# CP500-110 Neutron™ Mini-Controller User's Guide

## Welcome!

We greatly appreciate your purchase of the CP500-110 Neutron Mini-Controller. We are sure you will find it reliable and simple to use. Superior performance for the right price, backed by solid technical and customer support is what ALTINEX has to offer.

We are committed to providing our customers with Signal Management Solutions® to the most demanding audiovisual installations at competitive pricing and we welcome you to join the ranks of our many satisfied customers throughout the world.

## 1. Precautions and Safety Warnings

Please read this manual carefully before using your Mini-Controller. Keep this manual handy for future reference. These safety instructions are to ensure the long life of your Mini-Controller and to prevent fire and shock hazards. Please read them carefully and heed all warnings.

### 1.1 General

- Qualified ALTINEX service personnel or their authorized representatives must perform all service.

### 1.2 Installation Precautions

- To prevent fire or shock, do not expose this unit to water or moisture. Do not place the Mini-Controller in direct sunlight, near heaters or heat-radiating appliances, or near any liquid. Exposure to direct sunlight, smoke, or steam can harm internal components.
- Handle the unit carefully. Dropping or jarring can cause damage.
- Do not pull any cables that are attached to the Mini-Controller.

### 1.3 Cleaning

- Clean only the connector area with a dry cloth. Never use strong detergents or solvents such as alcohol or thinner. Do not use a wet cloth or water to clean the card. Do not clean or touch any component or PCB.

### 1.4 FCC Notice

- This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.
- This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions found herein, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.
- Any changes or modifications to the unit not expressly approved by ALTINEX, Inc. could void the user's authority to operate the equipment.

## 2. Installation Procedures *(Read and become familiar with the entire online manual. Installation requires control commands in the online manual.)*

- Refer to the Application Diagrams section of the online manual for details on making connections to the Neutron Mini-Controller.

- Only use the terminal block captive screws for making connections. Do NOT tin the bare wires.

- Use multi-conductor cable, 24-28 AWG, stranded wire where possible for connections and make note of which equipment is connected to which port. For example, a 4-conductor cable works well for the RS-232 connection providing conductors for RCV, XMT, GND, and SHIELD.

Step 1. Accessories: Gather all accessories included with the Neutron Mini-Controller.

Step 2. Ethernet Port: Connect the controller's Ethernet port to a MultiTouch using the CAT-5 crossover cable provided or connect directly to a network or router using standard CAT-5 cable. The default static IP is 192.168.1.80 port 23. See the TCP Control sections (p. 10 and p. 13) for changing the IP. UDP control is available for units with firmware version 690-0332-004 and greater. See section 7.13 About TCP and UDP Communication.

Step 3. Power Connection: Connect the power adapter leads to the 2-pin terminal block. The Positive lead (white text or white stripe) connects to the Positive (+) input. Connect the AC cord provided between an AC outlet and the power adapter input; the Power LED turns ON and GREEN.

Step 4. IR Connections: The two emitters provided connect between IR1, IR2, and GND. The black wires with white stripes connect to the IR1+ and IR2+ pins. The emitter is applied over the eye of the receiving device using the emitter's adhesive. IR data is output from internal IR memory or the pass-through IN connection. The pass-through IR method uses an IR receiver like the AC301-103 connected to the IN and GND pins of the terminal block. If needed, the +12V pin provides power for the IR receiver.

Step 5. RS-232 Connections: Connect serial devices to the controller RS-232 connector; the device transmit pins connect to the Neutron RX pins and device receive pins to the controller TX. The LEDs next to the RS-232 ports are ON and GREEN if there is a proper hardware connection. Typically, Data Terminal Equipment (DTE) uses pin 3 of the DB9 connector to transmit; pin 2 to receive. Use RX1/TX1 for one device and RX2/TX2 for a second.

Step 6. Relay Connections: Wire the relays directly to the external device. There are 2 relays; each is single-pole-single-throw. Verify the load specifications of the external device do not exceed those of the relays listing in the specifications tables.

Step 7. Sensor Connections: There are 2 sensor inputs and 2 sensor power pins. One or both of the supplies can be used to power the external sensors.

Step 8. Configure the Controller: Download and install the Neutron Configuration software from the ALTINEX website, or refer to the Operation and TCP Control sections for programming and control of the Neutron Mini-Controller IR memory, setting RS-232 port properties, sensor trigger levels. A label is provided on the bottom of the unit to write in the IP and Port once programmed.

Step 9. Mount the Controller: The controller can be used as is, or mounted to a wall or shelf using the optional mounting bracket, AC101-401.

## 3. Limited Warranty/Return Policies

Please see the ALTINEX website at www.altinex.com for details on warranty and return policies.

### 4. Technical Specifications

Specifications are subject to change. See www.altinex.com for up-to-date information.

| Features/Description | CP500-110<br>Neutron Mini-Controller |
|---|---|
| General | |
| Connectors | |
| Power | 2-pin Terminal Block (1) |
| Ethernet | RJ-45 F (1) |
| Sensors | 5-pin Terminal Block (1) |
| Relays | 5-pin Terminal Block (1) |
| RS-232 | 5-pin Terminal Block (1) |
| IR | 5-pin Terminal Block (1) |
| Accessories Included | |
| Power Adapter | +12 VDC, 1.2 A |
| AC Cord, NEMA to NEC | 6 ft (2 m) |
| IR Emitters (2) | 6 ft (2 m) |
| CAT-6 Crossover Cable | 7 ft (2 m) |
| Hardware | - Hook & loop fastening tape with adhesive backing<br>- Terminal block connectors<br>- Cable ties |
| Optional Accessories | |
| Ceiling Occupancy Sensor | AC301-101 |
| IR Receiver | AC301-103 |
| Occupancy Sensor | AC301-105 |
| Mounting Bracket | AC101-401 |
| Power Controller, NEMA | AC301-102 |
| Power Controller, IEC | AC301-106 |
| Spare 5-pin Term. Block Connector | AC101-302 |
| Spare Power Adapter, 1.2 A | PS100-101 |
| Power Adapter, 4 A (power up to 4 CP500-110s) | PS100-102 |

Table 1. CP500-110 General

| Mechanical | CP500-110<br>Neutron Mini-Controller |
|---|---|
| Material/Color | 0.9 in Al / Black |
| Height | 1.0 in (25 mm) |
| Width | 5.4 in (137 mm) |
| Depth | 2.3 in (58 mm) |
| Weight | 0.5 lb (0.2 kg) |
| Shipping Weight (approx.) | 4.5 lb (2.0 kg) |
| T°Operating | 10℃-50℃ |
| T°Maximum | 75℃ |
| Humidity | 90% non-condensing |
| MTBF (calc.) | 38,000 hrs |

Table 2. CP500-110 Mechanical

| Electrical | CP500-110<br>Neutron Mini-Controller |
|---|---|
| Inputs | |
| Sensor Input | 0-24 VDC max. |
| IR IN | Pass-Through to IR1+ and IR2+ |
| Outputs | |
| Sensor Power, +12 VDC | +12 VDC, 0.15 A max. |
| Sensor Power, +24 VDC | +24 VDC, 0.15 A max. |
| Relays (capacity) | +24 VDC, 0.5 A max. |
| IR1 & IR2 Modulation | 38 or 56 kHz |
| IR Power, +12 VDC | +12 VDC, 0.15 A max. |
| Control | |
| Ethernet | 10/100 Base-T |
| RS-232 RX1/TX1<br>Rx2,/Tx2 | Baud Rates ....1200 to 57600<br>Data Bits ........8 or 9<br>8 Bit Parity .....None, Odd, or Even<br>Stop Bits ........1 or 2 |
| Power Consumption | |
| +12V | 1 A |
| Total Power | 12.0 W max. |

Table 3. CP500-110 Electrical

### 5. About Your CP500-110 Neutron Mini-Controller

- UDP and/or TCP Ethernet control
- 2 bidirectional RS-232 ports
- 2 control relays
- 2 sensor inputs
- 2 IR outputs
- IR pass-through and programmable IR memory
- Auto-discover units on LAN (local area network)

The Neutron Mini-Controller is an Ethernet based controller designed primarily for use with ALTINEX's MultiTouch family of touch panels for use in boardrooms, lecture halls, residential control systems, etc. The Neutron Mini is compact and sturdy allowing installation even in tight spaces. An optional mounting bracket is available for mounting the Neutron Mini to a shelf, wall, or other location.

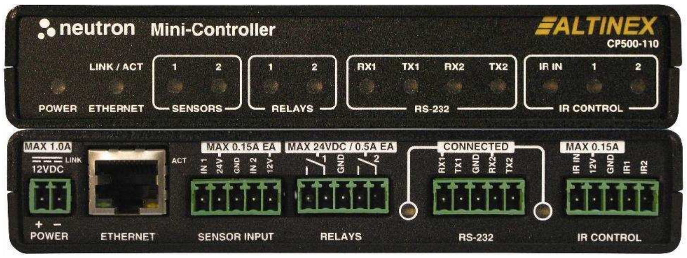
The Ethernet connection is a standard RJ-45 F connector that can be connected directly to a LAN or connected directly to a MultiTouch panel using the crossover cable provided. Using the LAN connection, the Neutron allows control from one or several MultiTouch panels. The remaining power, control, and communication connectors are all terminal blocks providing quick and flexible installation to a variety of equipment.

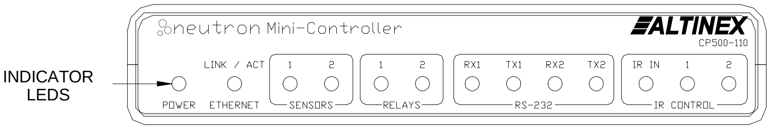
The Neutron Mini allows control of serial and IR devices over Ethernet. Additionally, there are two relays and two sensor inputs. The relays can be used to control or power external equipment independently over Ethernet. The sensor inputs trigger events based on voltage levels supplied by motion sensors, heat sensors, light sensors, RF sensors, etc. The sensor connector also provides two voltages for powering external sensors; +12VDC and +24VDC.

