APPLICATION	
NEXT ASSY	USED ON
	iPM

REVISIONS			
REV	DESCRIPTION	DATE	APPROVED
A	Released in support of ECO E61836	5/12/11	*PLM*

1. SCOPE
   This document describes format and command syntax for RS485 serial communication interface of model iPM. In addition, a description of how to use soft panel utility program to communicate with the iPM is provided. Appendix A and Appendix B contain detailed information needed to write programs using driver dll's.

2. GENERAL
   a. Description. Model iPM provides an EIA RS485 compatible serial data interface allowing enhanced control features, access to diagnostic data and real-time readout of measured parameters of the AC power input such as frequency, voltage, phase and others. In addition, changes can be made to configuration of features and limits using the serial interface.
   b. Interface settings. Hardware settings of the serial interface are as follows.

Table 1. Interface Settings

Baud Rate	2400 to 115,200 (default)*
Data Bits	8
Stop Bits	1
Parity	None
Protocol	None
Isolation	Isolated*

*See available features, below

   c. Available features. The following interface features are available, based on hardware and firmware revision.

Table 2. Available Features

Feature		Hardware Rev.	Firmware Rev.
Fixed baud rate, 115,200 baud		----	up to 3
Selectable baud rate, 2400 to 115,200 baud		----	4 & up
Multi-drop addressability		----	4 & up
Configuration limited to model, S/N & Delta Cal		----	up to 3
Full configuration with enable		----	4 & up
Data logging		----	A & up
Isolation	2-pole units	5	----
	3-pole units	4	----

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCE ON FRACTIONS: ± 1/64 ANGLES: ± 1/2° DECIMALS .XX ± 0.02 DECIMALS .XXX ± 0.005	DRAWN PVR	DATE 12/16/10
	CHECKED *PLM*	
	ENG APPVL *PLM*	
	MFG APPVL *PLM*	
MATERIAL:	QA APPVL *PLM*	
FINISH:		

North Atlantic Industries

110 Wilbur Place
Bohemia, New York 11716
Tel: 631-567-1100
Fax: 631-567-1823
Web: www.naii.com

INTELLIGENT POWER MONITOR
PROGRAMMING REFERENCE

SIZE	CODE IDENT	
A	ØVGU1	iPMA052

SCALE	REV A	SHEET 1 OF 22

3. COMMAND FORMAT
All commands are sent via the RS485 serial interface as ASCII text. Each command is terminated with an ASCII Linefeed character (0x0A) indicated as <lf> in paragraphs and tables below. Response messages for all commands are as indicated below. Character string data responses will be terminated with an ASCII Linefeed character. Binary data response messages will not have a terminating character, but will terminate when the specified number of bytes has been sent.

4. MULTI-DROP ADDRESSABILITY
Multi-drop addressability feature (firmware version 4 and up) permits multiple model iPM units to be connected to the same RS485 bus and addressed individually. Each unit must be configured to addressable mode and programmed with a unique address (0 to 255). **Use of address feature is optional.** This configuration can be done by the end user through the RS485 serial interface. The steps required to use this feature are as follows.

   a. Prior to installation of unit:
      1) Enable configuration mode using the CONFIG ON command.
      2) Set desired device address (0 to 255) using the SETADDRESS <address> command
      3) Enable multi-drop addressable mode using the SETMULTIDROP ON command
   b. Operation of unit with addressable mode enabled:
      1) When unit is first powered up, it will not respond to any address-restricted commands or queries.
      2) Control computer must command desired unit on bus to be a listener by using the ADR <address> command, where <address> is the address of the desired listener.
      3) When unit receives the ADR command with matching address, it will respond to all address-restricted commands and queries until any of the following occurs.
         ADR broadcast command is received with a non-matching address parameter.
         UNADDRESS broadcast command is received.
         Unit is powered off.
      4) Note that only one unit may be addressed at a time.
      5) There should not be more than one unit with a given address on the same RS485 bus.
   c. Operation of unit when addressable mode is disabled (command: SETMULTIDROP OFF).
      1) There can be only 1 iPM on a given RS485 bus when it is non-addressable.
      2) Unit will respond to all address-restricted commands and queries without need for ADR or UNADDRESS commands.

5. SELECTABLE BAUD RATE
RS485 baud rate is set to **115.2K default value** when unit is shipped from factory. This the only available baud rate for older units with firmware revision 3 or below. On newer units, baud rate may be set using the SETBAUD command to one of the following values.

   115.2K , 57.6K, 38.4K, 28.8K, 19.2K, 14.4K, 9600, 4800, 2400

6. SUMMARY OF COMMANDS
See Table 3 below for summary of available commands. Note: <lf> = ASCII linefeed (0x0A) and <sp> = single ASCII space (0x20).

