, 1049, 2097, 3146, and 4194	
LUX Control Level	Adjustable from daylight to darkness Hardware: 2 modes (Dawn and dusk) Software: 5-step	
Detection Range	Radius: 4 meters Max.	
Detection Field of View	360° (Max. Diameter of 8 meters)	
Temperature Sensor		
Measuring Range	-25 to +100°C	
Fire Alarm	65°C (Programmable)	
Resolution	0.0625°C	
Accuracy	±2°C	
Relay Output		
Channels	1	
Type	Power Relay, Form C	
Max. Load Current	NO: 10 A @ 250 VAC NC: 6A @ 250 VAC	
Load Wattage	Incandescent Bulb: 1500 W Max. Fluorescent Lamp: 300 W Max.	
RS-485 Interface		
COM Port	RS-485	
Transmission Distance (m)	Dependent on Baud Rate.	

	For example, 1200 m Max. at 9600 bps.	
Baud Rate (bps)	Software:1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	
Protocol	DCON, Modbus RTU	
Node Addresses	Hardware:160 to 191 Software:1 to 255	
LED Indicators		
System LED Indicators	Yes.1 as Power/Communication Indicator. 1 as Alarm Indicator	
EMS Protection		
ESD (IEC 61000-4-2)	±4 kV Contact for each Terminal ±8 kV Air for Random Point	
EFT (IEC 61000-4-4)	±4 kV for Power Line	
Power Requirements		
Power Supply	100 ~ 240 VAC	10 ~ 30 VDC
Protection	Power reverse polarity protection Over-voltage brown-out protection	
Power Consumption	2 W	1.3 W
Mechanical		
Installation	Ceiling mounting	
Protection Class	IP20	
Dimensions (D x H)	121 mm x 52 mm	
Environment		
Operating Temp.	-25 to 75℃	
Storage Temp.	-30 to 80℃	
Humidity	10 to 90% RH, Non-condensing	

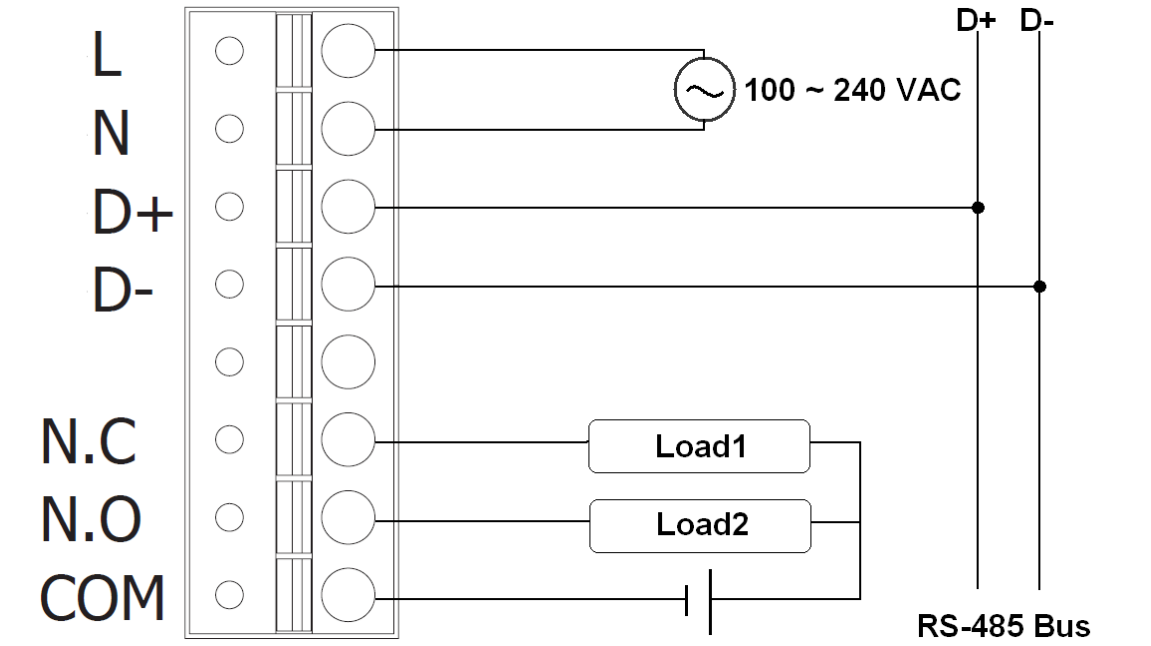
1.3 Pin Assignments

PIR-130-AC	Pin	Descriptions
	**L**	Power Line's Live Wire(100 ~ 240 VAC)
	N	Power Line's Natural Wire
	D+	RS-485 Serial Communication Interface
	D-	
	N.C	Relay's Normally Closed Contact
	N.O	Relay's Normally Open Contact
	COM	Relay's Common Contact

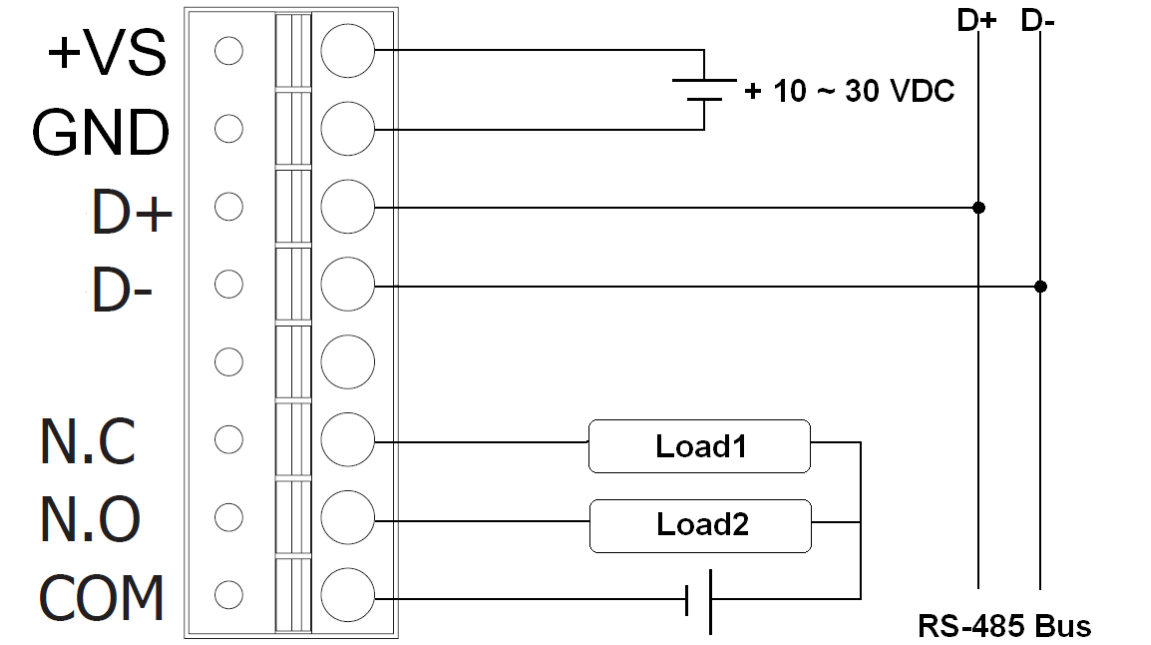
PIR-130-DC	Pin	Descriptions
	**+VS**	Power Input (+10 ~ +24 VDC)
	GND	Ground
	D+	RS-485 Serial Communication Interface
	D-	
	N.C	Relay's Normally Closed Contact
	N.O	Relay's Normally Open Contact
	COM	Relay's Common Contact