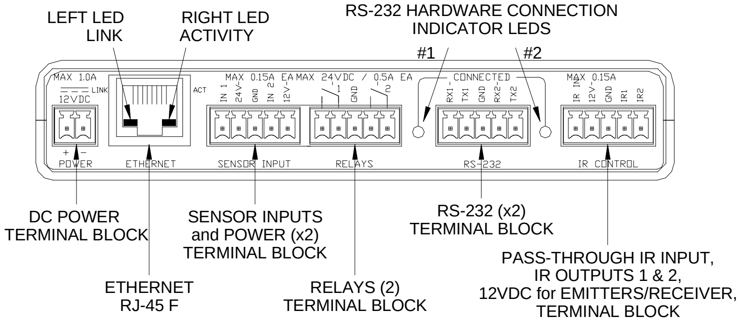
The Neutron is capable of supplying information to one or MultiTouch panels over Ethernet. Simply connect to the controller's static IP address in order to check the status of the relays (opened or closed), the input status of the sensor inputs (high or low) based on trigger level, or request RS-232 data from devices connected to the RS-232 ports. Only one TCP/IP connection is allowed at a time with the last connection taking control of the mini-controller. The Neutron can also be controlled using UDP broadcasts.

The mini-controller can store IR codes in its internal memory for direct control of external IR devices, or an IR room receiver like the ALTINEX AC301-103 can be used to pass-through IR signals from remote controls to devices connected to one or both of the controller's IR outputs.

Built-in auto-discover allows units to be detected on the LAN. This feature is an excellent troubleshooting tool and allows for identification of units improperly configured or not labeled during installation. See the Troubleshooting section for details.






LEFT LED LINK
RIGHT LED ACTIVITY
RS-232 HARDWARE CONNECTION INDICATOR LEDS
#1
#2
MAX 1.0A
12VDC
MAX 0.15A EA
MAX 24VDC / 0.5A EA
CONNECTED
MAX 0.15A
POWER
ETHERNET
SENSOR INPUT
RELAYS
RS-232
IR CONTROL
DC POWER TERMINAL BLOCK
SENSOR INPUTS and POWER (x2) TERMINAL BLOCK
RS-232 (x2) TERMINAL BLOCK
PASS-THROUGH IR INPUT, IR OUTPUTS 1 & 2, 12VDC for EMITTERS/RECEIVER, TERMINAL BLOCK
ETHERNET RJ-45 F
RELAYS (2) TERMINAL BLOCK

## 6. Application Diagrams

### Diagram 1: Typical Setup

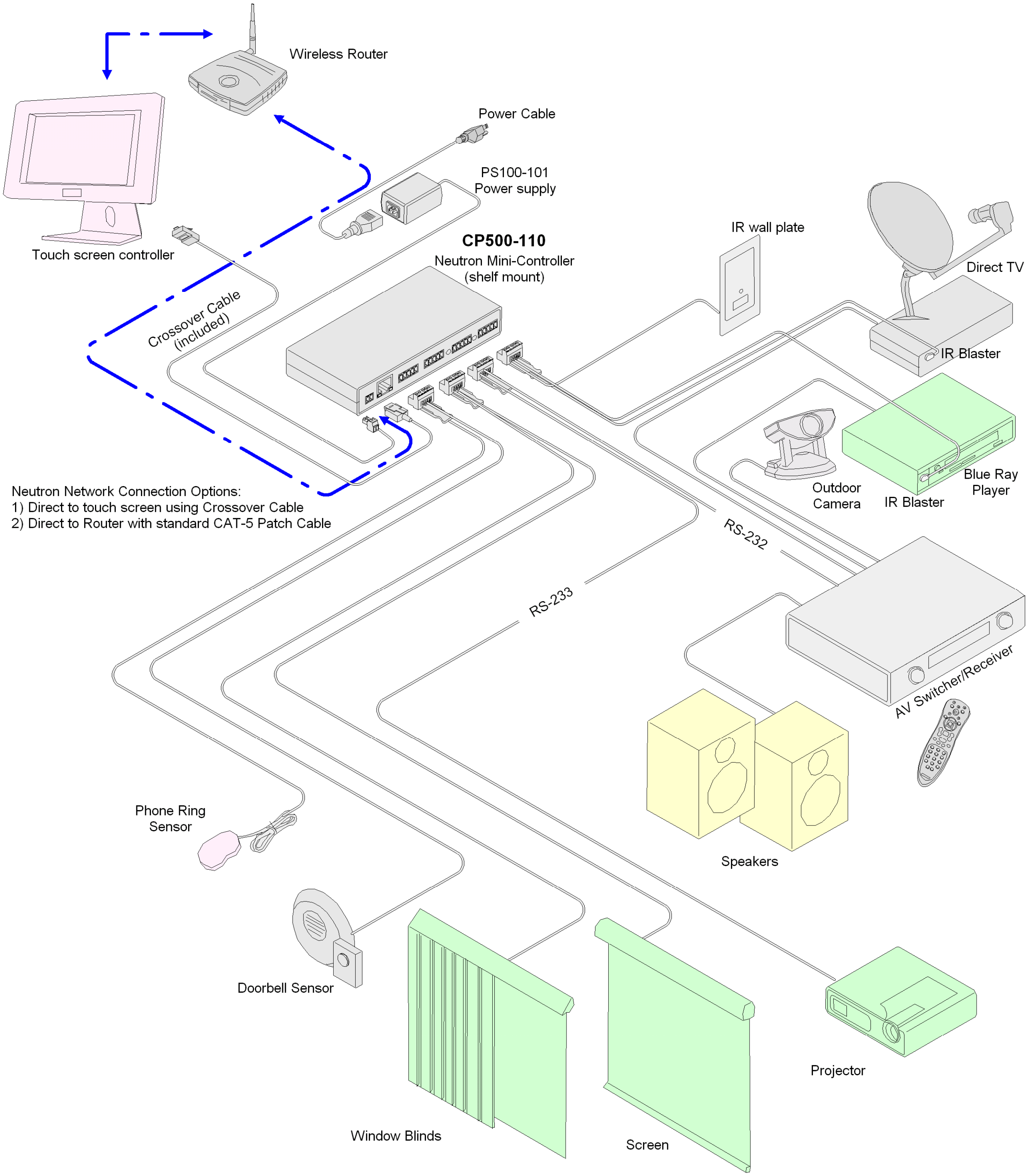


**Diagram 2: Internal View**

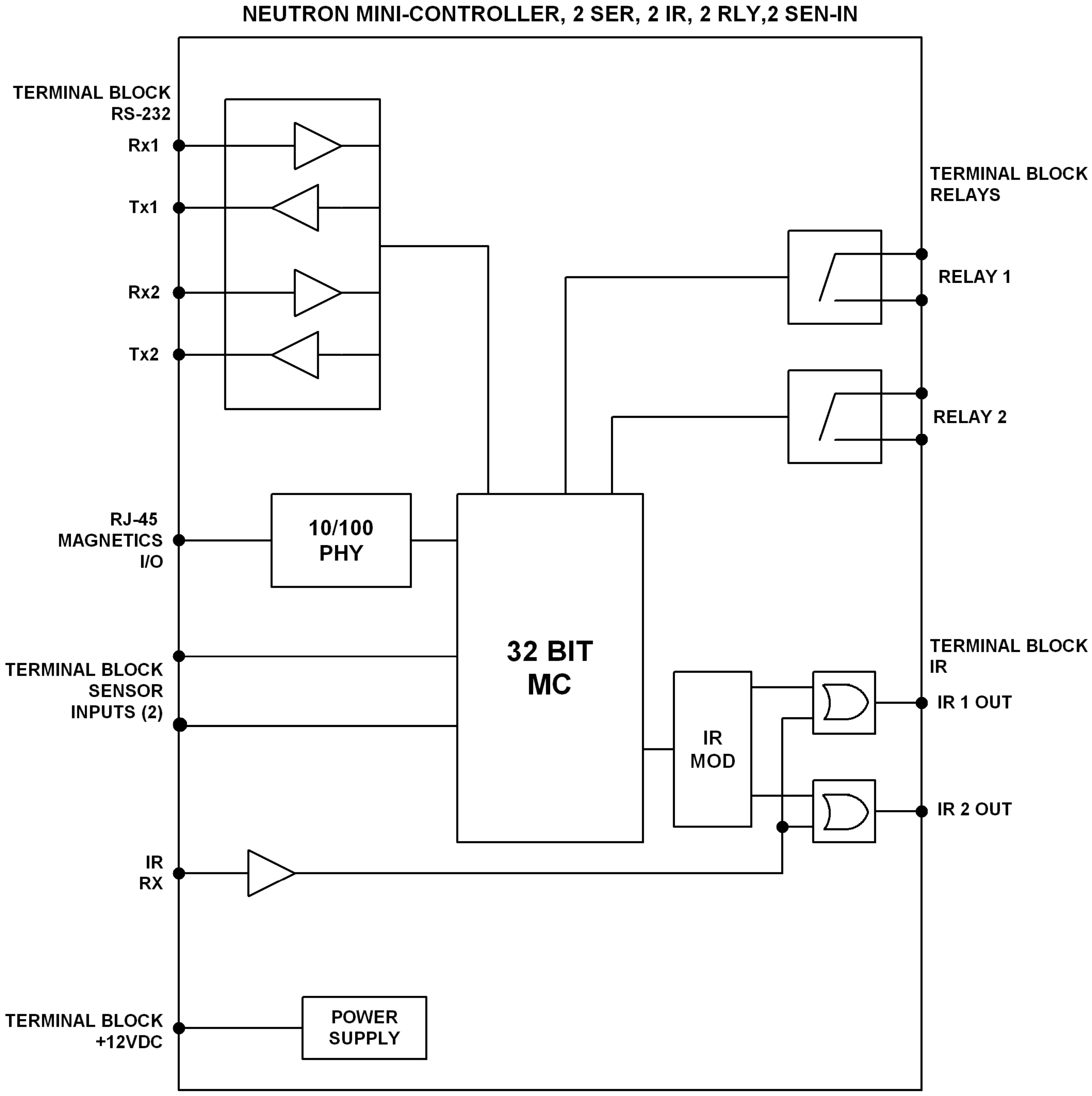
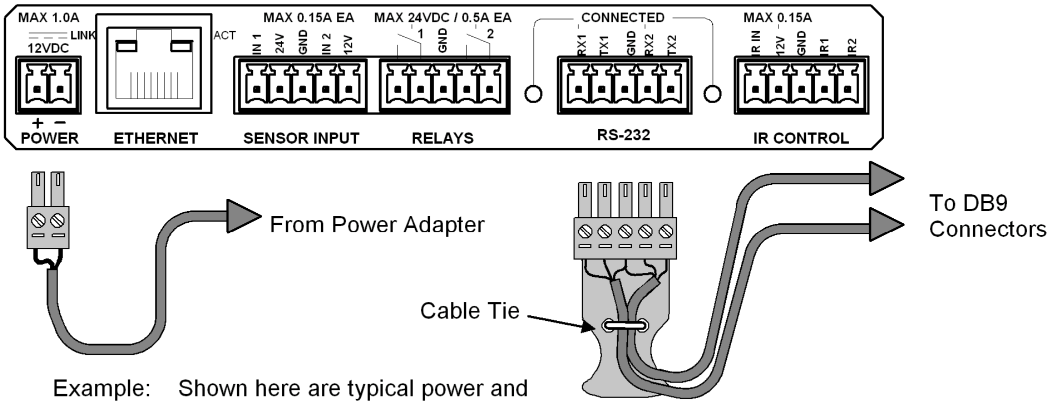
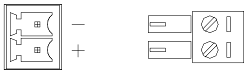


