Data Logging with the PLCA-22 Power Line Communications Analyzer

Introduction

The PLCA-22 Power Line Communications Analyzer provides a simple, cost-effective method of field testing the operation of Echelon's multi-frequency, CENELEC EN50065-1 and EIA709.2 compliant, power mains communication technology.

Now PLCA-22 analyzers include an output-only RS-232 communication port. This RS-232 port gives users the ability to log network performance over time. There are many uses for this information, some of which are:

- Tracking time dependent network problems
- Monitoring long term network reliability
- Determining packet throughput

The data from the PLCA-22 serial port may be viewed on a VT100 terminal emulator program set for 9600 baud, 8 data bits, no parity, and 1 stop bit. Using VT100 escape codes, the PLCA-22 creates a duplicate copy of its internal LCD screen on the external VT100 terminal emulator.

Purpose

This document explains the VT100 escape codes used by the PLCA-22 for placing data on the screen. Also, the location and format of the data written to the screen is discussed in detail to aid in creating custom data logging applications for the PLCA-22.

An unsupported example data logging application including source code is available at Echelon's website at www.echelon.com under the developer's toolbox.

Related Documentation

The following Echelon documents are suggested reading:

LONWORKS PLCA-22 Power Line Communications Analyzer User's Guide (078-0176-01)

LONWORKS PLT-22 Power Line Transceiver Module User's Guide (078-0175-01)

RS-232 Serial Port Operation

The RS-232 serial port of the PLCA-22 operates at 9600 baud with 8 data bits, no parity, and 1 stop bit. During operation of the PLCA-22, a mirror image of the built-in LCD screen is displayed using a small subset of VT100 escape codes.

Connecting to the PC

Connecting the PLCA-22 serial port to the PC is done using a regular modem cable with a female connector for the PC COM port and a male connector for the PLCA-22 serial port. Do not use a null modem cable to connect the PC to the PLCA-22. Null modem cables switch pins 2 and 3 (the transmit and receive data) and will not work.

The serial port of the PLCA-22 is wired for data communication equipment (DCE). This means that it receives transmitted data (TXD) from the data terminal equipment (DTE) which is the PC. Received data (RXD) is an output from the PLCA-22 to the RXD input of the PC. The serial port pinout is detailed below.

DB9 Serial Port Pinout

DB9 Pin	Signal Name	In/Out	Description
1	CD	Out	Carrier Detect
2	RXD	Out	Received Data
3	TXD	In	Transmitted Data
4	DTR	In	Data Terminal Ready
5	GND	-	Signal Ground
6	DSR	Out	Data Set Ready
7	RTS	In	Request To Send
8	CTS	Out	Clear To Send
9	RI	Out	Ring Indicator

The PLCA-22 only implements RXD, TXD, GND, DSR, and DTR. Other signal names are provided as a general reference only.

Serial Communication Configuration

To communicate with the PLCA-22, the serial communication configuration on the PC must be as follows:

- 9600 Baud
- 8 data bits
- No parity
- One stop bit
- No flow control

VT100 Escape Codes

The PLCA-22 uses a small subset of VT100 escape codes for its operation. The escape codes used are for positioning the cursor and for clearing the screen.

Clear Screen

Clearing the screen is accomplished using the following escape sequence:

<esc>[H<esc>[J

In hexadecimal, this sequence is:

0x1b 0x5b 0x48 0x1b 0x3b 0x4a

Clear Screen happens only a few times. Once at power up and whenever the user changes the analyzer to phase mode.

Set Cursor Position

Setting the cursor position is accomplished using the following escape sequence:
<esc>[Row;ColH

Row and Col above are the ASCII representation of the row and column numbers. For example, if the desired location for the cursor on the screen is row 2, column 17, then the escape sequence would look like this:

<esc>[2;17H

In hexadecimal:

0x1b 0x5b 0x32 0x3b 0x31 0x37 0x48

Note that the numbers 0-9 correspond to character codes 0x30 through 0x39 in ASCII.

Serial Port Data Format

The serial port data format is straightforward for all fields with the exception of the Rcvd and Lost fields on the Recv screen. Here's an example of how the Recv mode screen would appear in Ackd4try mode:

R	e	c	v		A	c	k	d	4	t	r	y		C	-	b	a	n	d
P	k	t	s	:						T	x	V	p	p	:	7	V		
R	c	v	d	:	1	0	0	0		A	t	t	n	:	0	d	B		
L	o	s	t	:	0					E	r	r	:	0	.	0	0	%	

The screen is 4 rows by 20 columns. Each location on the screen may be addressed by a row and column position. For example, to replace the message Recv in the upper left hand corner with Send, the PLCA-22 would issue the set cursor position escape code with a row value of 1 and a column value of one, followed by each character of the word Send.