1.4 Wiring Connections

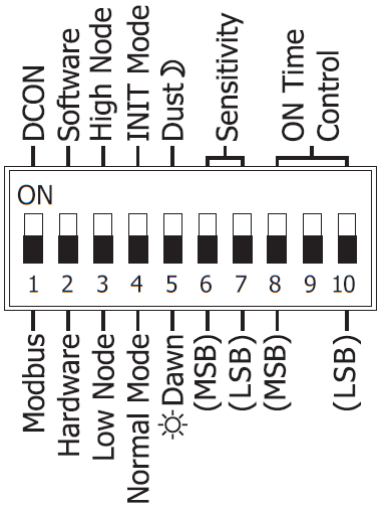
PIR-130-AC



PIR-130-DC



1.5 DIP Switch Configuration

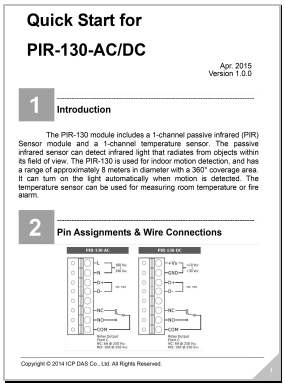


DIP [1]	**Protocol:** Used to specify the communication protocol to be used by the module ON: DCON OFF: Modbus RTU (default)
DIP [2]	**Configuration:** Used to specify the configuration settings for the module ON: Configure the module using DCON/Modbus commands OFF: Configure the module via DIP Switch (default)
DIP [3]	**Address:** Used to specify the module address when DIP [2] is set to OFF ON: Use Rotary Switch positions 0 to F for node addresses 176 to 191 OFF: Use Rotary Switch positions 0 to F for node addresses 160 to 175 (default)
DIP [4]	**Mode:** Used to specify the Operating Mode ON: Operating in INIT mode OFF: Operating in Normal mode (default)
DIP [5]	**PIR Operation:** Used to specify the Lux level at which the sensor will activate the light when movement is detected. ON: The PIR Sensor will only detect motion when the Lux level is between 0 and 200 OFF: The PIR Sensor will detect motion continuously (default)

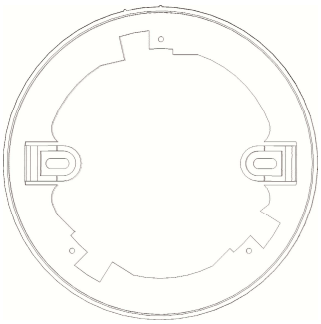
<table>
<tr><td>DIP [6:7]</td><td>Sensitivity:
Used to specify sensitivity of the PIR Sensor, and adjust the detection range of the sensor
<table>
<tr><th>DIP 6</th><th>DIP 7</th><th>Sensitivity</th></tr>
<tr><td>OFF</td><td>OFF</td><td>Maximum (default)</td></tr>
<tr><td>OFF</td><td>ON</td><td>High</td></tr>
<tr><td>ON</td><td>OFF</td><td>Low</td></tr>
<tr><td>ON</td><td>ON</td><td>Minimum</td></tr>
</table></td></tr>
<tr><td>DIP [8:10]</td><td>ON Time Control:
Used to specify the ON time for the relay after the PIR Sensor has been triggered.
<table>
<tr><th>DIP 8</th><th>DIP 9</th><th>DIP 10</th><th>ON Time</th></tr>
<tr><td>OFF</td><td>OFF</td><td>OFF</td><td>6 seconds (default)</td></tr>
<tr><td>OFF</td><td>OFF</td><td>ON</td><td>16 seconds</td></tr>
<tr><td>OFF</td><td>ON</td><td>OFF</td><td>33 seconds</td></tr>
<tr><td>OFF</td><td>ON</td><td>ON</td><td>66 seconds</td></tr>
<tr><td>ON</td><td>OFF</td><td>OFF</td><td>131 seconds</td></tr>
<tr><td>ON</td><td>OFF</td><td>ON</td><td>262 seconds</td></tr>
<tr><td>ON</td><td>ON</td><td>OFF</td><td>524 seconds</td></tr>
<tr><td>ON</td><td>ON</td><td>ON</td><td>1049 seconds</td></tr>
</table></td></tr>
</table>

1.6 Package Contents

PIR-130


Quick Start for
PIR-130-AC/DC
Apr. 2015
Version 1.0.0
1 Introduction
The PIR-130 module includes a 1-channel passive infrared (PIR) Sensor module and a 1-channel temperature sensor. The passive infrared sensor can detect infrared light that radiates from objects within its field of view. The PIR-130 is used for indoor motion detection, and has a range of approximately 8 meters in diameter with a 360° coverage area. It can turn on the light automatically when motion is detected. The temperature sensor can be used for measuring room temperature or fire alarm.
2 Pin Assignments & Wire Connections
Copyright © 2014 ICP DAS Co., Ltd. All Rights Reserved.