**Diagram 3: Terminal Block Connections**



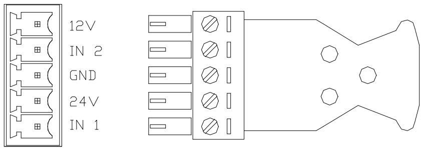
Example: Shown here are typical power and RS-232 connections.

## POWER



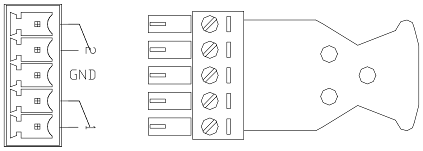
-V (GND) from power adapter
+V from power adapter (black lead with white text or stripe)

## SENSOR INPUT



Power for sensor 1 and/or 2: +12VDC.
Input for Sensor #2.
Ground
Power for sensor 1 and/or 2: +24VDC.
Input for Sensor #1.

## RELAYS

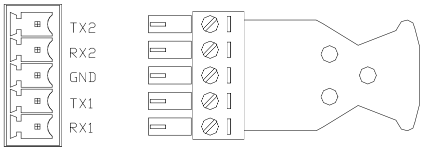


Relay #2 connections.

Ground

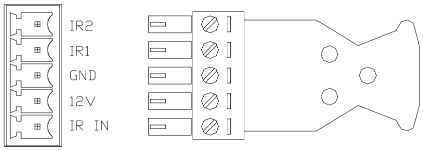
Relay #1 connections.

## RS-232



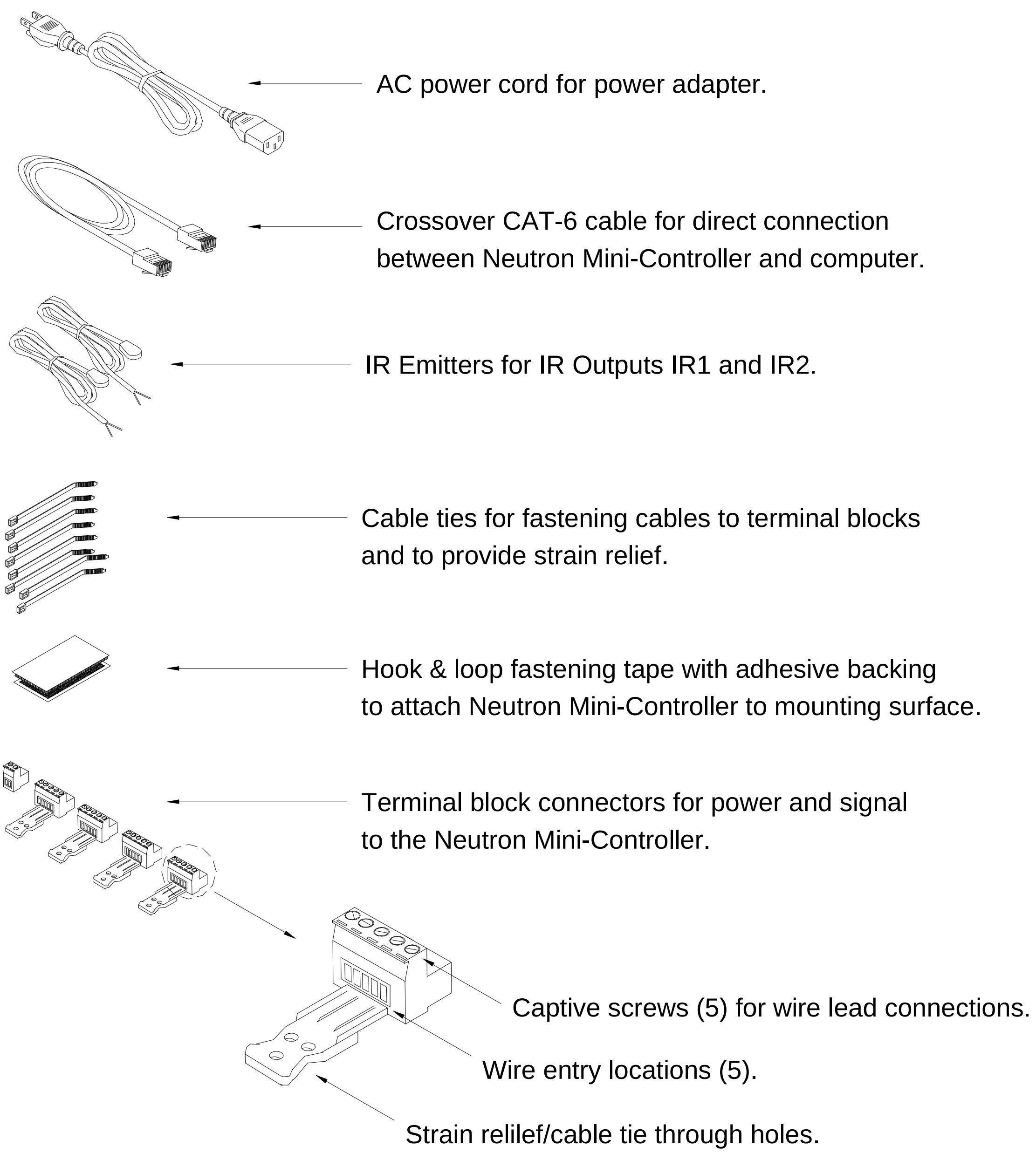
RS-232 port #2 receive connects to device transmit.
RS-232 port #2 transmit connects to device receive.
Ground
RS-232 port #1 transmit connects to device receive.
RS-232 port #1 receive connects to device transmit.

## IR CONTROL



IR port #2 output from IR IN above (pass-through) or from memory.
IR port #1 output from IR IN above (pass-through) or from memory.
Ground
Power for emitter or receiver if needed, +12VDC.
IR input for pass-through transmission to IR1 and IR2 outputs.

Diagram 4: Accessories



AC power cord for power adapter.

Crossover CAT-6 cable for direct connection between Neutron Mini-Controller and computer.

IR Emitters for IR Outputs IR1 and IR2.

Cable ties for fastening cables to terminal blocks and to provide strain relief.

Hook & loop fastening tape with adhesive backing to attach Neutron Mini-Controller to mounting surface.

Terminal block connectors for power and signal to the Neutron Mini-Controller.

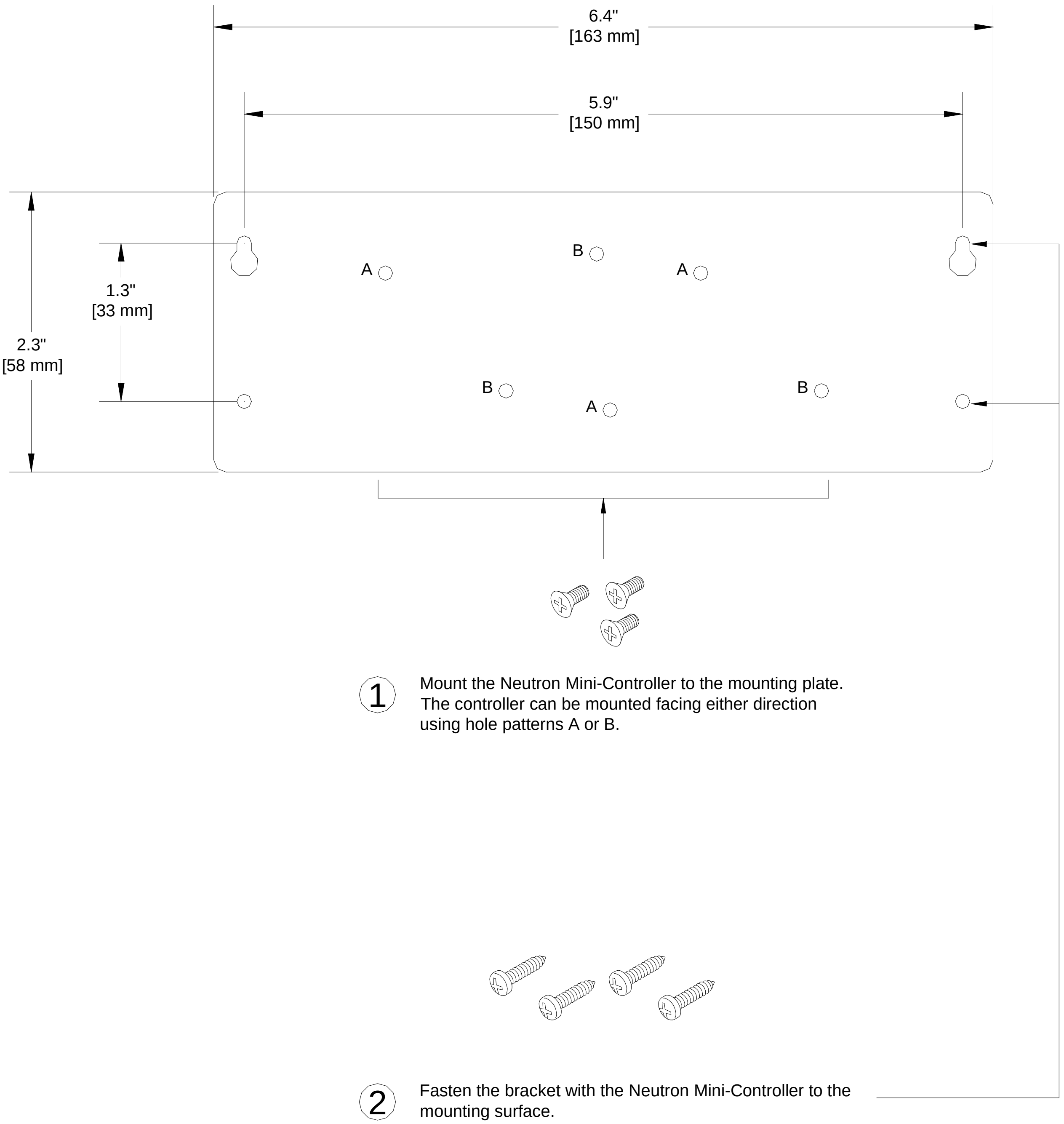
Captive screws (5) for wire lead connections.