Table 3. Summary Of Commands

Note	Command	Parameter	Response	N o LF	Data type	Comment
	IDENTITY COMMANDS					
A	PARTNO?		similar to "iPM42A0101(047-812)(087-211)(005-999)00"		string	
A	SERNO?		Serial Number, up to 32 characters, alphanumeric		string	
A	VER?		similar to "VER 004 2010-03-16"		string	
A	MANUF?		always = "North Atlantic Industries www.naii.com"		string	
	CONTROL COMMANDS					
A	OFF		"OK"		string	
A	RESET		"OK"		string	
A	TEST		"OK"		string	
4	ADR	<address> (0 to 255)	none			
4	UNADDRESS		none			
	STATUS COMMANDS					
A	STATUS?		12 bytes	X	binary	
A	POWER?		0 or 1		string	
A	OPSTATE?		0 thru 4		string	
	MEASUREMENT COMMANDS					
A	MEASURE?		34 bytes	X	binary	
A	BITRESULT?		24 bytes	X	binary	
	DATA LOGGING COMMANDS					
A, R	RECCOUNT?		number of records stored		string	
A, R	RECSIZE?		size of rec (string)		string	
A, R	GET	<start rec>,<end rec> (0 to 65535)	many bytes (68 x num records)	X	binary	
A, R	RECORD?		68 bytes	X	binary	
A, R	LASTTRIP?		68 bytes	X	binary	
A, R	CLEARCOUNT		"OK"		string	
A, R	HIGH?		0 thru 65535 (string)		string	
A, R	NEXT?		0 thru 65535 (string)		string	
C, R	ERASEFLASH		"OK"		string	

Table 3. Summary Of Commands (continued)

Note	Command	Parameter	Response	No LF	Data type	Comment
	CONFIGURATION COMMANDS					
C	SETPARTNO	<part number> (similar to "iPM42A0101(047-812)(087-211)(005-999)00")	"OK"		string	
C	SETSERNO	<serial number> (up to 32 characters, alphanumeric)	"OK"		string	
C	CALDELTA		"OK"		string	
4	SETCONFIG	"ON" or "OFF"	none			
4	CONFIG?		"ON" or "OFF"		string	
C, 4	SETBAUD	<index> (0 to 15)	"ARE YOU SURE?"		string	follow with "YES" command
C, 4	BAUD?		<index>,<baud rate> (index is 0-15) 0 = 115200 (default) 1 to 3 = not used (set s 115200) 4 = 2400 5 = 4800 6 = 9600 7 = 14400 8 = 19200 9 = 28800 10 = 38400 11 = 57600 12 to 15 = not used (sets 115200)		string	
C, 4	SETADDRESS	<address> (0 to 255)	"ARE YOU SURE?"		string	follow with "YES" command
C, 4	ADDRESS?		0 thru 255		string	
C, 4	YES		"OK"		string	
C, 4	SETMULTIDROP	"ON" or "OFF"	"ARE YOU SURE?"		string	follow with "YES" command
C, 4	MULTIDROP?		"ON" or "OFF"		string	

NOTES:

A = Addressable in revision 4 and up

C = Requires configuration mode in revision 4 and up

4 = Functionality available in revision 4 and up

R = Functionality available in revision A and up

7. IDENTITY COMMANDS

Identity query commands permit identification of model iPM including manufacturer, model code, serial number and firmware revision. All commands in this group are effected by optional multi-drop addressability mode. See paragraph 4.

a. MANUF?

1) Description: Queries the Manufacturer identification.

2) Format of query:

MANUF?<lf>

3) Format of response (38 character ASCII string):

North Atlantic Industries www.naii.com<lf>

b. PARTNO?

1) Description: Queries the configuration part number.

2) Format of query:

PARTNO?<lf>

3) Format of response (39 character ASCII string, similar to following):

iPM42A0101(047-812)(087-268)(005-999)00<lf>

c. SERNO?

1) Description: Queries the serial number of the unit.

2) Format of query:

SERNO?<lf>

3) Format of response (up to 32 character ASCII string):

203456-7<lf>

d. VER?

1) Description: Queries the firmware version of the power monitor.

2) Format of query:

VER?<lf>

3) Format of response (up to 32 character ASCII string, similar to following):

VER 004 2010-03-16<lf>

8. CONTROL COMMANDS

Control commands provide control of various functions including: turn-off or allow control of output, perform internal self-test and establish which unit is listener in a multi-drop addressable configuration. All commands in this group except ADR and UNADDRESS are effected by optional multi-drop addressability mode. See paragraph 4.

a. OFF

1) Description: Commands power monitor to OFF state, output relays not energized, regardless of input power quality.

2) Format of command:

OFF<lf>

3) Format of response (ASCII string):

OK<lf> if normal or
ERR01 - FAILED STATE<lf> if monitor is in failed state

b. RESET

1) Description: Commands power monitor enable output to ON state when AC power is within specified limits.

2) Format of command:

RESET<lf>

3) Format of response (ASCII string):

OK<lf> if normal or
ERR01 - FAILED STATE<lf> if monitor is in failed state

c. TEST

1) Description: Commands execution of built-in self-test. Result of test can be retrieved using BITRESULT? query.

2) Format of command:

TEST<lf>

3) Format of response (ASCII string):

OK<lf>

d. ADR

1) Description: Sends listen address to all iPM units on a multi-drop bus. Unit that is configured to matching address and has multi-drop addressability enabled will begin listening to all subsequent address restricted commands. Units that do not support this feature or are not configured for multi-drop addressability will ignore this command.

2) Format of command:

ADR<sp><address><lf> where <address> is 0 thru 255

3) Format of response:

No response.

e. UNADDRESS

1) Description: Commands all iPM units on a multi-drop bus to stop listening to address restricted commands, if unit is configured for multi-drop addressability. Units that do not support this feature or are not configured for multi-drop addressability will ignore this command.

2) Format of command:

UNADDRESS<lf>

3) Format of response:

No response.

9. STATUS COMMANDS

Status query commands provide a real time status of the operational stat of the power monitor, on/off status of the output and details of parameters that over trip limit or over warning limit. All commands in this group are effected by optional multi-drop addressability mode. See paragraph 4.

a. STATUS?