All of the text fields of the PLCA-22 are updated using a set cursor position escape sequence followed by a string of characters for that field.

The special locations are the numeric portions of the Rcvd and Lost data fields. Only the numbers that have changed are updated. For example, if the 1000 above changes to 1001, a set cursor position escape sequence positions the cursor at row 3 column 9 and then sends a single 1.

The locations specified in the following mode descriptions are in the format row,col.

Function Mode Field (Location 1,1)

The function mode field can be Setup, Send, Recv, or Phase. While the PLCA-22 is idle with the cursor on this field, it will occasionally send out a set cursor location escape code.

When the user changes this field, it will flash off and on. The PLCA-22 performs the flashing by positioning the cursor at location 1,1 and sending 4 space (' ') characters, pausing, then locating the cursor at 1,1 and sending the mode string again. This process of printing and clearing the characters continues until the user either presses enter to select the new function mode or move to abort the change.

Communication Service Field (Location 1,6)

The communication service field can be Ackd4try, UnackPri, or UnackSec. The value of this field determines what type of testing is being done on this communications analyzer.

When the user changes this field, it will flash off and on. The flash operation is identical to the function mode field.

Frequency Band Field (Location 1,15)

The values for the frequency band field are C-band or A-band.

Packets Field (Location 2,6)

The packets field contains the number of packets to be sent on the sender, and nothing on the receiver. The field is a number terminated by a space or a 'k' and a space. If the last character after the numbers is a 'k', the value of the field should be multiplied by 1000.

Transmit Voltage Field (Location 2,18)

The possible values for the transmit voltage field are 1.7V, 3.5V, 7V, or 10V.

Packet Count Field (Location 3,6)

The packet count field contains numbers from 0 to 9999K. At 99,999 packets, the analyzer begins displaying packet count with a k following the number. For example, 100,000 packets will be displayed as 100K.

Only the number that changes is updated on the packet count field. For example, if the current number is 123 and a new packet is received, the field must be updated to show 124. To do this, the analyzer determines which character it will have to update and changes only that character on the display. The escape sequence for positioning the cursor at location 3,8 is sent, followed by the character 4.

One possible method for handling this number update is to store the updates into a string of characters based on their screen position. Screen positions 3,6 through 3,10 could be stored into a text string and converted to an integer after the number update is complete.

Note that this field always starts at location 3,6 and grows towards location 3,10. Location 3,11 will always be a space.

Transmit Attenuation Field (Location 3,17)

The transmit attenuation field will contain values of 0dB, 6dB, 12dB, 18dB, and 24dB. During receiver or transmit, the cursor is usually located on this field. This fact can be useful for determining when it is safe to read the packet count field and other fields. If the cursor is on the transmit attenuation field, then other fields are not being updated and contain valid information.

Lost Packets Field (Location 4,6)

The lost packets field reports how many of the received packets have been lost. This is a space-terminated field that contains a number from 0 to 99,999. When the number exceeds 99,999 it is divided by 1000 and a 'k' is appended to the number. For example, 100,000 is 100k.

The PLCA-22 updates this field exactly as the Packet Count field is updated. Only the number that changes is updated on the display. Take a change from 16 to 17 lost packets. The cursor position would be set to the location of the 6 and a 7 would be sent to overwrite the 6.

Note that this field always starts at location 4,6 and grows towards location 4,10. Location 4,11 will always be a space.

Error Rate Field (Location 4,16)

The error rate field contains a fixed-point percentage that represents the packet error rate. This field is terminated at location 4,20 with a percent (%) sign. The error rate field is updated sequentially starting with location 4,16 and ending with location 4,20.

Fourth Line Status Messages (Locations 4,1 through 4,20)

Occasionally the PLCA-22 communicates with the remote PLCA-22 to configure, start, or stop a test. During this communication, the fourth line of the LCD screen is used to display status messages. These messages blink on and off and can be any of COMMUNICATING, REMOTE READY, NO REMOTE UNIT FOUND, and PENDING. These messages can be used as an indication that a test has been started or stopped, or as an indication of problems with the test setup.