Wire entry locations (5).

Strain relilef/cable tie through holes.

**Diagram 5: Mounting Bracket (optional)**

## Optional Mounting Bracket
## Part No. AC101-401



1 Mount the Neutron Mini-Controller to the mounting plate. The controller can be mounted facing either direction using hole patterns A or B.

2 Fasten the bracket with the Neutron Mini-Controller to the mounting surface.

## 7. Operation

### 7.1 Default Settings

After the initial installation and set up, the Neutron Mini-Controller does not require adjustments for optimum performance and will work trouble-free without user intervention.

Following are the default settings of each Neutron Mini-Controller:

| | |
|---|---|
| IP ........................ | 192.168.1.80 port 23 |
| Subnet ................ | 255.255.255.0 |
| Gateway ............ | 192.168.1.1 |
| RS-232 Ports:...... | 9600 baud,<br>8 data bits+ 1 stop bit,<br>No parity |
| Sensors: .............. | Trigger Level 18V<br>De-bounce 250 ms<br>LED Operation Normal |
| Relays: ............... | Open |
| IR Ports: ............ | Modulation 38 kHz (IR1 and IR2)<br>All Memory Blank (FFs) |

### 7.2 Security

Contact your network administrator about limiting access to the Neutron Mini-Controller on your network.

### 7.3 Front Side Indicators

There are several LED indicators on the front of the controller to indicate the status of the controller.

Power This LED is ON when power is applied.

Ethernet This LED is ON and solid if a link is present, and flashes to indicate activity.

Sensors These LEDs turn ON when the voltage on a sensor input is ABOVE its trigger level.

Each LED can be set for inverted operation to turn ON when the sensor input is BELOW the trigger level. See the TCP Control section for setting properties.

Relays The relay LEDs are ON when a relay contact is CLOSED, and OFF when a relay contact is OPEN.

RS-232 The RS-232 LEDs flash indicating transmit and receive traffic on the RS-232 control lines.

IR Control These LEDs indicate activity on the IR outputs and pass-through input. IR data passing through the IR IN pin flashes the IN LED and the IR1 and IR2 LEDs flash when IR data is output on the IR1 and IR2 outputs.

### 7.4 Rear Side Indicators

The LED indicators on the rear of the controller show the status of the RS-232 and Ethernet connections.

Ethernet There are 2 LEDs on the Ethernet input connector. The LEFT LED indicates an Ethernet LINK is present. The RIGHT LED indicates ACTIVITY on the transmit/receive lines.

RS-232 These LEDs are ON when a COM port has a proper hardware connection to an RS-232 device.

### 7.5 Input Power

The power supplied from the power adapter is always on as long as there is an AC voltage present on the adapter's supply side. It is not necessary to power down the controller unless it will not be used for an extended period.

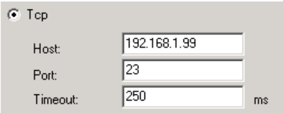
### 7.6 Output Power

The Neutron Mini-Controller provides DC voltages for powering external devices. Check the specifications of the external devices and make sure they do not exceed the output ratings on the Neutron.

### 7.7 Ethernet

The Neutron Mini-Controller works on both 10 and 100 Base-T networks and supports the DOS "ping" command for use in testing network operation and to verify the IP is reachable.

Connect to the Neutron Mini-Controller using AVSnap to make a TCP connection using the following settings in AVSnap's communication mode, or within an application using a TCP object.



● Tcp

Host: 192.168.1.99

Port: 23

Timeout: 250 ms

The static IP, port, subnet, and gateway can be changed but it is recommended the port remain fixed at 23. See the TCP Control section for details on changing the Ethernet properties. Reference commands <<CSN>>, <<CSP>>, <<CSB>>, <<CSG>> and <<CSO>>.

### 7.8 IR Control

IR devices can be controlled using 3 different methods with the Neutron Mini-Controller. The first method uses an IR receiver like the ALTINEX AC301-103 connected to the IR IN pin of the controller. This method allows a user to aim a remote control at a wall-mount or other receiver and have the remote control's signal redirected to the IR emitters connected to the IR1 and IR2 pins.

The next 2 methods send IR data from internal memory or RAM to IR1 or IR2 and allow the MultiTouch to send IR command strings from ALTINEX's IR library directly to the Neutron Mini-Controller using AVSnap. These command strings are stored on the computer application side, not within the Neutron.

The third method uses IR commands stored in the Neutron's internal memory to control the external devices. This method only requires the control application to tell the controller to recall a command stored in a specific memory location. The IR commands can be programmed into memory directly from ALTINEX's IR library using AVSnap.

See the TCP Control section for details on setting up and using the IR outputs. Reference commands <<CSI>>, <<CRI>>, <<COI>>, <<CSM>>, <<CRM>>, <<COP>>, and <<CSF>>.

### 7.9 Relays

Inside the Neutron are 2 relays that can be used to control or trigger devices like projector screens.  The relays can also be used to route power to low-power DC devices, but make sure to verify the load requirements against the relay specifications.

The state of each relay can be recalled by a control application in order to determine the state of external devices connected to the relays. See the TCP Control section for details on using the relay outputs. Reference commands <<COR>>, <<CTR>>, and <<CRR>>.

### 7.10 RS-232 Control

The Neutron Mini-Controller has 2 bidirectional RS-232 COM ports available to communicate with virtually any serial device. Each port can be individually set with different baud rates, parity, stop bits, and data bits allowing a control application to operate different devices from a single remote location through the TCP connection over Ethernet.

See the TCP Control section for details on setting up and using the RS-232 COM ports. Reference commands <<COU>> and <<CSU>>.

### 7.11 Sensors

The Neutron Mini-Controller is designed to work with a wide variety of sensors including motion sensors, IR detectors, RF detectors, etc. ALTINEX offers the following optional sensors that work for the most common applications and provide excellence performance.

AC301-101 - Ceiling Occupancy Sensor
AC301-103 - IR Receiver
AC301-105 - Occupancy Sensor

The Neutron Mini-Controller has 2 sensor inputs that accept sensor output voltages up to 24 VDC. Each sensor input has an independent trigger level setting that can be used to determine if a sensor has been "tripped" or "activated" based on movement, occupancy, etc. depending on the type of sensor. Each sensor also has an individually programmable de-bounce time that can be used to prevent false sensor "trips" being detected by the control application.

The control application does not need to query the status of the sensor inputs to determine if the trigger level has been reached. The Neutron automatically sends any status change to the host application if a TCP/IP connection is available, otherwise the application must periodically connect to the Neutron and request the sensor status manually. The application can then take actions based on the status of the sensor.

See the TCP Control section for details on setting up and using the sensor inputs. Reference commands <<CST>>, <<CSD>>, <<CSL>>, <<CRL>>, and <<CRS>>.

### 7.12 TCP Control

The Neutron Mini-Controller has many capabilities using an ALTINEX MultiTouch panel and running AVSnap application software. The TCP connection between the PC and the Neutron is made over the local area network (LAN) or directly using a crossover CAT-5 cable.

ALTINEX MultiTouch panels and a wireless router can be used to create a wireless network to run the control application. The Neutron is connected directly to the router and the touch panels communicate directly through the router to the Neutron.

### 7.12.1 Initial TCP Connection

Each Neutron is shipped with the following Ethernet settings:

| | |
|---|---|
| Static IP | 192.168.1.80 |
| Subnet Mask | 255.255.255.0 |
| Gateway | 192.168.1.0 |
| Port | 23 |

If these settings are unreachable or conflict with another device on the network, configure your PC with a fixed IP on the same network as the Neutron; for example, 192.168.1.100. (See Figure 1 below for the TCP/IP property page in Windows®.) Use the crossover cable provided to establish the initial connection and make the setting changes necessary to communicate directly with the Neutron over the existing network.

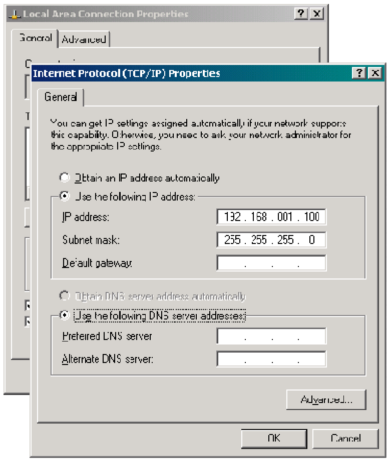
Next, use the following commands to configure the Neutron's IP settings to match those provided to you by your IT administrator:

<<CSN>>, <<CSP>>, <<CSB>>, <<CSG>>, <<CSO>>, <<CRC>>

### 7.12.2 TCP Interface

Control commands for the CP500-110 are in a simple ASCII format.