1) Description: Queries the status data structure which contains information about the condition of the monitored AC power and operation status of the power monitor. Response to command will consist of 2 parts. First part will be an ASCII string indicating number of bytes to follow. Second part will be a binary message with no terminating linefeed character.

2) Format of command:

STATUS?<lf>

3) Format of response:

12<lf> ASCII string indicating number of bytes to follow
<12-byte binary data> Note: no terminating linefeed character

Table 4. Status Message Format

Byte	**Data**
1	OpState (8-bit)
2	PowerOK (8-bit)
3-6	TripFlag (32-bit)
7-10	CautionFlag (32-bit)
11-12	bitStatus (16-bit)

Message Details:
OpState - 8-bit ASCII: 0 = OFF (output off); 1 = reserved; 2 = RESET (allows output on if power is ok); 3 = TRIPPED (output off because power no longer good); 4 = FAILED (output off and unit disabled due to internal self-test fail. Will not turn back on until power is removed and restored).

PowerOK - 8-bit ASCII: 0 = Power No Good or 1 = Power Good

TripFlag & CautionFlag - 32-bit binary register. TripFlag indicates abnormal power conditions exceeding trip limit. CautionFlag indicates marginal power conditions. For each bit of register, “1” indicates value exceeds trip or caution limit, “0” indicates value within limit. Bit definitions follow.

Table 5. Trip & Caution Flag Format

Bit	Parameter	Bit	Parameter
0	Improper Phase Sequence	15	Ph. A High Peak volt.
1	High Steady-State Freq.	16	Ph. B High Peak volt.
2	Low Steady-State Freq.	17	Ph. C High Peak volt.
3	Ph. A High Steady-State Volt.	18	High Transient Freq.
4	Ph. A Low Steady-State Volt.	19	Low Transient Freq.
5	Ph. B High Steady-State Volt.	20	Ph. A High Transient Volt.
6	Ph. B Low Steady-State Volt.	21	Ph. A Low Transient Volt.
7	Ph. C High Steady-State Volt.	22	Ph. B High Transient Volt.
8	Ph. C Low Steady-State Volt.	23	Ph. B Low Transient Volt.
9	Ph. A High DC component	24	Ph. C High Transient Volt.
10	Ph. B High DC component	25	Ph. C Low Transient Volt.
11	Ph. C High DC component	26-31	Reserved
12	Ph. A High THD		
13	Ph. B High THD		
14	Ph. C High THD		

bitStatus - 16-bit binary register. Indicates abnormal internal self-test conditions. For each bit of register, "1" indicates value exceeds limit, "0" indicates value within limit. Bit definitions follow.

Table 6. bitStatus Format

Bit	Parameter	Bit	Parameter
0	Half-Ref voltage Hi or Lo	6	Reserved
1	VREF voltage Hi or Lo	7	Phase A input test fail
2	+5V voltage Hi or Lo	8	Phase B input test fail
3	+5VA voltage Hi or Lo	9	Phase C input test fail
4	Relay Drive voltage Hi or Lo	10	Temperature too high
5	Reserved	11-15	Reserved

b. POWER?

1) Description: Queries PowerOK status indicating whether the condition of monitored AC power is within all configured trip limits..
2) Format of command:

   POWER?<lf>

3) Format of response (ASCII string):

   1<lf> if AC power is within limits
   0<lf> if at least 1 monitored parameter exceeds limit

c. OPSTATE?

1) Description: Queries OpState status indicating the operational state of the power monitor.
2) Format of command:

   OPSTATE?<lf>

3) Format of response (ASCII string):

   <NUMERIC STRING><lf> Valid values = 0 thru 4
   0 = OFF state (output is off)
   1 = reserved
   2 = RESET state (allows output on if power is ok)
   3 = TRIPPED state (output is off because power exceeds limits)
   4. = FAILED state (output off and unit disabled due to internal self-test failure)

10. MEASUREMENT COMMANDS

Measurement query commands allow for real time readout of measured parameters of AC power being monitored as well as real time readout of internal hardware conditions of the monitor. All commands in this group are effected by optional multi-drop addressability mode. See paragraph 4.

a. MEASURE?:

1) Description: Queries data structure which contains real time measurements of all parameters of the AC signal that is being monitored by the iPM. Response to command will consist of 2 parts. First part will be an ASCII string indicating number of bytes to follow. Second part will be a binary message with no terminating linefeed character.

2) Format of command:

MEASURE?<lf>

3) Format of response:

34<lf> ASCII string indicating number of bytes to follow

<34-byte binary data> Note: no terminating linefeed character

Table 7. Measurement Data Format

Byte	**Data**
1-2	Frequency (16-bit, lsb = 0.1Hz)
3-4	Reserved
5-6	Temperature (16-bit signed, lsb = 0.1ºC)
7-8	Phase A voltage rms (16-bit, lsb = 0.1 volt)
9-10	Phase B voltage rms (16-bit, lsb = 0.1 volt)
11-12	Phase C voltage rms (16-bit, lsb = 0.1 volt)
13-14	Phase A voltage peak (16-bit, lsb = 0.1 volt)
15-16	Phase B voltage peak (16-bit, lsb = 0.1 volt)
17-18	Phase C voltage peak (16-bit, lsb = 0.1 volt)
19-20	Phase A DC component (16-bit, lsb = 1 mV DC)
21-22	Phase B DC component (16-bit, lsb = 1 mV DC)
23-24	Phase C DC component (16-bit, lsb = 1 mV DC)
25-26	Phase A Phase Angle (16-bit, lsb = 0.1 degree) relative to Phase A
27-28	Phase B Phase Angle (16-bit, lsb = 0.1 degree) relative to Phase A
29-30	Phase C Phase Angle (16-bit, lsb = 0.1 degree) relative to Phase A
31	Phase A THD (8-bit, lsb = 0.1%)
32	Phase B THD (8-bit, lsb = 0.1%)
33	Phase C THD (8-bit, lsb = 0.1%)
34	PowerOK (8-bit, 1 = power good, 0 = no good)

b. BITRESULT?