Quick Start Guide

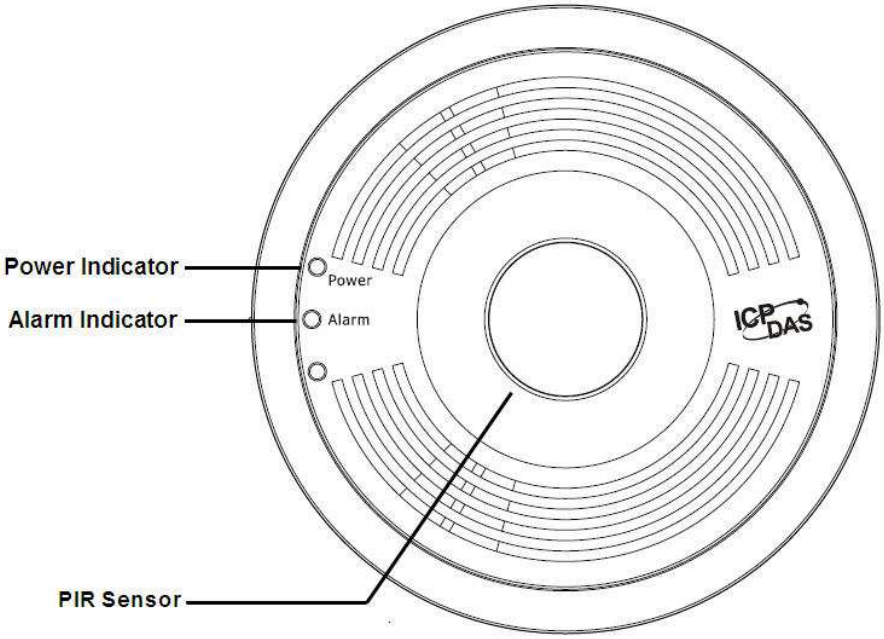


Mounting Plate

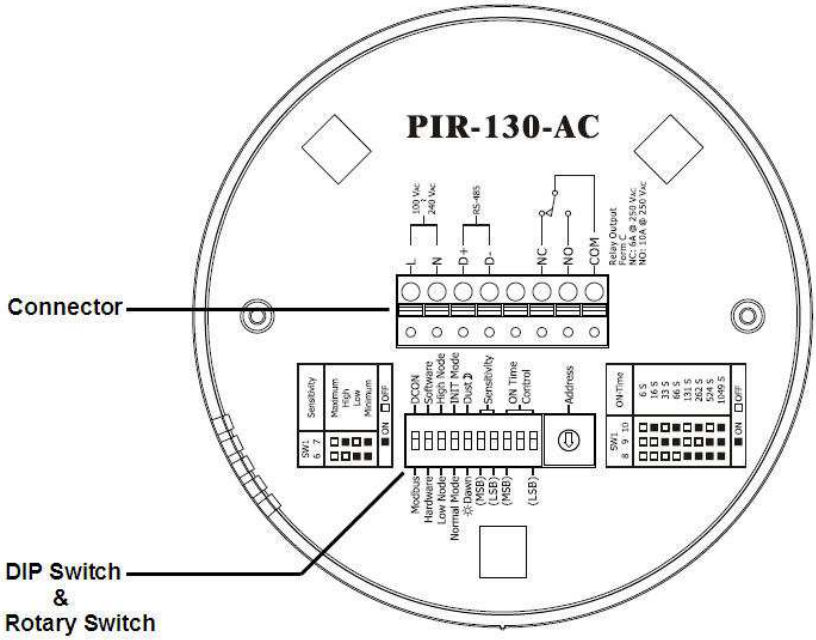
M4x12 Drywall Screws

1.7 Hardware Overview

Front



Rear



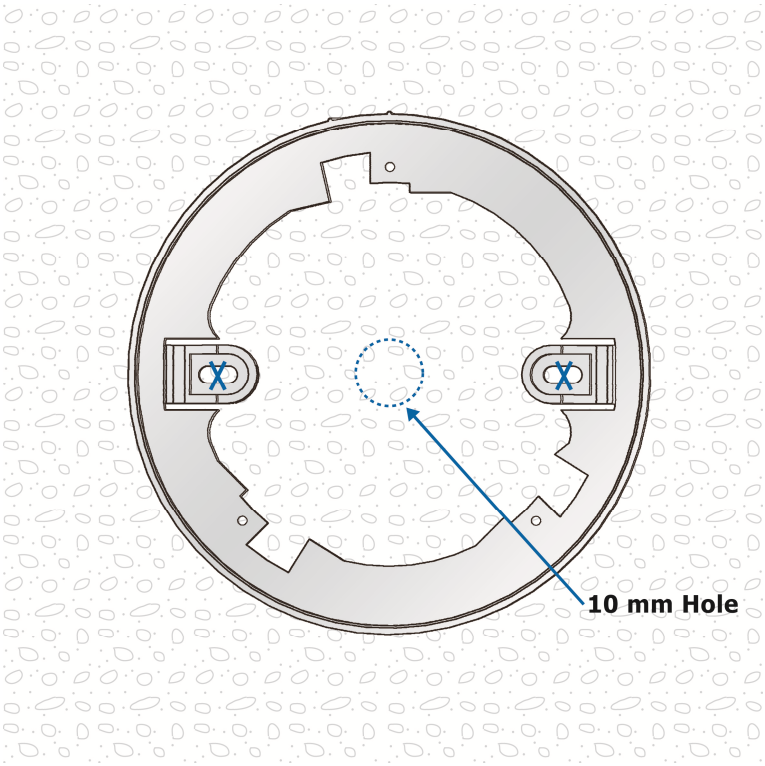
1.8 Hardware Installation

Installation Tips

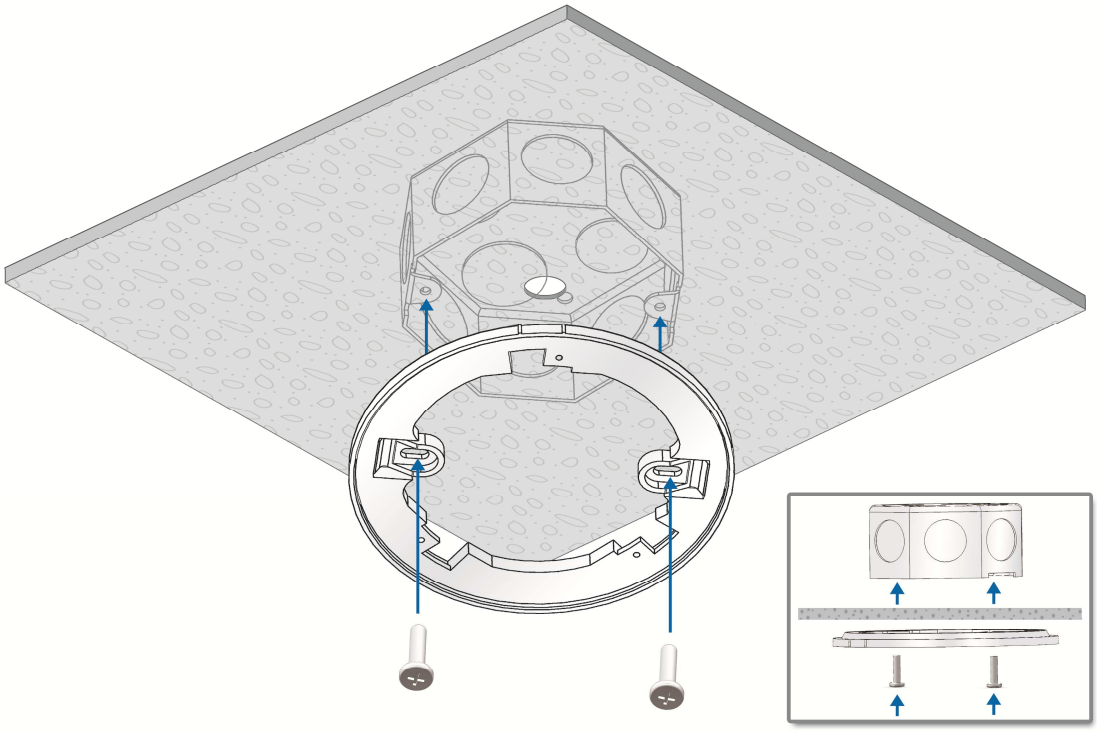
- Avoid installing the PIR-130 in areas where it will face direct or reflected sunlight.
- Avoid installing the PIR-130 in areas where the environmental temperature may change rapidly.
- Ensure that the PIR-130 is located at least one meter away from the nearest fluorescent light so as to avoid interference.
- Ensure that there are no obstructions in the field of view.

Installation Instructions

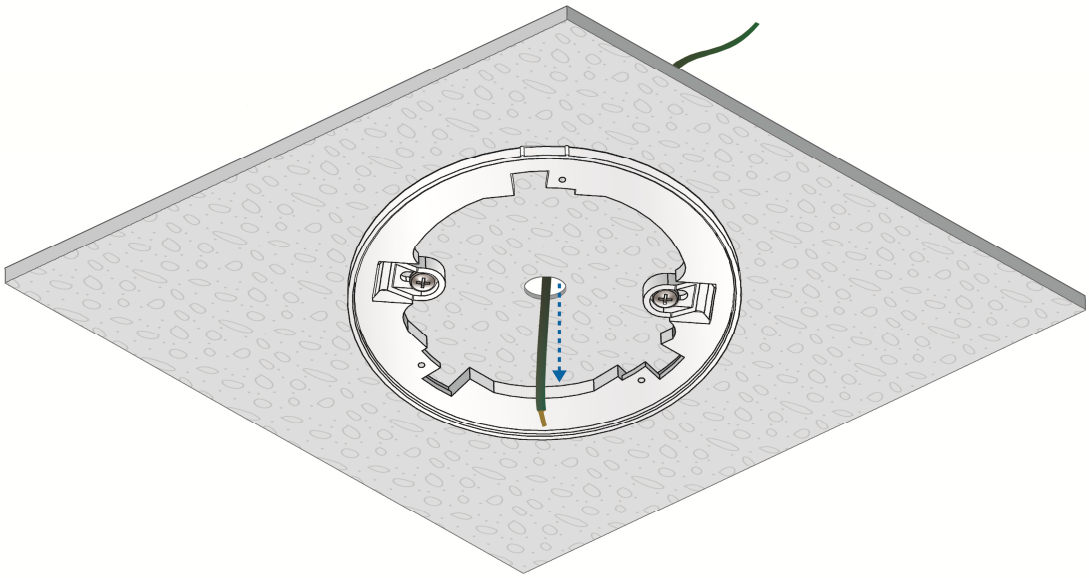
1. Position the Mounting Plate in the desired location. Mark the positions of the two screw holes and a 10 mm hole, as indicated below.



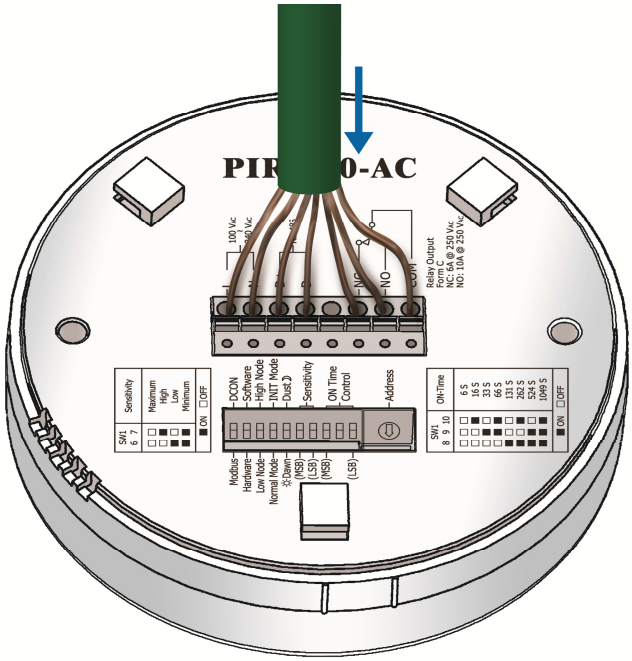
2. Secure the Mounting Plate to the ceiling using the M4x12 drywall screws and the optional octagonal box.