1. Triangle brackets "<< >>" are part of the command.
2. Use uppercase letters for all commands.
3. Spaces are NOT legal characters.



Local Area Connection Properties

Internet Protocol (TCP/IP) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:
IP address: 192 . 168 . 001 . 100
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:
Preferred DNS server: . . .
Alternate DNS server: . . .

Figure 1.

### 7.13 About TCP and UDP Communication

UDP Operation
UDP commands are available for units with firmware version 690-0332-004 and greater.

Make a standard TCP connection with the Neutron using AVSnap or other communication software (default 192.168.1.80 port 23) and send the command <<VER>> to check the firmware version.

If you need UDP operation, contact ALTINEX Technical Support about a firmware upgrade.

### 7.13.1 Command Acknowledgement

All control/configuration commands start with "C" in order to differentiate them from the status feedback strings that start with "S." Complete commands and feedback are enclosed in "<< >>" brackets.

Command Acknowledgement Feedback - TCP Connection ONLY
Control commands that do not generate feedback from the Neutron receive an asterisk ( * ) as feedback to indicate a command was processed. UDP commands do not generate this feedback.

### 7.13.2 Feedback Request Description

Status Feedback - TCP Connection
All status feedback from the Neutron is enclosed in triangle brackets and prefixed with "S" as is in the following sample:

Send: <<CRF>>

Receive: <<SRF,4001,6201,632,97,0,0,0,1,0,0,0,0>>

The above sample feedback provides information in two parts. The first part is the information prefix, or "SRF" telling the control application that the data to follow is the Neutron's status.

Status Feedback - UDP
UDP feedback from the Neutron is enclosed in triangle brackets and prefixed with "S" the same as TCP connection feedback, but includes the source IP address as is in the following sample:

Send: <<CRF>>

Receive: <<SRF,4001,6201,632,97,0,0,0,1,0,0,0,0,**192.168.1.80**>>

### 7.13.3 Unsolicited Feedback

The Neutron provides feedback for the following unsolicited events not related to a command sent to the controller:

1) Sensor input goes above or below the trigger level

   If there is a TCP connection to the Neutron, the Neutron sends the feedback to the connected device. Additionally, the Neutron broadcasts the same feedback using UDP protocol. The UDP feedback also includes the source IP address.

2) Data is received on one of the RS-232 ports

   If there is a TCP connection to the Neutron, the Neutron sends the feedback to the connected device but does not broadcast. If there is no TCP connection the Neutron broadcasts using UDP protocol.

TCP Connection Example:

The input for Sensor #1 goes above its trigger level and the Neutron sends the following using the connection IP and port number:

<<SRS11>>

UDP Connection Example

The input for Sensor #1 goes above its trigger level and the Neutron broadcasts the following:

<<SRS11,192.168.1.80>>

An AVSnap TCP object configured as a UDP Server (see the next section) catches the data broadcast by the Neutron.

### 7.14 Configuring TCP and UDP Objects in AVSnap

The AVSnap TCP object can be configured for TCP, UDP, or UDP Server communication depending on the needs of the application. In the case of the Neutron, the following 3 objects are used.

Object 1 - STANDARD TCP

The first option is to create a standard TCP object that is used as standard TCP connection to a fixed port (default 23) on the Neutron. This method maintains a constant connection to the Neutron to send and receive data.

Object 2 - UDP BROADCAST

The next option is to create a TCP object and set its properties to broadcast to IPs on a specific port number. This method allows the application to send UDP commands to be executed by any or all Neutron on the network without having to establish a TCP connection. In effect, it is a send it and forget it type of control. The host can be set to send UDP commands to various combinations of IP address. For example:

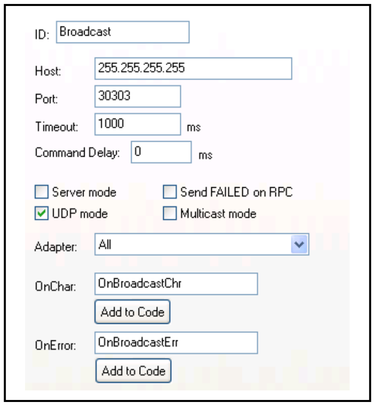
| Host | Port | Broadcast Range |
|---|---|---|
| 255.255.255.255 | 30303 | All IPs |
| 192.168.1.255 | 30303 | 192.168.1.1 - 192.168.1.254 |
| 192.168.1.81 | 30303 | 192.168.1.81 |

Object 3 - UDP SERVER

The third option is to set the TCP object as a UDP Server. In server mode, the TCP object captures all broadcast data (255.255.255.255) sent to port 30303 from the Neutron controller. This data include RS-232 port data, sensor triggers, etc.

## BROADCASTING

Broadcast UDP data using the AVSnap TCP Object.



ID: Broadcast
Host: 255.255.255.255
Port: 30303
Timeout: 1000 ms
Command Delay: 0 ms
☐ Server mode ☐ Send FAILED on RPC
☑ UDP mode ☐ Multicast mode
Adapter: All
OnChar: OnBroadcastChr
OnError: OnBroadcastErr

(1) Create a TCP object and configure its properties for UDP BROADCAST as shown.

(2) Establish a UDP connection by adding the following line of code:

```
Broadcast.Connect;
```

(3) Broadcast the desired data. For example, send "<<CRF>>" to check the status of Neutrons on the network.

```
Broadcast.writeStr('<<CRF>>');
```

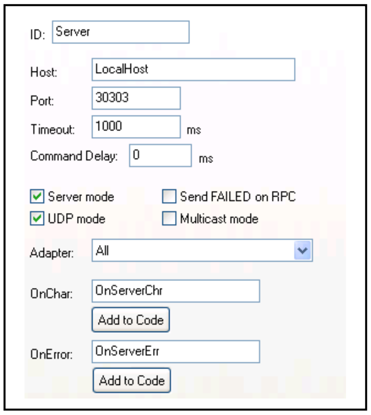
The command "<<CRF>>" is sent to the Neutrons on the network. Any Neutrons on the network respond with their status data, which includes their IP address. The data received from the Neutrons is received by the server object in the next example.

(4) Close the connection by sending the following code:

```
Broadcast.Disconnect;
```

## UDP SERVER

Receive UDP data broadcasts from controllers using the AVSnap TCP Object configured as a UDP Server.



ID: Server
Host: LocalHost
Port: 30303
Timeout: 1000 ms
Command Delay: 0 ms
☑ Server mode ☐ Send FAILED on RPC
☑ UDP mode ☐ Multicast mode
Adapter: All
OnChar: OnServerChr
OnError: OnServerErr

(1) Create a TCP object and configure its properties for UDP SERVER as shown.

(2) Start the server by adding the following code:

```
Server.StartServer;
```

(3) Add the procedure **OnServerChr** to capture data from UDP transmissions.

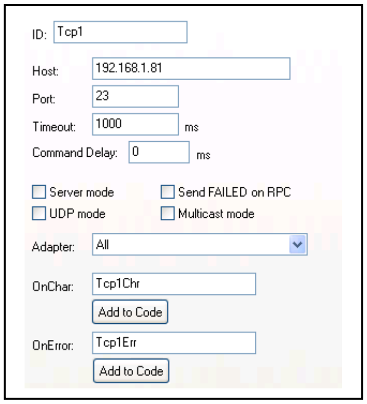
```
procedure OnServerChr(Obj:TObj; Count:integer);
var s:string;
begin
   UdpS.ReadStr(s, Count);
   // Do something with the data received.
end;
```

(4) Stop the server (if needed) by adding the following code:

```
Server.StopServer;
```

## STANDARD TCP

Transmit and receive data while maintaining a standard TCP connection.



ID: Tcp1
Host: 192.168.1.81
Port: 23
Timeout: 1000 ms
Command Delay: 0 ms
☐ Server mode ☐ Send FAILED on RPC
☐ UDP mode ☐ Multicast mode
Adapter: All
OnChar: Tcp1Chr
OnError: Tcp1Err

(1) Create a TCP object and configure its properties for TCP CONNECTION as shown.

(2) Establish a TCP connection by adding the following line:

```
Tcp1.Connect;
```

(3) Create the procedure **Tcp1Chr** to capture the data received through the TCP connection.

```
procedure Tcp1Chr(Obj:TObj; Count:integer);
var s: string;
begin
   Tcp1.readstr(s,count);
   // Do something with the data received.
end;
```

(4) Close the TCP connection by adding the following line:

```
Tcp1.Disconnect;
```

### 7.15 Description of Commands

The following commands are supported through the Ethernet communication port on the Neutron Mini-Controller.

Note: All information received from the Neutron's RS-232 ports is sent directly to the Ethernet port without analyzing the data.

1. **<<CSU>>**

Configure the RS-232 ports for baud rate, parity, number of data bits, and number of stop bits. Once configured, data sent to or received on a given port is in the set format.

| | | |
|---|---|---|
| TCP Command Format: | <<CSUnbpms>> | (default port 23) |
| UDP Command Format: | <<CSUnbpms>> | (port 30303) |

n – RS-232 port number (1 or 2)

b – baud rate (1 ~ 7)

1 – 1200bps
2 – 2400bps
3 – 4800bps
4 – 9600bps
5 – 19200bps
6 – 38400bps
7 – 57600bps

p – parity for 8-bit data (0 ~ 2)

0 – None    1 – Odd    2 – Even

m – data bits (0 or 1)

0 – 8 data bits    1 – 9 data bits

s – stop bits (1 or 2)

1 – 1 stop bit    2 – 2 stop bits

Example:

Configure the controller's RS-232 port #1 for 19200 baud, 8 data bits, 1 stop bit, and no parity by sending the following to the controller:

<<CSU15001>>

Once complete, the Neutron sends confirmation.