1) Description: Queries data structure which contains measured values for selt-test parameters such as internal power supply voltages. Response to command will consist of 2 parts. First part will be an ASCII string indicating number of bytes to follow. Second part will be a binary message with no terminating linefeed character.

2) Format of command:

    BITRESULT?<lf>

3) Format of response:

    24<lf> ASCII string indicating number of bytes to follow
    <24-byte binary data> Note: no terminating linefeed character

Table 8. BIT Result Data Format

Byte	Data
1-2	bitStatus (16-bit, see description above)
3-4	Half-Ref voltage (16-bit, lsb = 4.89mV)
5-6	VREF voltage (16-bit, lsb = 4.89mV)
7-8	+5V voltage (16-bit, lsb = 9.78mV)
9-10	+5VA voltage (16-bit, lsb = 9.78mV)
11-12	Relay Drive voltage (16-bit, lsb = 53.76mV)
13-14	Reserved (16-bit)
15-16	Reserved (16-bit)
17-18	Phase A input test voltage (16-bit, lsb = 4.89mV) - reserved
19-20	Phase B input test voltage (16-bit, lsb = 4.89mV) - reserved
21-22	Phase C input test voltage (16-bit, lsb = 4.89mV) - reserved
23-24	Temperature (16-bit signed, lsb = 0.1ºC)

11. DATA LOGGING COMMANDS

Data logging commands provide access to a continually updated non-volatile record of AC power source that is being monitored. Records consist of 64 bytes of binary data representing parameters of the AC power source. Minimum and maximum voltage, frequency, DC content, distortion and peak voltage value as well as output status, trip and warning conditions and status change events are recorded. A 32-bit CRC is provided for each record to verify it's integrity. Records are saved when an event occurs (i.e. trip, power-up, turn-off) and at fixed time intervals. Up to 65,536 64-byte records are stored in a circular buffer in FLASH memory. The format of the data record is similar to that used in MIL-PRF-24021 shipboard aircraft external power monitors. All commands in this group except ERASEFLASH are effected by optional multi-drop addressability mode. See paragraph 4. See detailed information below.

a. Data record format.

Table 9. Data Record Format

Byte	Data	Units	Data Type
0	Event Type		8-bit UINT
1	Operating State		8-bit UINT
2-4	Power Up Count		32-bit UINT
5-9	Elapsed Time since Power Up	1 ms	32-bit UINT
10-13	Trip Flag		32-bit UINT
14-17	Caution Flag		32-bit UINT
18-19	Phase A Voltage Min	0.1 V rms	16-bit UINT
20-21	Phase A Voltage Max	0.1 V rms	16-bit UINT
22-23	Phase B Voltage Min	0.1 V rms	16-bit UINT
24-25	Phase B Voltage Max	0.1 V rms	16-bit UINT
26-27	Phase C Voltage Min	0.1 V rms	16-bit UINT
28-29	Phase C Voltage Max	0.1 V rms	16-bit UINT
30-31	Frequency Min	0.1 Hz	16-bit UINT
32-33	Frequency Max	0.1 Hz	16-bit UINT
34-35	Phase A Dc Content Min	1 mV	16-bit UINT
36-37	Phase A Dc Content Max	1 mV	16-bit UINT
38-39	Phase B Dc Content Min	1 mV	16-bit UINT
40-41	Phase B Dc Content Max	1 mV	16-bit UINT
42-43	Phase C Dc Content Min	1 mV	16-bit UINT
44-45	Phase C Dc Content Max	1 mV	16-bit UINT
46	Phase A Distortion Min	0.1 %	8-bit UINT
47	Phase A Distortion Max	0.1 %	8-bit UINT
48	Phase B Distortion Min	0.1 %	8-bit UINT
49	Phase B Distortion Max	0.1 %	8-bit UINT
50	Phase C Distortion Min	0.1 %	8-bit UINT
51	Phase C Distortion Max	0.1 %	8-bit UINT
52-53	Phase A Peak Voltage Min	0.1 V	16-bit UINT
54-55	Phase A Peak Voltage Max	0.1 V	16-bit UINT
56-57	Phase B Peak Voltage Min	0.1 V	16-bit UINT
58-59	Phase B Peak Voltage Max	0.1 V	16-bit UINT
60-61	Phase C Peak Voltage Min	0.1 V	16-bit UINT
62-63	Phase C Peak Voltage Max	0.1 V	16-bit UINT
64-67	CRC-32		32-bit UINT

b. Events that cause record to be saved.

Table 10. Data Logging Events

Event	Byte Value
Max Interval	0
Power Up	1
Power Down	2
Off	3
Reset	4
Trip	5
Fail	6
Output On	7
Output Off	8

c. Time interval for saving records.

1) When output is on (RESET state) or tripped (TRIPPED state), records will be logged every 2 minutes.
2) When output is off (OFF state) records will be logged every 10 minutes.

d. 32-bit CRC calculation.