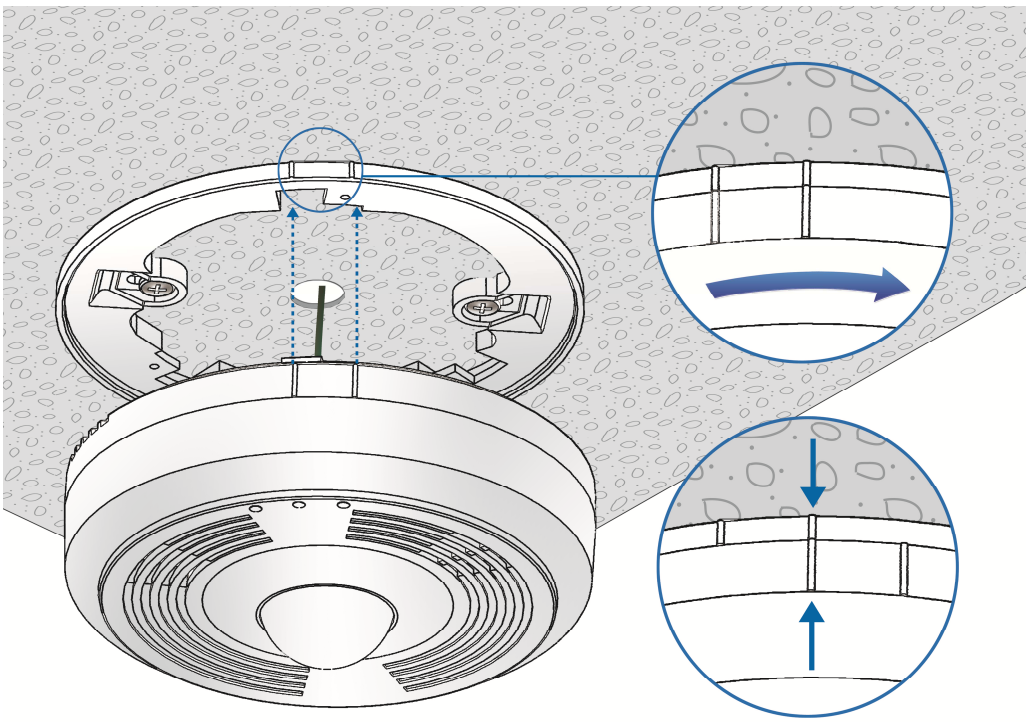
3. Feed the wires through the wiring hole.



4. Connect all the wires to the appropriate locations on the connector.



5. Align the marks on the PIR-130 with the marks on the Mounting Plate.
6. Rotate the PIR-130 clockwise until it locks into place.



1.9 Software Configuration Tables

Baud Rate Settings (CC)

7	6	5	4	3	2	1	0
Parity		Baud Rate Code					

Parity (Bits 6 and 7)

Code	00	01	10	11
Parity	n,8,1	n,8,2	e,8,1	o,8,1

Baud Rate Code (Bits 0 to 5)

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

Data Format Settings (FF)

7	6	5	4	3	2	1	0
Reserved	CS	Reserved					

Key	Description
CS	Checksum Settings 0: Disabled 1: Enabled

Note: All Reserved bits should be zero.

2. DCON Protocol

All communication with the PIR-130 module consists of commands generated by the Host and responses transmitted by the PIR-130 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.

General Command Sets			
Command	**Response**	**Description**	**Section**
%AANNTTCCFF	!AA	Sets the Configuration of the Module	2.1
$AA2	!AANNTTCCFF	Reads the Configuration of the Module	2.2
$AA5	!AAS	Reads the Reset Status of the Module	2.3
$AAF	!AA(Data)	Reads the Firmware Version of the Module	2.5
$AAM	!AA(Data)	Reads the Name of the Module	2.6
$AAP	!AASC	Reads the Communication Protocol currently used by the module	2.7
$AAPN	!AA	Sets the Communication Protocol to be used by the module	2.8
~AARD	!AA(Data)	Reads the current Response Delay Time	2.22
~AARDVV	!AA	Sets the Response Delay Time for the Module	2.23

PIR Relay Output Status Command Sets			
Command	**Response**	**Description**	**Section**
@AADI	!(Data)	Reads the current status of the PIR Relay Output	2.9
~AAD	!AAVV	Reads the current Active State of the PIR Relay Output	2.20
~AADVV	!AA	Sets the Active State of the PIR Relay Output	2.21

PIR Argument Command Sets			
Command	**Response**	**Description**	**Section**
$AALC3C0NNNN	!AA	Sets the Active Delay Time for the Relay Output	2.10
$AALC4C0	!AANNNN	Reads the current Active Delay Time for the Relay Output	2.11
$AALC5C0NN	!AA	Sets the value of the Luminance Level for the PIR Sensor	2.12
$AALC6C0	!AANN	Reads the current value of the Luminance Level	2.13

		for the PIR Sensor	
$AALC7C0NN	!AA	Sets the Relay Output ON Time for when the PIR Sensor is triggered	2.14
$AALC8C0	!AANN	Reads the current Relay Output ON Time for when the PIR Sensor is triggered	2.15
$AALC9C0NN	!AA	Sets the Operation Mode for the Buzzer	2.16
$AALCAC0	!AANN	Reads the current Operation Mode for the Buzzer	2.17
$AALCBC0NN	!AA	Sets the Sensitivity Value for the PIR Sensor	2.18
$AALCC0V	!AANN	Reads the current Sensitivity Value for the PIR Sensor	2.19

High Alarm Command Sets			
Command	**Response**	**Description**	**Section**
@AAEAT	!AA	Enables the High Alarm Function	2.24
@AAHI(Data)	!AA	Sets the High Alarm Condition Value	2.25
@AADA	!AA	Disables the High Alarm Function	2.26
@AACHC0	!AA	Clears the Status of the High Alarm	2.27
@AARH	!AA(Data)	Reads the current value of the High Alarm	2.28
@AARAO	!AAHH00	Read the currently activated alarm	2.29

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	The Type Code, which should be set to 40 for DIO modules
CC	The new Baud Rate, see Section 1.10 for details. The INIT DIP Switch (DIP 4) must be set to the ON position in order to change Baud Rates, see Section 1.5 for details.
FF	The command used to set the update direction of the counter and the Checksum, see Section 1.10 for details. The INIT DIP Switch (DIP 4) must be set to the ON position, see Section 1.5 for details.

Note that the INIT DIP Switch (DIP 4) must be set to the ON position before using this command. See Section 1.5 for more details.

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 moving the INIT DIP Switch (DIP 4) to the ON position, the module will return a response indicating that the command was invalid. See Section 1.5 for more details.
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

Attempts to change the Baud Rate of module 01 to 115200 bps, but the module returns a response indicating that the command was invalid because the INIT DIP Switch (DIP 4) hasn't been set to the ON position. See Section 1.5 for more details.

Command: %0101200A00 Response: !01

Changes the Baud Rate of module 01 to 115200 bps and the he INIT DIP Switch (DIP 4) has been set to the ON position. The module returns a response indicating that the command was valid.

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 direction of the counter update is Falling Edge.

Related Commands:

Section 2.2 $AA2

Related Topics:

Section 1.5 DIP Switch Configuration

Section 1.10 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 $AA2

Description:

This command is used to read the current 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 1.10 for details.
FF	The Checksum and counter update direction settings for the module. See Section 1.10 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: %0101200600 Response: !01

Changes the Baud Rate of module 01 to 9600 bps and the INIT DIP Switch (DIP 4) is in the ON position. The module returns a response indicating that the command was valid.

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.

Related Commands:

Section 2.1 %AANNTTCCFF

Related Topics:

Section 1.5 DIP Switch Configuration

Section 1.10 Software Configuration Tables

2.3 $AA5

Description:

This command is used to read the current 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.

Related Commands:

None

2.4 $AA6

Description:

This command is used to read the current status of the PIR Relay Output channel 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 current status of the PIR Relay Output channel

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 current status of the PIR Relay Output channel represented by a four-digit hexadecimal value followed by 00. The first two digits represent the status of the PIR Relay Output channel and the second two are reserved.

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 current status of the PIR Relay Output channel for module 02 and returns a response indicating that the command was valid and that the current status of the PIR Relay Output channel is active.

Related Commands:

Section 2.9 @AADI

2.5 $AAF

Description:

This command is used to read the current 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 current 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 current 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 current firmware version of module 01, and returns a response indicating that the command was valid, and that the firmware is version 01.00.

Related Commands:

None

2.6 $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 read 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: !02PIR130

Reads the name of module 02 and returns a response indicating that the command was valid, and that the name of the module is “PIR-130”.

Related Commands:

None

2.7 $AAP

Description:

This command is used to read the current communication protocol information configured 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 current 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 communication protocol(s) supported by the module:

0: Only the DCON protocol is supported

1: Both the DCON and Modbus RTU protocols are supported

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

0: The communication protocol currently saved in the EEPROM is DCON

1: The communication protocol currently 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: $01P1 Response: !01

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

Command: $01P Response: !0110

Reads the current communication protocol information configured for module 01, and 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 DCON protocol will be used at the next power-on reset.

Related Commands:

Section 2.8 $AAPN

Related Topics:

Section 1.5 DIP Switch Configuration

2.8 $AAPN

Description:

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

Syntax:

$AAPN[CHKSUM](CR)

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

Note that the INIT DIP Switch (DIP 4) must be set to the ON position before using this command. See Section 1.5 for more details. 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 by module 01 to Modbus RTU, but the module returns a response indicating that the command was invalid because the INIT DIP Switch (DIP 4) has not been set to the ON position. See Section 1.5 for more details.