* (TCP only)

2. **<<COU>>**

Send a string of data through one of the RS-232 ports. Regardless of the TCP settings, the data sent out of an RS-232 ports is sent using the baud rate, parity, stop, and data bits for which the port is configured. The default configuration for both RS-232 ports is as follows:

Baud Rate......... 9600    Data Bits........... 8
Parity ............... None    Stop Bits........... 1

| | | |
|---|---|---|
| TCP Command Format: | <<COUnddd>> | (default port 23) |
| UDP Command Format: | <<COUnddd>> | (port 30303) |

n = UART port number (1 or 2)

ddd = data to be transmitted (ASCII characters)
Use % to transmit HEX value. (ex: %0D for carriage return)

Example:

Output the following string on RS-232 port #1:

[VER]

by sending the following to the controller:

<<COU1%5BVER%5D>>

3. **<<SRU>>**

This is unsolicited feedback-only which is sent to the Ethernet port when RS-232 data is received on either of the Neutron's RS-232 ports. The data is not analyzed; it is simply relayed to the Ethernet port for use by the MultiTouch control application and may be received in one or more groups depending on how the data is received and the length of the data string.

| | |
|---|---|
| TCP Feedback Format: | <<SRUnddd>> |
| UDP Feedback Format: | <<SRUnddd,*sourceIP*,>> |

n = RS-232 port number (n=1 or 2)

ddd = Data received through RS-232 port

sourceIP = The IP address of Neutron (ex: 192.168.1.80)

Example:

RS-232 port #1 of the Neutron is connected to a projector in a classroom. In addition to controlling the projector from the MultiTouch, the MultiTouch application can be used to respond to data received from the projector; for example "TEMP1" indicating its internal temperature is too high.

The Neutron receives the data on port #1's receive line (RX1) and then sends the following, or similar, to the Ethernet port:

TCP <<SRU1TEMP1>>
UDP <<SRU1TEMP1,192.168.1.80,>>

The data could also be transmitted in any of the following 3 examples depending on how the data is received and the length of the string:

1. <<SRU1TE>><<SRU1MP1>>
2. <<SRU1TE>><<SRU1MP>><<SRU11>>
3. <<SRU1T>><<SRU1E>><<SRU1MP>><<SRU11>>

The MultiTouch application must receive the above response(s) and take action based upon the message "TEMP1. If additional information is required, the control application could then request the Neutron's alias. In this case, the alias is "Room 502."

Next, the control application could be use to send an email alert to the maintenance department to make sure the projector air vents are clear.

4. **<<CSI>>**

Save an IR command string from ALTINEX's IR library into the Neutron's internal memory. A single IR command can take up to 4 save commands to save the entire command string. After each command is processed, the Neutron will return an asterisk ( * ) as feedback indicating it has finished processing the previous command.

| | | |
|---|---|---|
| TCP Command Format: | <<CSInpddd>> | (default port 23) |
| UDP Command Format: | *Not Available* | |

n = IR memory location number (01 to 99, 2-digits)

p = Save instructions (1=Overwrite, 0=Append)

ddd = IR data in ASCII characters (250 per command),
Valid characters are "0123456789ABCDEF"

Example:

A typical IR library command string is 1,000 characters and requires a minimum of 4 save commands to save the IR string into memory. The following command samples are used to save an IR string into memory location "01". Remember to wait for acknowledgement from the Neutron after sending each command before sending the next.

| | |
|---|---|
| <<CSI011FFFFFFFFFF...*250*>> | Overwrite Mem 01 with 250 char. |
| * | Feedback from the Neutron. (TCP) |
| <<CSI010AAA000FFF...*250*>> | Append Mem 01 with 250 char. |
| * | Feedback from the Neutron. (TCP) |
| <<CSI010FFFF000FFF...*250*>> | Append Mem 01 with 250 char. |
| * | Feedback from the Neutron. (TCP) |
| <<CSI010FFFFFFFFFF...*250*>> | Append Mem 01 with 250 char. |
| * | Feedback from the Neutron. (TCP) |

5. **<<CRI>>**

Read IR data from memory.

| | | |
|---|---|---|
| TCP Command Format: | <<CRIn>> | (default port 23) |
| UDP Command Format: | <<CRIn>> | (port 30303) |

n = IR memory location number (01 to 99, 2-digits)

| | |
|---|---|
| TCP Feedback Format: | <<SRIddd>> |
| UDP Feedback Format: | <<SRIddd,sourceIP>> |

ddd = IR data in 1,000 ASCII characters

sourceIP = The IP address of Neutron (ex: 192.168.1.80)

Example:

Send <<CRI21>> to read the IR data string in memory location 21.

For TCP connections, the Neutron responds with a single data block in the format "<<SRI" plus 1000 data characters and concludes with ">>".

<<SRIFFFFFFFFFFFFFF0000000000FFFF...>>

UDP connections send 10 data blocks with 100 data characters each.

<<SRI01,1, FFFFFFFFFFFFFF0000000000FFFF...,192.168.1.80>>
<<SRI01,2, FFFFFFFFFFFFFF0000000000FFFF...,192.168.1.80>>
etc.
<<SRI01,10, FFFFFFFFFFFFFF0000000000FFFF...,192.168.1.80>>

6. **<<COI>>**

Output the IR data stored in a memory location to one of the IR outputs in order to activate the IR emitter and control an external device.

| | | |
|---|---|---|
| TCP Command Format: | <<COInmm>> | (default port 23) |
| UDP Command Format: | <<COInmm>> | (port 30303) |

n = IR output number (n= 1 or 2)

mm = IR memory location number (01 to 99, 2-digits)

Example:

Output the IR string stored in memory location 1 on IR output port IR2 by sending the following command:

<<COI201>>

The entire IR string is read from memory and then sent to the IR port with this single command. Once the entire process is complete, the Neutron sends confirmation.

* (TCP only)

7. **<<CSM>>**

Preset the pass-through IR by loading the data into RAM from the ALTINEX library directly from the MultiTouch controller. This command allows virtually unlimited IR commands to be used in place of, or in addition to, the built-in IR memory of the Neutron. This command is conjunction with the <<COP>> command that instructs the controller to output the data to the IR ports.

| | | |
|---|---|---|
| TCP Command Format: | <<CSMpddd>> | (default port 23) |
| UDP Command Format: | <<CSMpddd>> | (port 30303) |

p = Saving instructions (1=Overwrite, 0=Append)

ddd = IR data in ASCII characters (250 per command),
Valid characters are "0123456789ABCDEF"

Example:

A typical IR library command string is 1,000 characters and requires a minimum of 4 save commands to save the IR string into RAM. The following command samples are used to load an IR string into RAM. Remember to wait for acknowledgement from the Neutron after sending each command before sending the next.

| | |
|---|---|
| <<CSM1FFFFFFFFFF...*250*>> | Overwrite RAM with 250 char. |
| * | Feedback from the Neutron. (TCP) |
| <<CSM0AAA000FFF...*250*>> | Append RAM with 250 char. |
| * | Feedback from the Neutron. (TCP) |
| <<CSM0FFFF000FFF...*250*>> | Append RAM with 250 char. |
| * | Feedback from the Neutron. (TCP) |
| <<CSM0FFFFFFFFFF...*250*>> | Append RAM with 250 char. |
| * | Feedback from the Neutron. (TCP) |

8. **<<CRM>>**

Extract IR data stored in a memory location and load it into RAM for faster response time when needed. This command is conjunction with the <<COP>> command that instructs the controller to output the data to the IR ports.

| | | |
|---|---|---|
| TCP Command Format: | <<CRMmm>> | (default port 23) |
| UDP Command Format: | <<CRMmm>> | (port 30303) |

m = IR memory location number (01 to 99, 2-digits)

Example:

Load the IR data from memory location 02 into RAM by sending the following command:

<<CRM02>>

The entire IR string is read from memory and loaded into RAM. Once complete, the Neutron sends confirmation.

*   (TCP only)

9. **<<COP>>**

Output the IR data previously preloaded in RAM from 1 to 9 times. This command is especially useful for commands such and volume, brightness, etc. where single step increments take too long to make large adjustments.

| | | |
|---|---|---|
| TCP Command Format: | <<COPnp>> | (default port 23) |
| UDP Command Format: | <<COPnp>> | (port 30303) |

n = IR output number (n= 1 or 2 for outputs IR1+ or IR2+)

p = Number of times to output data (p= 1 to 9)

Example:

The IR volume UP command for a TV has been recalled from memory and loaded into RAM in order to more quickly adjust the volume. The control application has been programmed send the volume command 3 times every time the volume UP key in the application is pressed. Send the volume UP command 3 times by sending the following:

<<COP3>>

The entire IR string is sent in rapid succession to the IR outputs. Once complete, the Neutron sends confirmation.

*   (TCP only)

10. **<<CSF>>**

Set the IR output modulation frequency to 38 or 56 kHz for outputs IR1+ and IR2+ independently.

| | | |
|---|---|---|
| TCP Command Format: | <<CSFnp>> | (default port 23) |
| UDP Command Format: | <<CSFnp>> | (port 30303) |

n = IR output number (n= 1 or 2 for outputs IR1+ or IR2+)
p = IR frequency (0 = 38 kHz, 1 = 56 kHz)

Example:

Set the IR modulation frequency of IR output 2 to 38 kHz by sending:

<<CSF20>>

Once complete, the Neutron sends confirmation.

*   (TCP only)

11. **<<CST>>**

Independently set the trigger level for each of the sensor inputs. The trigger levels can be set from 0 to 24 V in 27 mV steps. The value must be entered as 3-digits. Each time the sensor input transitions above or below the trigger level the Neutron sends <<SRS>> feedback:

| | |
|---|---|
| <<SRS10>> | Input 1 - HIGH to LOW transition occurred. |
| <<SRS11>> | Input 1 - LOW to HIGH transition occurred. |
| <<SRS20>> | Input 2 - HIGH to LOW transition occurred. |
| <<SRS21>> | Input 2 - LOW to HIGH transition occurred. |

| | | |
|---|---|---|
| TCP Command Format: | <<CSTnv>> | (default port 23) |
| UDP Command Format: | <<CSTnv>> | (port 30303) |

n = Sensor Input number (n= 1 or 2)

v = Trigger level (v=# from 000 to 999, default= 666 or @ 18 V)

Example:

Set the trigger level for sensor input #2 to approximately 2.5 V by sending the following to the Neutron:

<<CST2093>>

If 3 digits are NOT sent, zeros are appended to the value sent. In this case, if only "93" was sent instead of "093" the trigger level would be set to 930 instead of 93. Once the trigger level is saved, the Neutron sends confirmation.