Example Data Logging Application

The example data logging application can be located on Echelon's website at
www.echelon.com under the developer's toolbox. The LOGPLCA.EXE is a
win32 program that will run under Windows 95 and Windows NT. This
program has undergone limited testing and is provided with source code as an
example of how a data logging application might be written.

LOGPLCA.EXE is built from 3 C source files and 2 header files. Here are the
names of the files along with a brief description of each:

Filename	Description
logplca.c	Main routine for data logging
logplca.h	Header file containing default com port and log interval settings
comm.c	Windows serial port communications routines
comm.h	Windows header files needed for communications
screen.c	VT100 escape code parsing and screen management

The program is compiled with a freely available c-compiler named GCC that is
include with the EMX and RSXNT development distributions. The version used
(obtained using gcc –v):

    gcc driver version 2.7.2.1 (rsxnt) executing gcc version 2.7.2.1

LOGPLCA Usage

LOGPLCA is a console application, so it must be used from a MS-DOS command
prompt window under Windows 95 or NT.

LOGPLCA accepts two arguments. The first argument is the communications
port to use when talking to the PLCA-22 analyzer. The default port is COM1: if
no port is specified.

The second argument is the name of the log file. If COM1: is used, the only
argument that needs to be specified is the name of the log file.

When using COM1:, your dos command line would look something like this:

   Logplca mylog.log

When using another port, such as COM2:, the command line would look like:

   Logplca com2: mylog.log

After the application has been started, a PLCA-22 analyzer should be connected
to the selected serial port and turned on. LOGPLCA will then being receiving
characters from the PLCA-22 and synchronize the internal virtual screen with the
LCD screen of the PLCA-22.

Make sure there is a second PLCA-22 plugged into your power line network.
The second PLCA-22 will be referred to as the remote unit. The unit connected
to the PC will be referred to as the local unit.

Set the mode of the local unit to Recv. Set the packet count on the remote unit to
a number greater than or equal to 1000 packets. Log information will be
recorded at 1000 packet intervals.

**NOTE: IF THE PLCA IS NOT SET FOR 1000 OR MORE PACKETS,
LOGPLCA WILL RECORD NOTHING TO THE LOGFILE.**

Press start on the remote or local unit and wait. After at least 1000 packets have
been received by the local unit, stop LOGPLCA and examine the log file. There
should be a tab-delimited line of data in the following format:

   Date   Time   PacketCount   PacketsLost

The date field is in the format MM/DD/YY. MM is month, DD is date, and YY is
year.

The time field is in the format hh:mm:ss, where hh is hours in 24 hour format,
mm is minutes, and ss is seconds.
Finally, PacketCount is the value recorded from the packet count field, and
PacketsLost is the number recorded from the lost packets field.

How LOGPLCA Works

LOGPLCA captures the characters and VT100 escape codes from the RS-232 port and places them on a virtual screen in memory. The virtual screen is a duplicate of the PLCA-22 LCD screen. This virtual screen is a linear array 80 characters long. When a VT100 cursor position code is received, the row and column are processed using the following equation to yield the offset into the virtual screen array:

$$\text{Screen position} = (\text{row} - 1) * 20 + \text{col} - 1$$

When the upper left hand corner of the screen starts with an 'R', the lower left hand corner starts with an 'L', and the cursor moves from the packet count field to the attenuation field, the virtual screen is examined for the current packet count. When the packet count is a multiple of 1000 packets, the packet count and number of packets lost are recorded to the log file.

The only fields used by LOGPLCA are the function mode field, packet count field, transmit attenuation field, and the lost packets field. The transmit attenuation field is used to determine when the other fields are safe to read. When the cursor is not on the transmit attenuation field, the cursor could be updating another area of the screen. This precaution ensures that valid packet and lost packet information is recorded to the log file. If this check were not performed, partially updated information could be stored to the log file giving incorrect results for the test.

Other precautions are taken in setting the screen position for the cursor. Sometimes the VT100 escape sequences can be corrupted if the RS-232 data link between the PLCA-22 and the computer is noisy. When this happens the result of a set cursor position escape sequence could place the cursor position outside the bounds of the virtual screen buffer causing a segmentation violation in C. An upper limit of 79 and a lower limit of 0 are placed upon the VT100 processing routines to limit the range of the cursor positions.