Command: $01P1 Response: !01

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

Command: $01P Response: !0110

Reads the current communication protocol information configured 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 DCON protocol will be used at the next power-on reset.

Response: ?01

Command: $01P1

Attempts to set the current communication protocol into Modbus RTU for module 01, but the module returns a response indicating that the command was invalid because the INIT DIP Switch (DIP 4) hasn't been set to the ON position. See Section 1.5 for more details.

Related Commands:

Section 2.7 $AAP

Related Topics:

Section 1.5 DIP Switch Configuration

2.9 @AADI

Description:

This command is used to read the current status of the PIR Relay Output channel and PIR active status on a specified module.

Syntax:

@AA[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
DI	The command to read the current status of the PIR Relay Output channel and PIR active status

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)
S	High temperature alarm enable status, 0=alarm disable, 1=momentary alarm enabled, 2=latch alarm enabled.
OO	The status of the PIR Relay Output channel represented by a two-digit hexadecimal value. 00: PIR Relay Output is inactive; 01: PIR Relay Output is active.
II	The PIR active status represented by a two-digit hexadecimal value. 00: The status of the PIR is active; 01: The status of the PIR is inactive.

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: @02DI Response: >0100101

Reads the status of the PIR Relay Output channel for module 02 and returns a response indicating that the command was valid, and that both the PIR Relay Output channel and the status of the PIR are active.

Related Commands:

Section 2.4 $AA6

2.10 $AALC3C0NNNN

Description:

This command is used to set the Active Delay Time for a specific PIR Relay Output channel on a specified module.

Syntax:

$AALC3C0NNNN[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
LC3	The command to set the Active Delay Time for the PIR Relay Output channel
C	The command to set the PIR Relay Output channel
0	Specifies the PIR Relay Output channel to be set, zero based. Note that as there is only one PIR Sensor channel on the PIR-130 module, the only valid value is 0.
NNNN	A four-digit hexadecimal value representing the Active Delay Time in milliseconds. The maximum delay time is 0x0BB8 (3000 milliseconds).

Response:

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

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

>	Delimiter character to indicate the command was valid
?	Delimiter character to indicate 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: $01LC3C003E8 Response: >01

Sets the Active Delay Time for the PIR Relay Output channel of module 01 to 0x03E8 (1000 milliseconds) and the module returns a response indicating that the command was valid. The PIR Relay Output channel will be active for 1000 milliseconds after the module is powered on.

Command: $01LC4C0 Response: !010BB8

Reads the Active Delay Time for the PIR Relay Output channel of module 01 and returns a response indicating that the command was valid, with a value of 0BB8 meaning that the Active Delay Time is 3000 milliseconds. The PIR Relay Output channel will be active for 3000 milliseconds after the module is powered on.

Command: $01LC3C00BB9 Response: ?01

Attempts to set the Active Delay Time for the PIR Relay Output channel of module 01 to 0x0BB9 (3001 milliseconds), but the module returns a response indicating that the command was invalid because the value for the Active Delay Time was not within the valid range.

Related Commands:

Section 2.11 $AALC4C0

2.11 $AALC4C0

Description:

This command is used to read the Active Delay Time for a specific PIR Relay Output channel on a specified module.

Syntax:

$AALC4C0[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
LC4	The command to read the Active Delay Time for the PIR Relay Output channel
C	The command to read the PIR Relay Output channel
0	Specifies the PIR Relay Output channel to be read, zero based. Note that as there is only one PIR Sensor channel on the PIR-130 module, the only valid value is 0.

Response:

Valid Command: **!AANNNN[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)
NNNN	A four-digit hexadecimal value representing the Active Delay Time in milliseconds

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: $01LC3C00BB8 Response: !01

Sets the Active Delay Time for PIR Relay Output channel 0 of module 01 to 0x0BB8 (3000 milliseconds) and the module returns a response indicating that the command was valid. The PIR Relay Output channel will be active for 3000 milliseconds after the module is powered on.

Command: $01LC4C0 Response: !010BB8

Reads the Active Delay Time for Relay Output channel 0 of module 01 and returns a response indicating that the command was valid, with a value of 0BB8, meaning that the Active Delay Time is 3000 milliseconds, so the PIR Relay Output channel will be active for 3000 milliseconds after the module is powered on.

Related Commands:

Section 2.10 $AALC3C0NNNN

2.12 $AALC5C0NN

Description:

This command is used to set the Luminance Value for the PIR Sensor on a specific channel of a specified module.

Syntax:

$AALC5C0N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
LC5	The command to set the Luminance Value for the PIR Sensor
C	The command to set the PIR Relay Output channel
0	Specifies the PIR Relay Output channel to be set, zero based. Note that as there is only one PIR Sensor channel on the PIR-130 module, the only valid value is 0.
NN	The command to set the Luminance Value for the PIR Sensor, where: 00: Disabled 01: 100 Lux 02: 200 Lux 03: 500 Lux 04: 1000 Lux The valid range is 0 to 4. This value will be stored in the EEPROM.

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: $01LC5C00 Response: !01

Sets the Luminance Value for the PIR Sensor on channel 0 of module 01 to a value of 0 meaning that the Luminance value is disabled, and the module returns a response indicating that the command was valid.

Command: $01LC5C05 Response: ?01

Attempts to set the Luminance Value for the PIR Sensor on channel 0 of module 01 to a value of 5, but the module returns a response indicating that the command was invalid because the value is not within the valid range.

Command: $01LC6C0 Response: >011

Reads the Luminance Value for the PIR Sensor on channel 0 of module 01 and returns a response indicating that the command was valid, with a value of 1, meaning that the Luminance Value is 100 Lux

Related Commands:

Section 2.13 $AALC6C0

2.13 $AALC6C0

Description:

This command is used to read the Luminance Value for the PIR Sensor on a specific channel of a specified module.

Syntax:

$AALC6C0[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
LC6	The command to read the Luminance Value for the PIR Sensor
C	The command to read the PIR channel
0	Specifies the PIR channel to be read, zero based. Note that as there is only one PIR Sensor channel on the PIR-130 module, the only valid value is 0.

Response:

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

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

Ignored 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)
NN	The Luminance Value for the PIR Sensor, which is stored in the EEPROM. The valid range is 0 to 4, where: 0: Disabled 1: 100 Lux 2: 200 Lux 3: 500 Lux 4: 1000 Lux

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: $01LC5C01 Response: !01

Sets the Luminance Value for the PIR Sensor on channel 0 of module 01 to a value of 1, meaning that the Luminance Value is 100 Lux, and the module returns a response indicating that the command was valid.

Command: $01LC6C0 Response: >0101

Reads the Luminance Value for the PIR Sensor on channel 0 of module 01, and the module returns a response indicating that the command was valid with a value of 1, meaning that the Luminance Value is 100 Lux.

Command: $01LC6C1 Response: ?01

Attempts to read the Luminance Value for the PIR Sensor on channel 1 of module 01, but the module returns a response indicating that the command was invalid because channel 1 does not exist on the PIR-130.

Related Commands:

Section 2.12 $AALC5C0N

2.14 $AALC7C0NN

Description:

This command is used to set the PIR Relay Output ON Time for when the PIR Sensor is triggered on a specific module.

Syntax:

$AALC70NN[CHKSUM](CR)

$ Delimiter character

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

LC7 The command to set the PIR Relay Output ON Time for when the PIR Sensor is triggered

C The command to set the PIR Relay Output channel

0 Specifies the PIR Relay Output channel to be set, zero based.
Note that as there is only one PIR Sensor channel on the PIR-130 module, the only valid value is 0.

NN The command to set the PIR Relay Output ON Time for when the PIR Sensor is triggered in hexadecimal format. This value will be stored in the EEPROM, and the valid range is 00 to 0F.

NN	Seconds	NN	Seconds	NN	Seconds	NN	Seconds
00	2	01	4	02	6	03	8
04	16	05	33	06	49	07	66
08	131	09	262	0A	393	0B	524
0C	1049	0D	2097	0E	3146	0F	4194

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: $01LC7C000 Response: !01

Sets the PIR Relay Output ON Time value for channel 0 of module 01 to 00, meaning that the ON Time will be 2 seconds, and the module returns a response indicating that the command was valid.

Command: $01LC8C0 Response: >0100

Reads the PIR Relay Output ON Time value for channel 0 of module 01, and the module returns a response indicating that the command was valid, with a value of 00 meaning that the ON Time will be 2 seconds.

Command: $01LC7C100 Response: ?01

Attempts to set the PIR Relay Output ON Time value for channel 1 of module 01 to 00, but the module returns a response indicating that the command was invalid because channel 1 does not exist on the PIR-130 module.