*   (TCP only)

12. **<<CSD>>**

Independently set the de-bouncing time for a sensor input giving the signal change time to settle before triggering an event. The de-bounce time is set in multiples of 25 ms with a maximum of 0.5 s.

| | | |
|---|---|---|
| TCP Command Format: | <<CSDnv>> | (default port 23) |
| UDP Command Format: | <<CSDnv>> | (port 30303) |

n = Sensor Input number (n= 1 or 2)

v = De-bounce time (n= # from 0 to 20, default: 10 or 250 ms)

Example:

Set the de-bounce time for sensor input #2 to 100 ms by sending the following to the controller:

<<CSD24>>

Once the de-bounce time is saved, the Neutron sends confirmation.

*   (TCP only)

### 13. <<CSL>>

Set the polarity for the sensor input LED operation. In normal operation, the LED is ON when the sensor input is above the trigger level and OFF when below the trigger level. In inverting operation, the LED is ON when the sensor input is below the trigger level and OFF when above the trigger level.

| | | |
|---|---|---|
| TCP Command Format: | <<CSLnp>> | (default port 23) |
| UDP Command Format: | <<CSLnp>> | (port 30303) |

n = Sensor Input number (n= 1 or 2)

p = Polarity (p= 0 or 1)

0 – Normal LED operation (default)

1 – Inverting LED operation

Example:

Set the LED polarity for sensor input #2 to inverting operation by sending the following to the controller:

<<CSL21>>

Now when there is nothing detected on sensor input #2, its LED will be ON. Once the LED polarity is saved, the Neutron sends confirmation.

* (TCP only)

### 14. <<CRS>>

Read the status of the sensor inputs. A "0" indicates the voltage from the sensor is below the trigger level and a "1" indicates the voltage is above the trigger level regardless of the LED polarity setting.

The Neutron also sends the status automatically when there is a change in state.

| | | |
|---|---|---|
| TCP Command Format: | <<CRSn>> | (default port 23) |
| UDP Command Format: | <<CRSn>> | (port 30303) |

n = Sensor Input number (n= 1 to 8)

| | |
|---|---|
| TCP Feedback Format: | <<SRSns>> |
| UDP Feedback Format: | <<SRSns,*sourceIP*>> |

n = Sensor Input number (n= 1 to 8)

s = Sensor status (n= 0 or 1)

0 - Sensor input is below trigger level
1 - Sensor input is above trigger level

sourceIP = The IP address of Neutron (ex: 192.168.1.80)

Example:

The trigger level for sensor input #2 is set to 93, or about 2.5 VDC. The output of sensor #2 is 4.5 VDC. Send the following to read the status:

<<CRS2>>

The Neutron responds with the following:

<<SRS21>> UDP = <<SRS21,192.168.1.80>>

The "1" indicates the voltage level on the sensor input is above the trigger level voltage of 2.5 VDC.

### 15. <<CRL>>

Read the current sensor level on a sensor input with each step representing approximately 27 mV.

| | | |
|---|---|---|
| TCP Command Format: | <<CRLn>> | (default port 23) |
| UDP Command Format: | <<CRLn>> | (port 30303) |

n = Sensor Input number (n= 1 to 8)

| | |
|---|---|
| TCP Feedback Format: | <<SRLns>> |
| UDP Feedback Format: | <<SRLns,*sourceIP*>> |

n = Sensor Input number (n= 1 to 8)

v = Sensor level (v=# from 0 to 999)

sourceIP = The IP address of Neutron (ex: 192.168.1.80)

Example:

The voltage level on sensor input #2 is 4.5 VDC. Send the following to read the sensor level:

<<CRL2>>

The Neutron responds with the following:

<<SRL2167>> UDP = <<SRL2167,192.168.1.80>>

The "167" indicates the sensor level on the sensor input is about 167 times 27 mV or approximately 4.5 VDC.

### 16. <<COR>>

Independently open and close the output relay contacts.

| | | |
|---|---|---|
| TCP Command Format: | <<CORns>> | (default port 23) |
| UDP Command Format: | <<CORns>> | (port 30303) |

n = Relay number (n= 1 or 2)

s = State of relay (s= 0 or 1)

0 – Contact open (default)

1 – Contact closed

Example:

Close relay #1 and open relay #2, wait 2 s, and then open relay #1 and close relay #2 using the following commands:

| | | |
|---|---|---|
| <<COR11>> | Close relay #1 | |
| * | Neutron confirms | (TCP only) |
| <<COR20>> | Open relay #2 | |
| * | Neutron confirms | (TCP only) |
| *... wait 2 s ...* | | |
| <<COR10>> | Open relay #1 | |
| * | Neutron confirms | (TCP only) |
| <<COR21>> | Close relay #2 | |
| * | Neutron confirms | (TCP only) |

17. **<<CTR>>**

Toggle a relay from its current state to closed, then pause up to 1 s and then open the relay. The pause time is in multiples of 100 ms. If the relay is closed when this command is issued, it will simply remain closed and then open after the defined pause time.

| | | |
|---|---|---|
| TCP Command Format: | <<CTRnt>> | (default port 23) |
| UDP Command Format: | <<CTRnt>> | (port 30303) |

n = Relay number (n= 1 or 2)

t = Pause time (t= 1 to 9)

Example:

Relay #1 is currently open. Toggle Relay #1 for 0.5 s by sending the following command:

<<CTR15>>

Relay #1 closes, stays closed for 0.5 s, and then opens. Once the entire close-open cycle is complete, the Neutron sends confirmation.

* (TCP only)

18. **<<CRR>>**

Independently read the current state of a relay.

| | | |
|---|---|---|
| TCP Command Format: | <<CRRn>> | (default port 23) |
| UDP Command Format: | <<CRRn>> | (port 30303) |

n = Relay number (n= 1 to 8)

| | |
|---|---|
| TCP Feedback Format: | <<SRRns>> |
| UDP Feedback Format: | <<SRRns,*sourceIP*>> |

n = Relay number (n= 1 to 8)

s = Relay status (n= 0 or 1)
0 - Relay contact open
1 - Relay contact closed

sourceIP = The IP address of Neutron (ex: 192.168.1.80)

Example:

Check to see if relay #1 is closed by sending the following:

<<CRR1>>

The Mini-Controller responds with a "1" if the relay is closed and "0" if the relay is open. If relay #1 is closed, the feedback is as follows:

<<SRR11>> UDP = <<SRR11,192.168.1.80>>

The first "1" is the relay number and the second "1" indicates the relay is closed.

19. **<<CSN>>**

Set an alias name for the controller. The alias allows a name to be associated with the controller instead of only the static IP address.

| | | |
|---|---|---|
| TCP Command Format: | <<CSNddd>> | (default port 23) |
| UDP Command Format: | <<CSNddd,*targetIP*>> | (port 30303) |

ddd = Controller's Alias (up to 10 ASCII characters)

targetIP = The current IP address of Neutron (ex: 192.168.1.80)

Example:

Set the alias of the controller to "Conf Rm A" as shown below.

TCP: <<CSNConf Rm A>>

UDP: <<CSNConf Rm A,192.168.1.80>>

Neutron sends a confirmation:

* (TCP only)

20. **<<CSP>>**

Set the static IP address of the controller for identification on the LAN. The IP must be unique and reserved for the controller by your IT administrator. The Neutron does not send a confirmation after changing the IP, but it does break the TCP/IP connection. A new connection must be established after changing.

| | | |
|---|---|---|
| TCP Command Format: | <<CSPddd >> | (default port 23) |
| UDP Command Format: | <<CSPddd,*targetIP*>> | (port 30303) |

ddd = IP address (4 octets, 001 to 255)

targetIP = The current IP address of Neutron (ex: 192.168.1.80)

Example:

Set the controller IP address to 10.200.2.100 by sending the following:

TCP: <<CSP10.200.2.100>>

UDP: <<CSP10.200.2.100,192.168.1.80>>

*Important: Write this new IP address on the label provided on the bottom of the unit. It will be needed to identify the unit.*

21. **<<CSB>>**

Set the subnet of the controller. The subnet should be provided by your IT administrator. The Neutron does not send a confirmation after changing the subnet mask, but it does break the TCP/IP connection. A new connection must be established after changing.

| | | |
|---|---|---|
| TCP Command Format: | <<CSBddd >> | (default port 23) |
| UDP Command Format: | <<CSBddd,*targetIP*>> | (port 30303) |

ddd = Subnet address (4 octets, 001 to 255)

targetIP = The current IP address of Neutron (ex: 192.168.1.80)

Example:

Set the controller subnet to 255.255.255.0 by sending the following:

TCP: <<CSB255.255.255.0>>

UDP: <<CSB255.255.255.0,192.168.1.80>>

22. **<<CSG>>**

Set the gateway of the controller. The gateway should be provided by your IT administrator. The Neutron does not send a confirmation after changing the gateway, but it does break the TCP/IP connection. A new connection must be established after changing.

| | | |
|---|---|---|
| TCP Command Format: | <<CSGddd >> | (default port 23) |
| UDP Command Format: | <<CSGddd,*targetIP*>> | (port 30303) |

ddd = Gateway address (4 octets, 001 to 255)

targetIP = The current IP address of Neutron (ex: 192.168.1.80)

Example:

Set the controller gateway to 192.168.1.0 by sending the following:

TCP: <<CSG192.168.1.0>>

UDP: <<CSG192.168.1.0,192.168.1.80>>

23. **<<CSO>>**

Set the TCP port of the controller. The default port is 23 but should not be changed unless necessary. The Neutron does not send a confirmation after changing the port, but it does break the TCP/IP connection. A new connection must be established after changing.

| | | |
|---|---|---|
| TCP Command Format: | <<CSOn>> | (default port 23) |
| UDP Command Format: | <<CSOn,*targetIP*>> | (port 30303) |

n = Gateway address (n= 1 to 65535)

targetIP = The current IP address of Neutron (ex: 192.168.1.80)