1) CRC is calculated consistent with the following C program excerpts.

```
unsigned long crc_table[256];
void GenerateCRCTable()
{
        unsigned long crc, poly;
         int        i, j;

        poly = 0xEDB88320L;
        for (i=0; i < 256; i++)
        {
                crc = i
                for (j = 8; j > 0; j--)
                {
                        if (crc & 1)
                                crc = (crc >> 1) ^ poly;
                        else
                                crc >>= 1;
                }
                crc_table[i] = crc;
        }
}

unsigned long CalculateCRC32 (unsigned char *buf, int ByteCount)
{
        unsigned long crc;
        int i, j, ch;

        crc = 0xFFFFFFFF;
        for (i=0; i < ByteCount; i++)
        {
                ch = *buf++;
                j = 0;
                crc = (crc>>8) ^ crc_table[(crc ^ ch) & 0xFF];
        }
         return (crc ^ 0xFFFFFFFF);
}
```

e. RECCOUNT?

1) Description: Queries the number of records containing data.
2) Format of command:

    RECCOUNT?<lf>

3) Format of response (ASCII string):

    <count><lf> if ok, where <count> is 0 - 65535
    ERR03 - CMD NOT SUPPORTED<lf> if monitor has firmware revision below A

f. RECSIZE?

1) Description: Queries size in byte of all records containing data. This will be equal to the RECCOUNT value times 68.

2) Format of command:
RECSIZE?<lf>

3) Format of response (ASCII string):
<size><lf> if ok, where <size> = number of records containing data times 68
ERR03 - CMD NOT SUPPORTED<lf> if monitor has firmware revision below A

g. HIGH?

1) Description: Queries the record number (0 - 65535) of the highest record containing data. When the circular buffer wraps around HIGH value will be 0.

2) Format of command:
HIGH?<lf>

3) Format of response (ASCII string):
<record number><lf> if ok, where <record number> is 0 - 65535
ERR03 - CMD NOT SUPPORTED<lf> if monitor has firmware revision below A

h. NEXT?

1) Description: Queries the record number (0 - 65535) of the next record that will be stored. This value will be equal to HIGH value + 1.

2) Format of command:
NEXT?<lf>

3) Format of response (ASCII string):
<record number><lf> if ok, where <record number> is 0 - 65535
ERR03 - CMD NOT SUPPORTED<lf> if monitor has firmware revision below A

i. RECORD?

1) Description: Queries the most current data record. Response to command will consist of 2 parts. First part will be an ASCII string indicating number of bytes to follow. Second part will be a binary message with no terminating linefeed character. Record will be 68 bytes of binary data in accordance with Table 9.

2) Format of command:
RECORD?<lf>

3) Format of response:
68<lf> ASCII string indicating number of bytes to follow
<68-byte binary data> Note: no terminating linefeed character
ERR03 - CMD NOT SUPPORTED<lf> if monitor has firmware revision below A

j. LASTTRIP?

1) Description: Queries the data record associated with the most recent TRIP event. Response to command will consist of 2 parts. First part will be an ASCII string indicating number of bytes to follow. Second part will be a binary message with no terminating linefeed character. Record will be 68 bytes of binary data in accordance with Table 9.

2) Format of command:
RECORD?<lf>

3) Format of response:
68<lf> ASCII string indicating number of bytes to follow
<68-byte binary data> Note: no terminating linefeed character
ERR08 - INVALID RECORD<lf> if no TRIP record found
ERR03 - CMD NOT SUPPORTED<lf> if monitor has firmware revision below A

k. GET

1) Description: Queries the data records specified by <start rec> thru <end rec>. If <start rec> and <end rec> are same, 1 record will be returned. Response to command will consist of 2 parts. First part will be an ASCII string indicating number of bytes to follow Number of bytes will be (<end rec> - <start rec> + 1) * 68. Second part will be a binary message with no terminating linefeed character. See Table 9.

2) Format of command:

GET<sp><start rec>,<end rec><lf>

3) Format of response:

<number of bytes><lf> ASCII string indicating number of bytes to follow
< <number of bytes> bytes binary data> Note: not terminating linefeed character, if normal, or
ERR03 - CMD NOT SUPPORTED<lf> if monitor has firmware revision below A

l. CLEARCOUNT

1) Description: Clears the record count (RECCOUNT) to 0 but leaves the NEXT record pointer intact. Note: This command requires CONFIG mode to be ON.

2) Format of command:

CLEARCOUNT<lf>

3) Format of response (ASCII string):

OK<lf> if normal or
ERR03 - CMD NOT SUPPORTED<lf> if monitor has firmware revision below A

m. ERASEFLASH

1) Description: Erases all data from FLASH memory but leaves the NEXT record pointer intact.

2) Format of command:

ERASEFLASH<lf>

3) Format of response (ASCII string):

OK<lf> if normal or
ERR03 - CMD NOT SUPPORTED<f> if monitor has firmware revision below A

12. CONFIGURATION COMMANDS

Configuration commands are used to set operational configuration and identification of the power monitor. The PART NUMBER parameter sets limits and conditions for monitoring the input AC power. Commands are provided to configure RS485 baud rate and optional multi-drop addressable mode.

a. SETCONFIG

1) Description: Enables access to configuration commands. Configuration commands are ignored by the iPM if sent when configuration mode is not enabled. Configuration mode is set OFF at power-up. This command is found in firmware version 4 and up.

2) Format of command:

SETCONFIG<sp>ON<lf> enables configuration mode
SETCONFIG<sp>OFF<lf> disables configuration mode

3) Format of response:

No response

b. CONFIG?