Related Commands:

Section 2.15 $AALC8C0

2.15 $AALC8C0

Description:

This command is used to read the current PIR Relay Output ON Time for when the PIR Sensor is triggered on a specific module.

Syntax:

$AALC1[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
LC8	The command to read the PIR Relay Output ON Time for when the PIR Sensor is triggered
C	The command to read the PIR Relay Output channel
0	Specifies the PIR Relay Output channel to be read, zero based. Note that as there is only one PIR Sensor channel on the PIR-130 module, the only valid value is 0.

Response:

Valid Command: **!AANN[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)
NN	The PIR Relay Output ON Time value for when the PIR Sensor is triggered in hexadecimal format. This value will be stored in the EEPROM, and the valid range is 00 to 0F.

NN	Seconds	NN	Seconds	NN	Seconds	NN	Seconds
00	2	01	4	02	6	03	8
04	16	05	33	06	49	07	66
08	131	09	262	0A	393	0B	524
0C	1049	0D	2097	0E	3146	0F	4194

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: $01LC7C000 Response: !01

Sets the PIR Relay Output ON Time value for channel 0 of module 01 to 00, meaning that the ON Time will be 2 seconds, and the module returns a response indicating that the command was valid.

Command: $01LC8C0 Response: >0101

Reads the PIR Relay Output ON Time value for channel 0 of module 01 and the module returns a response indicating that the command was valid, with a value of 00, meaning that the ON Time will be 2 seconds.

Command: $01LC8C1 Response: ?01

Attempts to read the PIR Relay Output ON Time value for channel 1 of module 01, but the module returns a response indicating that the command was invalid because channel 1 does not exist on the PIR-130 module.

Related Commands:

Section 2.14 $AALC7C0NN

2.16 $AALC9C0NN

Description:

This command is used to set the Buzzer Operation Mode for a specific channel on a specified module.

Syntax:

$AALC9C0N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
LC9	The command to set the Buzzer Operation Mode
C	The command to set the PIR Relay Output channel
0	Specifies the PIR Relay Output channel to be set, zero based. Note that as there is only one PIR Sensor channel on the PIR-130 module, the only valid value is 0.
NN	The command to set the Buzzer Operation Mode, where: 0: The Buzzer will **NOT** be activated when the PIR Sensor is triggered 1: The Buzzer **WILL** be activated when the PIR Sensor is triggered This value will be stored in the EEPROM.

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: $01LC9C01 Response: !01

Sets the Buzzer Operation Mode for channel 0 of module 01 to 1 meaning that the Buzzer will be activated when the PIR Sensor is triggered, and the module returns a response indicating that the command was valid.

Command: $01LCAC0 Response: !0101

Reads the Buzzer Operation Mode for channel 0 of module 01 and the module returns a response indicating that the command was valid, with a value of 1, meaning that the Buzzer will be activated when the PIR Sensor is triggered.

Command: $01LC9C11 Response: ?01

Attempts to set the Buzzer Operation Mode for channel 1 of module 01 to 1, but the module returns a response indicating that the command was invalid because channel 1 does not exist on the PIR-130 module.

Related Commands:

Section 2.17 $AALCAC0

2.17 $AALCAC0

Description:

This command is used to read the current Buzzer Operation Mode for a specific channel on a specified module.

Syntax:

$AALCAC0[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
LCA	The command to read the Buzzer Operation Mode
C	The command to read the PIR Relay Output channel
0	Specifies the PIR Relay Output channel to be read, zero based. Note that as there is only one PIR Sensor channel on the PIR-130 module, the only valid value is 0.

Response:

Valid Command: **!AAN[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)
NN	The Buzzer Operation Mode, which is stored in the EEPROM

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: $01LC9C01 Response: !01

Sets the Buzzer Operation Mode for channel 0 of module 01 to 1, meaning that the Buzzer will be activated when the PIR Sensor is triggered, and the module returns a response indicating that the command was valid.

Command: $01LCAC0 Response: >0101

Reads the Buzzer Operation Mode for channel 0 of module 01 and the module returns a response indicating that the command was valid, with a value of 1, meaning that the Buzzer will be activated when the PIR Sensor is triggered.

Command: $01LCAC1 Response: ?01

Attempts to read the Buzzer Operation Mode for channel 1 of module 01, but the module returns a response indicating that the command was invalid because channel 1 does not exist on the PIR-130 module.

Related Commands:

Section 2.16 $AALC9C0N

2.18 $AALCBC0NN

Description:

This command is used to set the Sensitivity value for the PIR Sensor on a specific channel of a specified module.

Syntax:

$AALCBC0N[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
LCB	The command to set the Sensitivity value for the PIR Sensor
C	The command to set the PIR Relay Output channel
0	Specifies the PIR Relay Output channel to be set, zero based. Note that as there is only one PIR Sensor channel on the PIR-130 module, the only valid value is 0.
NN	The command to set the Sensitivity value for the PIR Sensor. The valid range is 0 to 9, where a lower value denotes a higher sensitivity, and this value will be stored in the EEPROM.

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: $01LCBC01 Response: !01

Sets the Sensitivity value for the PIR Sensor on channel 0 of module 01 to 1, and the module returns a response indicating that the command was valid.

Command: $01LCCC0 Response: !011

Reads the Sensitivity value for the PIR Sensor on channel 0 of module 01 and the module returns a response indicating that the command was valid, with a value of 1.

Command: $01LCBC11 Response: ?01

Attempts to set the Sensitivity value for the PIR Sensor on channel 1 of module 01 to 1, but the module returns a response indicating that the command was invalid because channel 1 does not exist on the PIR-130 module.

Related Commands:

Section 2.17 $AALCCC0

2.19 $AALCCC0

Description:

This command is used to read the current Sensitivity value for the PIR Sensor on a specific channel of a specified module.

Syntax:

$AALCCC0[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
LCC	The command to read the Sensitivity value for the PIR Sensor
C	The command to read the PIR Relay Output channel
0	Specifies the PIR Relay Output channel to be read, zero based. Note that as there is only one PIR Sensor channel on the PIR-130 module, the only valid value is 0.

Response:

Valid Command: **!AAN[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)
NN	The Sensitivity value for the PIR Sensor, which is stored in the EEPROM. The valid range is 0 to 9, where a lower value denotes a higher sensitivity.

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: $01LCBC01 Response: !01

Sets the Sensitivity value for the PIR Sensor on channel 0 of module 01 to 1, and the module returns a response indicating that the command was valid.

Command: $01LCCC0 Response: >0101

Reads the Sensitivity value for the PIR Sensor on channel 0 of module 01, and the

module returns a response indicating that the command was valid, with a value of 1.

Command: $01LCCC1 Response: ?01

Attempts to read the Sensitivity value for the PIR Sensor on channel 1 of module 01, but the module returns a response indicating that the command was invalid because channel 1 does not exist on the PIR-130 module.

Related Commands:

Section 2.18 $AALCAC0N

2.20 ~AAD

Description:

This command is used to read whether the PIR Relay Output signal for a specified module is active or inactive.

Syntax:

~AAD [CHKSUM](CR)

~ Delimiter character

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

D The command to read whether the PIR Relay Output signal is active or inactive

Response:

Valid Command: **!AAVV[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)

VV A two-digit hexadecimal value representing the status of the PIR Relay Output signal. See below for details.

7	6	5	4	3	2	1	0
Reserved						OAS	Reserved

Key	Description
OAS	Specifies the status of the PIR Relay Output signal 0: An output value of 0 indicates that the relay is inactive An output value of 1 indicates that the relay is active 1: An output value of 0 indicates that the relay is active An output value of 1 indicates that the relay is inactive

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: ~02D02 Response: !02

Sets the PIR Relay Output signal for module 02 to 02, which denotes that the PIR Relay Output channel is in inactive mode, and returns a response indicating that the command was valid.

Command: ~02D Response: !0202

Reads the status of the Relay Output signal for module 02 and returns a response indicating that the command was valid, with a value of 02, which denotes that the PIR Relay Output channel is in inactive mode.

Related Commands:

Section 2.21 ~AADVV

2.21 ~AADVV

Description:

This command is used to set the PIR Relay Output signal for a specified module to active or inactive.

Syntax:

~AADVV[CHKSUM](CR)

~ Delimiter character

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

D The command to set the PIR Relay Output to active or inactive

VV A two-digit hexadecimal value representing the status of the PIR Relay Output signal. See below for details.

7	6	5	4	3	2	1	0
Reserved						OAS	Reserved

Key	Description
OAS	Specifies the status of the PIR Relay Output signal 0: An output value of 0 indicates that the relay is inactive An output value of 1 indicates that the relay is active 1: An output value of 0 indicates that the relay is active An output value of 1 indicates that the relay is inactive

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: ~02D02 Response: !02

Sets the PIR Relay Output signal for module 02 to 02, which denotes that the PIR Relay Output channel is in inactive mode, and returns a response indicating that the command was valid.