Example:

Set the controller IP port to 23 by sending the following:

TCP: <<CSO23>>

UDP: <<CSO23,192.168.1.80>>

*Important: Write this port on the label provided on the bottom of the unit. It will be needed to identify the unit.*

24. **<<CRC>>**

Read the complete TCP/IP configuration of the controller.

| | | |
|---|---|---|
| TCP Command Format: | <<CRC>> | (default port 23) |
| UDP Command Format: | <<CRC >> | (port 30303) |

TCP Feedback Format: <<SRC,a,i,p,s,g,m>>
UDP Feedback Format: <<SRC,a,i,p,s,g,m,*sourceIP*>>

a = Alias name (10 characters)

i = Static IP (4 octets)

p = Port (decimal value from 1 to 65535)

s = Subnet (4 octets)

g = Gateway (4 octets)

m = Mac Address (12 hex characters)

sourceIP = The IP address of Neutron (ex: 192.168.1.80)

Example:

A Mini-Controller has the following settings:

| Property | Value |
|---|---|
| Alias: .............................. | Conf Rm A |
| Static IP:......................... | 192.168.1.80 |
| Port:................................ | 23 |
| Subnet: ........................... | 255.255.255.0 |
| Gateway: ......................... | 192.168.1.1 |
| Mac Address (fixed):....... | 00-04-A3-10-27-23 |

Read the settings for the controller by sending the following:

<<CRC>>

The Neutron returns the following feedback:

TCP:

<<SRC,Conf Rm A,192.168.1.80,23,255.255.255.0,192.168.1.1, 00-04-A3-10-27-23>>

UDP:

<<SRC,Conf Rm A,192.168.1.80,23,255.255.255.0,192.168.1.1, 00-04-A3-10-27-23,**192.168.1.80**>>

25. **<<CRF>>**

Read the complete controller configuration that includes RS-232 port settings, relay states, and sensor input settings.

| | | |
|---|---|---|
| TCP Command Format: | <<CRF>> | (default port 23) |
| UDP Command Format: | <<CRF >> | (port 30303) |

TCP Feedback Format: <<SRF,bpms,$b^2p^2m^2s^2$,c,e,c$^{2,}$e$^2$,g,g$^2$>>
TCP Feedback Format: <<SRF,bpms,$b^2p^2m^2s^2$,c,e,c$^{2,}$e$^2$,g,g$^2$,*sourceIP*>>

| RS-232 Port #1 | RS-232 Port #2 |
|---|---|
| b ..... baud rate | b$^2$.....baud rate |
| p ..... parity | p$^2$.....parity |
| m .... data bits | m$^2$....data bits |
| s ...... stop bits | s$^2$......stop bits |
| Sensor Input #1 | Sensor Input #2 |
| c...... Trigger level (2 digits) | c$^2$ .....Trigger level (2 digits) |
| e...... De-bounce time (2 digits) | e$^2$ .....De-bounce time (2 digits) |
| Relay #1 | Relay #2 |
| g...... Open/closed | g$^2$ .....Open/closed |

sourceIP = The IP address of Neutron (ex: 192.168.1.80)

Example:

A Mini-Controller has the following settings:

| | |
|---|---|
| RS-232 port #1: | 9600 baud, 8 data bits, 1 stop bit, no parity |
| RS-232 port #2: | 57600 baud, 8 data bits, 1 stop bit, even parity |
| Sensor #1: | Trigger Level 50, De-bounce time 100 ms |
| Sensor #2: | Trigger Level 25, De-bounce time 200 ms |
| Relay #1: | Open |
| Relay #2: | Closed |

Read the settings for the controller by sending the following:

<<CRF>>

The Neutron returns the following feedback:

TCP: <<SRF,4001,7201,50,04,25,08,0,1>>

UDP: <<SRF,4001,7201,50,04,25,08,0,1,**192.168.1.80**>>

### 7.16 Summary of Commands

| | | |
|---|---|---|
| 1) | <<CSU>> | Configure RS-232 ports |
| 2) | <<COU>> | Send data to RS-232 port |
| 3) | <<SRU>> | Read data received through the RS-232 port |
| 4) | <<CSI>> | Save IR data to memory |
| 5) | <<CRI>> | Read saved IR data |
| 6) | <<COI>> | Output saved IR data |
| 7) | <<CSM>> | Load IR data directly into RAM |
| 8) | <<CRM>> | Extract IR date from memory into RAM |
| 9) | <<COP>> | Output IR data from RAM |
| 10) | <<CSF>> | Set modulation frequency for IR outputs |
| 11) | <<CST>> | Set sensor trigger levels |
| 12) | <<CSD>> | Set sensor input de-bouncing time |
| 13) | <<CSL>> | Set sensor LED polarity |
| 14) | <<CRS>> | Read sensor status |
| 15) | <<CRL>> | Read sensor level |
| 16) | <<COR>> | Open/close relays |
| 17) | <<CTR>> | Toggle relay |
| 18) | <<CRR>> | Read state of the relays |
| 19) | <<CSN>> | Set controller alias name |
| 20) | <<CSP>> | Set static IP address of controller |
| 21) | <<CSB>> | Set subnet of controller |
| 22) | <<CSG>> | Set gateway of the controller |
| 23) | <<CSO>> | Set IP port of the controller |
| 24) | <<CRC>> | Read complete TCP/IP configuration |
| 25) | <<CRF>> | Read complete configuration |

## 8. Troubleshooting Guide

We have carefully tested and have found no problems in the supplied Neutron Mini-Controller. However, we would like to offer suggestions for the following:

| Symptom | Resolution |
|---|---|
| LEDs Are OFF | 1. Make sure the unit is plugged into a working AC outlet and the DC plug is inserted all the way into the controller. The power terminal block is on the left side (view from back) of the controller.<br>2. Use only the power adapter provided. |
| Cannot Make TCP Connection | 1. Check the LEDs on the Ethernet connector. A green LED indicates the Neutron is recognizing the network and a flashing amber LED indicates communication traffic over the network.<br>2. Make sure the control application has the correct TCP/IP address for the Neutron Mini-Controller.<br>3. Review the TCP control section of this manual. Check the Ethernet properties for the Neutron. Each Neutron is shipped with the following static IP settings: IP=192.168.1.80, Subnet Mask=255.255.255.0, Gateway=192.168.1.1, and Port=23 but must be configured for use on other networks; reference commands <<CSN>>, <<CSP>>, <<CSB>>, <<CSG>>, and <<CRC>>.<br>4. Check with your IT administrator to verify the Neutron was assigned a valid static IP address that is accessible over the LAN and does not conflict with other devices on the network. Each Neutron controller requires its own unique static IP address. The DOS "ping" command can be used to determine if the internal server is up and to verify the assigned IP address is "reachable" on the IP network. |
| Don't Know<br>or Forgot<br>IP Address | The Neutron Mini-Controller ships with IP address 192.168.1.80, port 23. If the IP is changed and then lost or forgotten, there are 2 ways to "discover" the IP setting. One is through the TX1 output of the RS-232 terminal block and the other is using a UDP broadcast.<br>RS-232<br>1. Connect the Neutron's RS-232 port 1 to a PC or MultiTouch panel. Wire the Neutron's TX1 and GND pins to the receive and ground pins of the PC or MultiTouch DB9 connector (typically pins 2 and 5 respectively). The RX1 line is not needed but can also be connected (typically pin 3 of the DB9 connector).<br>2. Launch AVSnap on the PC or MultiTouch, enter communication mode, and connect to the COM port.<br>3. While monitoring the Neutron's TX1 output, reset power to the Neutron. After the unit initializes, the static IP, port, subnet mask, and gateway settings are sent to the TX1 port in the following format:<br>IP: 192.168.1.80<br>Subnet Mask: 255.255.255.0<br>Gateway: 192.168.1.1<br>Port: 23<br>UDP Broadcast<br>1. Send a UDP broadcast to 255.255.255.255 port 30303 with the following data string:<br>?Altinex<br>2. The Neutron responds with the following data in a UDP packet to the host IP:<br><CP500-110,00:04:A3:10:25:84,00023,192.168.1.80,CP500-110 ,255.255.255.0,192.168.1.1><br>The data string contains the model number, the Neutron MAC address, the TCP port number, IP address, alias, subnet mask, and gateway to use with a standard TCP connection. |
| Wrong Control/RS-232 Data | 1. See the TCP control section of this manual. Check the properties for both RS-232 ports (baud rate, parity, etc.); reference commands <<COU>>, <<CSU>>, and <<SRU>>.<br>2. Verify the port to which the device is connected matches the commands used to send data from the control application. |
| No Response Using IR Control | 1. Verify the port number to which the IR cable is connected and make sure the emitter end of the cable is mounted directly over the receiver eye of the controlled device.<br>2. See the TCP control section of this manual and verify the IR data string is saved to the correct memory location and that the control application is sending the correct commands to the controller;<br>reference commands: <<CSI>>, <<CRI>>, <<COI>>, <<CSM>>, <<CRM>>, and <<COP>>.<br>3. Verify the IR Output is set for the correct modulation frequency; reference command <<CSF>>. |
| Relays Do Not Close | 1. See the TCP Control section of this manual. Verify the control application is sending the correct commands to open or close the relays; reference commands <<COR>> and <<CRR>>.<br>2. Check the relay status on the front of the controller. If the LED is ON, the relay is CLOSED. |
| Sensor Inputs Do Not Trigger | 1. Make sure the sensors are connected to the proper input of the controller.<br>2. If the sensor requires external power, make sure the sensor is connected to the proper power source. The Neutron has 2 voltages available for powering sensors: +12VDC and +24VDC.<br>3. See the TCP Control section of this manual. Verify the Neutron sensor input properties have been set correctly for trigger levels and de-bounce times; reference commands <<CST>>, <<CSD>>, <<CSL>>, <<CRS>>, and <<CRL>>.<br>4. Verify the control application is checking the sensor status, and is programmed to respond to changes in the sensor input status. The trigger levels in the control application should match those programmed into the controller. |