1) Description: Queries the state of configuration mode. This command is found in firmware version 4 and up.

2) Format of command:

CONFIG?<lf>

3) Format of response (ASCII string):

ON<lf> or
OFF<lf>

c. SETPARTNO

1) Description: Programs the PART NUMBER parameter which sets the iPM monitor limits and set of conditions that are monitored. Limits and features programmed with the new part number will take effect at next power up. See iPM specification iPMA001 for details of the part number format. This command requires configuration mode to be ON when unit uses firmware revision 4 and up.

2) Format of command (includes 39 ASCII character part number similar to example below):

SETPARTNO<sp>iPM42A0101(047-067)(104-121)(005-010)00<lf><lf>

3) Format of response:

OK<lf> if normal or
ERR04 - INVALID PART NUMBER<lf> if format of part number is incorrect

d. SETSERNO

1) Description: Programs the part SERIAL NUMBER of the iPM. This value is set at the factory to identify the unit and is important for customer service and warranty issues. **This value should not be changed by the end user**. This command requires configuration mode to be ON when unit uses firmware revision 4 and up.

2) Format of command (includes up to 31 ASCII character serial number):

SETSERNO<sp><serial number><lf>

3) Format of response:

OK<lf> if normal or
ERR05 - INVALID SERIAL NUMBER<lf> if format of serial number is incorrect

e. CALDELTA

1) Description: Calibrates accuracy of 3-wire delta configured iPM. When unit is programmed with a PART NUMBER starting with “iPM3…” this calibration step is required. This command requires configuration mode to be ON when unit uses firmware revision 4 and up.

To calibrate, proceed as follows:

Apply 400Hz, 115V, 3-phase wye-connected AC signal to input of unit.
Disconnect the neutral wire from the iPM.
Send the CALDELTA command to the iPM.
Calibration will be active on next power up.

2) Format of command:

CALDELTA<lf>

3) Format of response:

OK<lf> if normal or
ERR06 - FAILED TO CAL<lf>

f. SETBAUD

1) Description: Programs the baud rate of the RS485 serial port. An index value of 0 to 15 is used to set the baud rate. See Table 3 for baud rate associated with each index value. This command must be followed with the YES command complete setup of new baud rate. The newly programmed baud rate will take effect at next power up. **Use caution with this command because it can potentially make communication with the iPM impossible**. This command requires configuration mode to be ON. This command is found in firmware version 4 and up.

2) Format of command:

SETBAUD<sp><baud rate index><lf>

3) Format of response:

ARE YOUR SURE?<lf> if normal (follow up with YES command) or
ERR07 - INVALID VALUE<lf>

g. BAUD?

1) Description: Queries the programmed baud rate of the iPM. This command requires configuration mode to be ON. This command is found in firmware version 4 and up.
2) Format of command:

   BAUD?<lf>
3) Format of response (ASCII string):

   <index><baud rate><lf> where <index> is 0 to 15 and <baud rate> is the baud rate value.

h. SETADDRESS

1) Description: Programs the **optional** multi-drop address of the RS485 serial port. A value of 0 to 255 will be accepted. See paragraph 4 for description of optional multi-drop addressability feature. This command must be followed with the YES command complete setup of new address. The newly programmed address will take effect at next power up. **Use caution with this command because it can potentially make communication with the iPM impossible**. This command requires configuration mode to be ON. This command is found in firmware version 4 and up.
2) Format of command:

   SETADDRESS<sp><address><lf> where <address> is 0 to 255
3) Format of response:

   ARE YOUR SURE?<lf> if normal (follow up with YES command) or
   ERR07 - INVALID VALUE<lf>

i. ADDRESS?

1) Description: Queries the programmed **optional** multi-drop address of the iPM. This command requires configuration mode to be ON. This command is found in firmware version 4 and up.
2) Format of command:

   ADDRESS?<lf>
3) Format of response (ASCII string):

   <address><lf> where <address> is 0 to 255

j. SET MULTIDROP

1) Description: Programs the state of **optional** multi-drop addressability feature of the RS485 serial port. Feature may be set ON or OFF. See paragraph 4 for description of optional multi-drop addressability feature. This command must be followed with the YES command complete setup of new addressability state. The newly programmed addressability state will take effect at next power up. **Use caution with this command because it can potentially make communication with the iPM impossible**. This command requires configuration mode to be ON. This command is found in firmware version 4 and up.
2) Format of command:

   SETMULTIDROP<sp>ON<lf> to enable feature or
   SETMULTIDROP<sp>OFF<lf> to disable feature
3) Format of response:

   ARE YOUR SURE?<lf> if normal (follow up with YES command) or
   ERR07 - INVALID VALUE<lf>

k. MULTIDROP?

1) Description: Queries the programmed **optional** multi-drop addressability state of the iPM. This command requires configuration mode to be ON. This command is found in firmware version 4 and up.
2) Format of command:

   MULTIDROP?<lf>
3) Format of response (ASCII string):

   ON<lf> if enabled or
   OFF<lf> if disabled

l. YES

1) Description: Provides confirmation to execute SETBAUD, SETADDRESS and SET MULTIDROP commands. YES command is required to complete execution. This command requires configuration mode to be ON. This command is found in firmware version 4 and up.
2) Format of command:

   YES<lf>
3) Format of response (ASCII string):

   OK<lf>

13. SOFT PANEL OPERATION

North Atlantic makes available a LabWindows/CVI driver dll and soft panel application for configuration of the iPM Power Monitor. The soft panel application provides an easy to read presentation of the all data from the power monitor and access to all configuration commands.

a. Computer resources needed.

