## iPAL Overview

The iPAL is a hardware and software system for remote control via Ethernet. The basic iPAL hardware has eight controllable outputs, and eight status inputs. This basic platform is combined with specific A/C power or communications switches, or is available as a standalone control/monitor device.

iPAL is controlled and monitored using standard TCP/IP Protocols. Most iPAL applications use Web or Telnet to control and monitor iPAL. This document is designed for those developers who wish to write applications to talk directly to iPAL via TCP utilizing port 9100. It provides the protocol required for connection, operation and monitoring of the iPAL hardware.
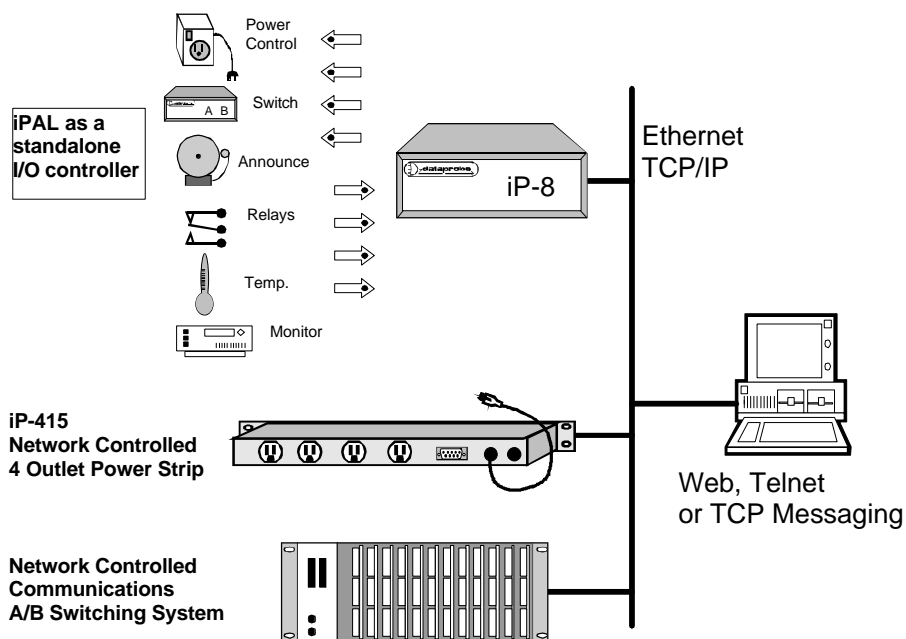
iPAL Protocol

The iPAL uses the TCP (Transport Communication Protocol) to communicate with the client PC. The actual iPAL protocol is wrapped in the TCP protocol. To use the iPAL, establish a TCP connection on **port 9100** (Stream Socket) to the iPAL. Once connected use the Send() function to send the commands to the iPAL and the Recv() function to receive the iPAL's response. The following outlines the commands and their responses. A Windows DLL is available free of charge from Dataprobe. This DLL handles all the winsock calls and has calls to handle all of the iPAL commands.

Command Format

All of iPAL's commands follow the following format. Some of the commands do not carry data. When using these commands, the data field must be a NULL.

<div align="center">

<esc><password><esc><data><command>

</div>

<esc>   ASCII Escape character.

<password> This field is a 1 to 8 character field that is user programmable. The default password is PASS. The password is case sensitive.

<data>   This field can be 0, 1 or 2 bytes in length depending on the command being sent. See the command list for data lengths.

<command>   This field is a single character in length.   See the command list for a list of valid commands.

Response Format

All of iPAL's commands return a response via the Ethernet. All responses are 8 bytes in length but the data that the response carries differs for different types of commands. These different responses are outlined below.

Control Command Response

Control commands are the commands that effect the state of the output relays. All of these command and the Output command return the current status of the outputs with feedback. The table below outlines all of the response codes. The response codes are returned as and 8 byte string. Byte 0 represents the current status of channel 1 and byte 7 represents the current status of channel 8.

Status Code	Description
0	Channel is off
1	Channel is on
4	Channel is off according to feedback but control is requesting that it be on
5	Channel is on according to feedback but control is requesting that it be off
6	Cable Error (Modes 2 or 5 only using Tri-State Feedback)

Input Command Response

There is only one input command. Just as the Control Command it too returns an 8 byte response where the first byte represents channel 1 and the last represents channel 8. There are only 2 possible response codes for the Input command, ASCII 0 and ASCII 1. The '0' indicates that the input is low and the '1' indicates that the channel is high.

Setup Commands

The setup commands also return an 8 byte response via the Ether Net. These commands return their result as either OK or ERROR. These results are padded with trailing spaces to keep the protocol consistent.

Command List

Energize: E

Description: This command is used to energize one or all of the iPAL's relays. The iPAL will return a result string that contains the current status of all 8 channels.

Example: **<ESC>PASS<ESC>1E**
Energizes channel 1's relay.

Note: To energize all channels substitute the channel number with the letter 'A'.

De-energize: D

Description: This command is used to de-energize one or all of the iPAL's relays. The -iPAL will return a result string that contains the current status of all 8 channels.

Example: **<ESC>PASS<ESC>1D**
De-energizes channel 1's relay.

Note: To de-energize all channels substitute the channel number with the letter 'A'.

Pulse: P

Description: This command is used to pulse one or all of the iPAL's relays. The iPAL will return a status string that contains the current status of all 8 channels.

Example: **<ESC>PASS<ESC>1P**
Pulses channel 1 for the duration set be the time command.

Note: To pulse all channels substitute the channel number with the letter 'A'.

Time: T

Description: This command is used to set the duration of the pulse in seconds. It returns the current status of the relay outputs.

Example: **<ESC>PASS<ESC>10T**
Sets the pulse duration for 10 seconds.

Note: This command will extend any pulses that are in progress. It is recommended that this command is not used when there are pulses in progress.

Input: I

Description: This command is used to get the input status of the iPAL. This command returns a status string that contains the current value of the input circuits.

Example: **<ESC>PASS<ESC>I**

Output: O

Description: This command is used to get the output status of the iPAL. This command returns a status string that represents the current status of the relay drivers.

Example: **<ESC>PASS<ESC>O**

Security: S

Description: This command is used to change the security code (password) of the iPAL.

Example: **<ESC>PASS<ESC>NEWS**
This will change the password form the default PASS to NEW. All subsequent command will use the new password.

Set Mode: M

Description: This command is used to set the operations mode of the iPAL. The iPAL has 6 different operation modes. The modes affect the function of the LED indicators (when provisioned) on the iPAL hardware, and the expected feedback on the input circuits.

Mode	Application	LED Operation	Feedback
0	A/B or Generic I/O	LED On = Relay Energized	None
1			Bi-State Low=Relay Energized
2			Tri-State*
3	Power Control with Default Power = ON	LED On = Relay De-Energized = Power to Receptacle On	None
4			Bi-State High/Open = Relay Energized
5			Tri-State*

* Tri-State feedback is found in Dataprobe products and is used to insure cable integrity. Contact Dataprobe for more information.

Example: **<ESC>PASS<ESC>1M**
This will set the unit to Mode 1