Command: ~02D Response: !0202

Reads the status of the PIR Relay Output signal for module 02 and returns a response indicating that the command was valid, with a value of 02, which denotes that the PIR Relay Output channel is in inactive mode.

Command: ~02D07 Response: ?02

Attempts to set the Relay Output signal for module 02 to 07, but returns a response indicating that the command was invalid because the output value was not within the valid range.

Related Commands:

Section 2.20 ~AAD

2.22 ~AARD

Description:

This command is used to read the Response Delay Time for a specified module.

Syntax:

~AARD[CHKSUM](CR)

~	Delimiter character
AA	The address of the module to be read in hexadecimal format (00 to FF)
RD	The command to read the Response Delay Time

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 two-digit hexadecimal value representing the Response Delay Time. The valid range is 00 to 1E in 1 ms intervals.

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: ~03RD1E Response: !03

Sets the Response Delay Time for module 03 to 1E (30 ms), and returns a response indicating that the command was valid.

Command: ~03RD Response: !031E

Reads the Response Delay Time for module 03 and returns a response indicating that the command was valid, with a value of 1E (30 ms).

Related Commands:

Section 2.23 ~AARDVV

2.23 ~AARDVV

Description:

This command is used to set the Response Delay Time for a specified module.

Syntax:

~AARDVV[CHKSUM](CR)

$	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
RD	The command to set the Response Delay Time.
VV	A two-digit hexadecimal value representing the Response Delay Time in milliseconds. The valid range is 00 to 1E in 1 ms intervals.

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: ~03RD1E Response: !03

Sets the Response Delay Time for module 03 to 1E (30 ms), and returns a response indicating that the command was valid.

Command: ~03RD Response: !031E

Reads the Response Delay Time for module 03 and returns a response indicating that the command was valid, with a value of 1E (30 ms).

Command: ~03RD1F Response: ?03

Attempts to set the Response Delay Time for module 03 to 1F (31 ms), but the module returns a response indicating that the command was invalid because the value specified for the Response Delay Time was not within the valid range.

Related Commands:

Section 2.22 ~AARD

2.24 @AAEAT

Description:

This command is used to enable the High Alarm function for a specified module and set the alarm type.

Syntax:

@AAEAT[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
EA	The command to enable the High Alarm function
T	The Alarm Type: M: Momentary Alarm L: Latch Alarm

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:@01EAM Response:!01

Enables the High Alarm function for module 01 and sets the alarm type to momentary, and returns a response indicating that the command was valid.

Command:@03DA Response:!03

Disables the High Alarm function for module 03, and returns a response indicating that the command was valid.

Related Commands:

Section 2.25 @AAHI(Data), Section 2.26 @AADA, Section 2.27 @AACHC0, Section 2.28 @AARH, Section 2.29 @AARAO

2.25 @AAHI(Data)

Description:

This command is used to set the High Alarm limits for a specified module.

Syntax:

@AAHI(Data)[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
HI	The command to set the High Alarm limits
(Data)	A signed value representing the High Alarm limits in degrees Celsius in the format xxx.xx. The valid range is +000.00 to +999.99 degrees Celsius.

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: @01HI+086.00 Response:!01

Sets the High Alarm limits for module 01 to +86.00 degrees Celsius, and returns a response indicating that the command was valid.

Command: @01RH Response: !03+086.00

Reads the high alarm limits for module 01, and returns a response indicating that the command was valid, with a value of +086.00, which denotes that the High Alarm limits is +86.0 degrees Celsius.

Command: @01HI+1000.00 Response:?01

Attempts to set the High Alarm limits for module 01 to +1000.00 degrees Celsius, but returns a response indicating that the command was invalid because the specified value was not within the valid range.

Related Commands:

Section 2.24 @AAEAT, Section 2.26 @AADA, Section 2.27 @AACHC0, Section 2.28 @AARH, Section 2.29 @AARAO

2.26 @AADA

Description:

This command is used to disable the High Alarm function for a specified module.

Syntax:

@AADA[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
DA	The command to disable the High Alarm function

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:@01EAM Response:!01

Enables the High Alarm function for module 01 and sets the alarm type to momentary, and returns a response indicating that the command was valid.

Command:@03DA Response:!03

Disables the High Alarm function for module 03, and returns a response indicating that the command was valid.

Related Commands:

Section 2.24 @AAEAT, Section 2.25 @AAHI(Data), Section 2.27 @AACHC0, Section 2.28 @AARH, Section 2.29 @AARAO

2.27 @AACHC0

Description:

This command is used to clear the status of a Latched High Alarm for a specified module.

Syntax:

@AACHC0[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be cleared in hexadecimal format (00 to FF)
CHC0	The command to clear the status of the Latched High Alarm.

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: @03CHC0 Response: !03

Clears the status of the Latched High Alarm for module 03, and returns a response indicating that the command was valid.

Related Commands:

Section 2.24 @AAEAT, Section 2.25 @AAHI(Data), Section 2.26 @AADA, Section 2.28 @AARH, Section 2.29 @AARAO

2.28 @AARH

Description:

This command is used to read the current High Alarm limits for a specified module.

Syntax:

@AARH[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
RH	The command to read the current High Alarm limits .

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 signed value representing the High Alarm limits in degrees Celsius in the format xxx.xx. The valid range is +000.00 to +999.99.

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: @03HI+090.50 Response:!03

Sets the High Alarm limits for module 03 to +90.5 degrees Celsius, and returns a response indicating that the command was valid.

Command: @03RH Response: !03+090.50

Reads the High Alarm limits for module 03, and returns a response indicating that the command was valid, with a value of +090.50, which denotes that the High Alarm limits is +90.5 degrees Celsius.

Related Commands:

Section 2.24 @AAEAT, Section 2.25 @AAHI(Data), Section 2.26 @AADA, Section 2.27 @AACHC0, Section 2.29 @AARAO, Section 2.29 @AARAO

2.29 @AARAO

Description:

This command is used to read the currently activated alarm for a specified module.

Syntax:

@AARH[CHKSUM](CR)

@	Delimiter character
AA	The address of the module to be set in hexadecimal format (00 to FF)
RAO	The command to read the currently activated alarm.

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)
HH	A two-digit hexadecimal value to represent the currently activated high alarms 00: High alarm is not active.
00	Reserved.

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: @03RAO Response:!030000

Reads the currently activated high alarm, and returns a response indicating that the command was valid.

Command: @03RH Response: !03+090.50

Reads the High Alarm limits for module 03, and returns a response indicating that the command was valid, with a value of +090.50, which denotes that the High Alarm limits is +90.5 degrees Celsius.

Related Commands:

Section 2.24 @AAEAT, Section 2.25 @AAHI(Data), Section 2.26 @AADA, Section 2.27 @AACHC0

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://www2.schneider-electric.com/sites/corporate/en/products-services/automation-control/automation-control.page. You can also visit http://www.modbus.org for more valuable information.

The PIR-130 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	3.1
0x02	Reads the Discrete Inputs	3.2
0x03	Reads Multiple Registers	3.3
0x04	Reads Multiple Input Registers	3.4
0x05	Writes a Single Coil	3.5
0x06	Writes a Single Register	3.6
0x0F	Writes Multiple Coils	3.7
0x10	Writes Multiple Registers	3.8
0x46	Reads/writes the Module Settings	3.9

Error Response

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.