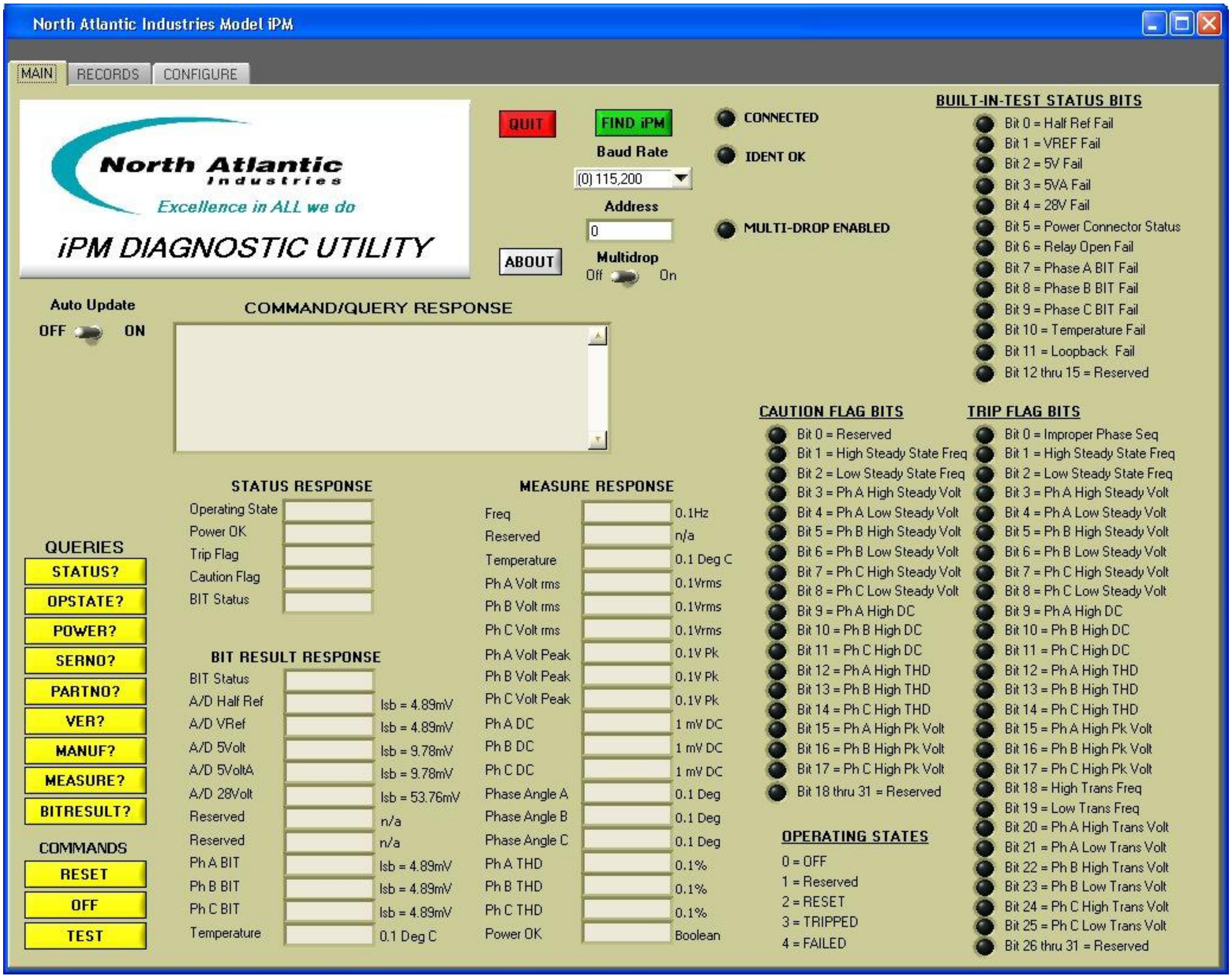
The driver library and soft panel application require a computer with Windows operating system and an available serial port. If the serial port is an RS232 type, an external adapter will be needed to provide RS485 compatible signals. Driver has only been tested with WindowsXP. It may work with other versions of Windows. To run the application, the CVI runtime engine and VISA software layer must be installed. The installer will provide these items if they are not already installed on the computer.

b. Installation of dll and Soft Panel.

A zip compressed driver installer file is provided on the setup disk/information disk. Insert CD. If it does not start automatically, navigate to CD drive and run autorun.exe. When browser opens, click on “click to browse software and documents” then click “iPM VISA Driver”. Extract files and run setup.exe. Follow prompts to install program.

c. Startup of Soft Panel.

After program is installed, navigate to Start>North Atlantic Industries>naIPM>naIPM Soft Panel Application to start the soft panel. It will appear as shown below.