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 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 Modbus Address Mapping

General Commands

<table>
<tr><th>Address</th><th>Description</th><th>Attribute</th></tr>
<tr><td>00257</td><td>Reads/sets the Communication Protocol
0: DCON
1: Modbus RTU</td><td>R/W</td></tr>
<tr><td>10273</td><td>Reads the Reset Status:
0: This is NOT the first time the module has been read since being powered on
1: This is the first time the module has been read since being powered on</td><td>R</td></tr>
<tr><td>40481-40482</td><td>Reads the Firmware Version</td><td>R</td></tr>
<tr><td>40483-40484</td><td>Reads the Name of the Module</td><td>R</td></tr>
<tr><td>30485</td><td>Reads/sets the Module address. The valid range is 1 to 247.</td><td>R/W</td></tr>
<tr><td>30486</td><td>Reads/sets the Baud Rate and the Data Format:
Bits 5:0 (Baud Rate)
<table><tr><td>Code</td><td>0x03</td><td>0x04</td><td>0x05</td><td>0x06</td></tr><tr><td>Baud</td><td>1200</td><td>2400</td><td>4800</td><td>9600</td></tr><tr><td>Code</td><td>0x07</td><td>0x08</td><td>0x09</td><td>0x0A</td></tr><tr><td>Baud</td><td>19200</td><td>38400</td><td>57600</td><td>115200</td></tr></table>
Baud Rate, valid range: 0x03 to 0x0A
Bits 7:6 (Data Format)
00: no parity, 1 stop bit
01: no parity, 2 stop bits
10: even parity, 1 stop bit
11: odd parity, 1 stop bit</td><td>R/W</td></tr>
<tr><td>30488</td><td>Reads/sets the Response Delay Time in milliseconds. The valid range is 0 to 30 ms (00 to 1E in 1 ms intervals).</td><td>R/W</td></tr>
</table>

PIR-related Commands

Address	Description	Attribute
10001	Reads the current status of the PIR Relay Output. 1: Active	R

<table>
<tr><td></td><td>0: Inactive</td><td></td></tr>
<tr><td>00262</td><td>Enables or disables the high temperature alarm..
0: Disabled
1: Enabled</td><td>R/W</td></tr>
<tr><td>00263</td><td>Reads/sets the high temperature alarm type.
0: Momentary Alarm
1: Latch Alarm</td><td>R/W</td></tr>
<tr><td>00274</td><td>Enables or disables Buzzer activation for when the PIR Sensor is triggered.
0: Disabled
1: Enabled</td><td>R/W</td></tr>
<tr><td>00305</td><td>Reads/sets the high alarm status, write 1 to clear latched high alarm</td><td>R/W</td></tr>
<tr><td>30226</td><td>Reads/sets the High alarm limits</td><td>R/W</td></tr>
<tr><td>30513</td><td>Reads/sets the PIR Relay Output On Time for when the PIR Sensor is triggered in hexadecimal format. The valid range is 00 to 0F.
<table>
<tr><th>Hex</th><th>Secs</th><th>Hex</th><th>Secs</th><th>Hex</th><th>Secs</th><th>Hex</th><th>Secs</th></tr>
<tr><td>00</td><td>2</td><td>01</td><td>4</td><td>02</td><td>6</td><td>03</td><td>8</td></tr>
<tr><td>04</td><td>16</td><td>05</td><td>33</td><td>06</td><td>49</td><td>07</td><td>66</td></tr>
<tr><td>08</td><td>131</td><td>09</td><td>262</td><td>0A</td><td>393</td><td>0B</td><td>524</td></tr>
<tr><td>0C</td><td>1049</td><td>0D</td><td>2097</td><td>0E</td><td>3146</td><td>0F</td><td>4194</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>30514</td><td>Reads/sets the Luminance Value for the PIR Sensor. The valid range is 0 to 4, where.
0: Disabled
1: 100 Lux
2: 200 Lux
3: 500 Lux
4: 1000 Lux</td><td>R/W</td></tr>
<tr><td>30516</td><td>Reads/sets the Active Delay Time for the Relay Output in milliseconds. The valid range is 0 to 0xBB8 (0 to 3000 milliseconds).</td><td>R/W</td></tr>
<tr><td>30517</td><td>Reads/sets Sensitivity value for the PIR Sensor. The valid range is 0 to 9, where a lower value denotes a higher sensitivity.</td><td>R/W</td></tr>
</table>

Modbus RTU Function Description:

(0xxxx): 0x05, 0x0F Function Code

(1xxxx): 0x01 Function Code

(3xxxx): 0x06, 0x10 Function Code

(4xxxx): 0x03 Function Code

3.2 01 (0x01) Read Coils

This function code is used to read the values at addresses 0xxxx and 1xxxx.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x01
02 - 03	Starting Address	2	Refer to the Modbus Address Mapping Table for details.
04 - 05	Number of Addresses Requested	2	0x0001 to 0x0001 + *N

*N = Number of addresses requested

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x01
02	Byte Count	1	*N
03	Value from the Requested Address	*N	

*N = (Number of addresses requested / 8)

Error Response

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

3.3 02 (0x02) Read Discrete Input

This function code is used to read the value at address 1xxxx.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x02
02 - 03	Starting Address	2	0x0020 to 0x003F
04 - 05	Number of Addresses Requested	2	0x0001 to 0x0001 + *N

*N = Number of addresses requested

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x02
02	Byte Count	1	*N
03	Value from the Requested Address	*N	

*N = (Number of addresses requested / 8)

Error Response

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

3.4 03 (0x03) Read Multiple Registers

This function code is used to read the values at addresses 3xxxx and 4xxxx.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x03
02 - 03	Starting Address	2	Refer to the Modbus Address Mapping Table for details.
04 - 05	Number of Addresses Requested	2	0x0001 to 0x0001 + *N

*N = Number of addresses requested

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x03
02	Byte Count	1	*N x 2
03 -	Value from the Requested Address	*N x 2	.

*N = Number of addresses requested

Error Response

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

3.5 04 (0x04) Read Multiple Input Registers

This function code is used to read the values at address 4xxxx.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x04
02 - 03	Starting Address	2	Refer to the Modbus Address Mapping Table for details.
04 - 05	Number of Addresses Requested	2	0x0001 to 0x0001 + *N

*N = Number of addresses requested

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x04
02	Byte Count	1	*N x 2
03 -	Value from the Requested Address	*N x 2	

*N = Number of addressee requested

Error Response

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

3.6 05 (0x05) Write Single Coil

This function code is used to write a value to address 0xxxx.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x05
02 - 03	Starting Address	2	Refer to the Modbus Address Mapping Table for details.
04 - 05	Value to be written	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 to 247
01	Function Code	1	0x05
02 - 03	Requested Address	2	The value is the same as bytes 02 and 03 of the Request
04 - 05	Value from the Requested Address	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 to 247
01	Function Code	1	0x85
02	Exception Code	1	Refer to the Modbus standard for more details.

3.7 06 (0x06) Write Single Register

This function code is used to write a value to address 3xxxx.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x06
02 - 03	Starting Address	2	Refer to the Modbus Address Mapping Table for details.
04 - 05	The value to be written	2	

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x06
02 - 03	Requested Address	2	The value is the same as bytes 02 and 03 of the Request
04 - 05	Value from the Requested Address	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 to 247
01	Function Code	1	0x86
02	Exception Code	1	Refer to the Modbus standard for more details.

3.8 15 (0x0F) Write Multiple Coils

This function code is used to write multiple values.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x0F
02 - 03	Starting Address	2	Refer to the Modbus Address Mapping Table for details.
04 - 05	Number of Addresses Requested	2	0x0001 to 0x0001 + *N
06	Byte Count	1	*N/8
07	The values to be written	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.

*N = Number of addresses requested

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x0F
02 - 03	Starting Address	2	The value is the same as bytes 02 and 03 of the Request
04 - 05	Value from the Requested Address	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 to 247
01	Function Code	1	0x8F
02	Exception Code	1	Refer to the Modbus standard for more details.

3.9 16 (0x10) Write Multiple Registers

This function code is used to write multiple values.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x10
02 - 03	Starting Address	2	Refer to the Modbus Address Mapping Table for details.
04 - 05	Number of Addresses Requested	2	0x0001 to 0x0001 + *N
06	Byte Count	1	*N x 2
07	The values to be written	*N x 2	

*N = Number of addresses requested

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x10
02 - 03	Starting Address	2	The value is the same as bytes 02 and 03 of the Request
04 - 05	Number of Addresses Requested	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 to 247
01	Function Code	1	0x90
02	Exception Code	1	Refer to the Modbus standard for more details.

3.10 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 Name of the Module	3.9.1
04 (0x04)	Sets the Address of the Module	3.9.2
05 (0x05)	Reads the Communication Settings	3.9.3
06 (0x06)	Sets the Communication Settings	3.9.4
32 (0x20)	Reads the Firmware Version	3.9.5

Error Response

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

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

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

This sub-function code is used to read the name of the PIR-130 module.

Request

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

Response

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 247
01	Function Code	1	0x46
02	Sub-function Code	1	0x00
03 - 06	Module Name	4	0x4C 0x43 0x01 0x01 (PIR-130)

Error Response

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

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

This sub-function code is used to set the address for the PIR-130 module.

Request

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

Response

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

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

This sub-function code is used to read the communication protocol settings for the PIR-130 module.

Request

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

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

This sub-function code is used to configure the communication protocol for the PIR-130 module.

Request

Byte	Description	Length (in Bytes)	Value
00	Address	1	1 to 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 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

Baud Rate Settings:

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

Response

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

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

This sub-function code is used to read the firmware version information for the PIR-130 module.

Request

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

Response

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