STATUS RESPONSE

- Operating State
- Power OK
- Trip Flag
- Caution Flag
- BIT Status

BIT RESULT RESPONSE

BIT Status	
A/D Half Ref	lsb = 4.89mV
A/D VRef	lsb = 4.89mV
A/D 5Volt	lsb = 9.78mV
A/D 5VoltA	lsb = 9.78mV
A/D 28Volt	lsb = 53.76mV
Reserved	n/a
Reserved	n/a
Ph A BIT	lsb = 4.89mV
Ph B BIT	lsb = 4.89mV
Ph C BIT	lsb = 4.89mV
Temperature	0.1 Deg C

MEASURE RESPONSE

Freq	0.1Hz
Reserved	n/a
Temperature	0.1 Deg C
Ph A Volt rms	0.1Vrms
Ph B Volt rms	0.1Vrms
Ph C Volt rms	0.1Vrms
Ph A Volt Peak	0.1V Pk
Ph B Volt Peak	0.1V Pk
Ph C Volt Peak	0.1V Pk
Ph A DC	1 mV DC
Ph B DC	1 mV DC
Ph C DC	1 mV DC
Phase Angle A	0.1 Deg
Phase Angle B	0.1 Deg
Phase Angle C	0.1 Deg
Ph A THD	0.1%
Ph B THD	0.1%
Ph C THD	0.1%
Power OK	Boolean

BUILT-IN-TEST STATUS BITS

- Bit 0 = Half Ref Fail
- Bit 1 = VREF Fail
- Bit 2 = 5V Fail
- Bit 3 = 5VA Fail
- Bit 4 = 28V Fail
- Bit 5 = Power Connector Status
- Bit 6 = Relay Open Fail
- Bit 7 = Phase A BIT Fail
- Bit 8 = Phase B BIT Fail
- Bit 9 = Phase C BIT Fail
- Bit 10 = Temperature Fail
- Bit 11 = Loopback Fail
- Bit 12 thru 15 = Reserved

CAUTION FLAG BITS

- Bit 0 = Reserved
- Bit 1 = High Steady State Freq
- Bit 2 = Low Steady State Freq
- Bit 3 = Ph A High Steady Volt
- Bit 4 = Ph A Low Steady Volt
- Bit 5 = Ph B High Steady Volt
- Bit 6 = Ph B Low Steady Volt
- Bit 7 = Ph C High Steady Volt
- Bit 8 = Ph C Low Steady Volt
- Bit 9 = Ph A High DC
- Bit 10 = Ph B High DC
- Bit 11 = Ph C High DC
- Bit 12 = Ph A High THD
- Bit 13 = Ph B High THD
- Bit 14 = Ph C High THD
- Bit 15 = Ph A High Pk Volt
- Bit 16 = Ph B High Pk Volt
- Bit 17 = Ph C High Pk Volt
- Bit 18 thru 31 = Reserved

OPERATING STATES

- 0 = OFF
- 1 = Reserved
- 2 = RESET
- 3 = TRIPPED
- 4 = FAILED

TRIP FLAG BITS

- Bit 0 = Improper Phase Seq
- Bit 1 = High Steady State Freq
- Bit 2 = Low Steady State Freq
- Bit 3 = Ph A High Steady Volt
- Bit 4 = Ph A Low Steady Volt
- Bit 5 = Ph B High Steady Volt
- Bit 6 = Ph B Low Steady Volt
- Bit 7 = Ph C High Steady Volt
- Bit 8 = Ph C Low Steady Volt
- Bit 9 = Ph A High DC
- Bit 10 = Ph B High DC
- Bit 11 = Ph C High DC
- Bit 12 = Ph A High THD
- Bit 13 = Ph B High THD
- Bit 14 = Ph C High THD
- Bit 15 = Ph A High Pk Volt
- Bit 16 = Ph B High Pk Volt
- Bit 17 = Ph C High Pk Volt
- Bit 18 = High Trans Freq
- Bit 19 = Low Trans Freq
- Bit 20 = Ph A High Trans Volt
- Bit 21 = Ph A Low Trans Volt
- Bit 22 = Ph B High Trans Volt
- Bit 23 = Ph B Low Trans Volt
- Bit 24 = Ph C High Trans Volt
- Bit 25 = Ph C Low Trans Volt
- Bit 26 thru 31 = Reserved

d. Find iPM.
Assure that iPM is connected to the computer RS485 port and that the iPM is powered on. Review the baud rate, multi-drop state and address settings on the Soft Panel to assure that they match the unit you wish to communicate with. Click the FIND iPM button. The CONNECTED and IDENT OK indicators should light up green. If your unit has optional multi-drop addressability enabled, the MULTI-DROP ENABLED indicator will light up yellow. You are now ready to communicate with the iPM.

e. Basic data monitoring and connection (MAIN) tab page.

Click, in turn, each of the yellow query buttons on the panel. Results should be as follows.
STATUS?: Fills in the STATUS RESPONSE table and decodes the Trip Flag, Caution Flag and BIT Status binary values and lights LED indicators as needed on right side of panel to correspond to any out-of-limit values.

OPSTATE?: Reports Operating State in the COMMAND/QUERY RESPONSE box.

POWER?: Reports POWER status in the COMMAND/QUERY RESPONSE box. 1 = Power Ok.

SERNO?: Reports unit serial number in the COMMAND/QUERY RESPONSE box.

PARTNO?: Reports unit part number in the COMMAND/QUERY RESPONSE box.

VER?: Reports unit software version in the COMMAND/QUERY RESPONSE box.

MANUF?: Reports unit manufacturer ID in the COMMAND/QUERY RESPONSE box.

MEASURE?: Fills in the MEASURE RESPONSE table with decoded real time parameter measurement values.

BITRESULT?: Fills in the BIT RESULT RESPONSE table with decoded real-time internal self-test values.

OFF: Forces monitor output to OFF state regardless of input power quality.

RESET: Set unit to monitor AC power input and turn on output if power is good (default). This is used if unit has been turned off with OFF command or if it has tripped.

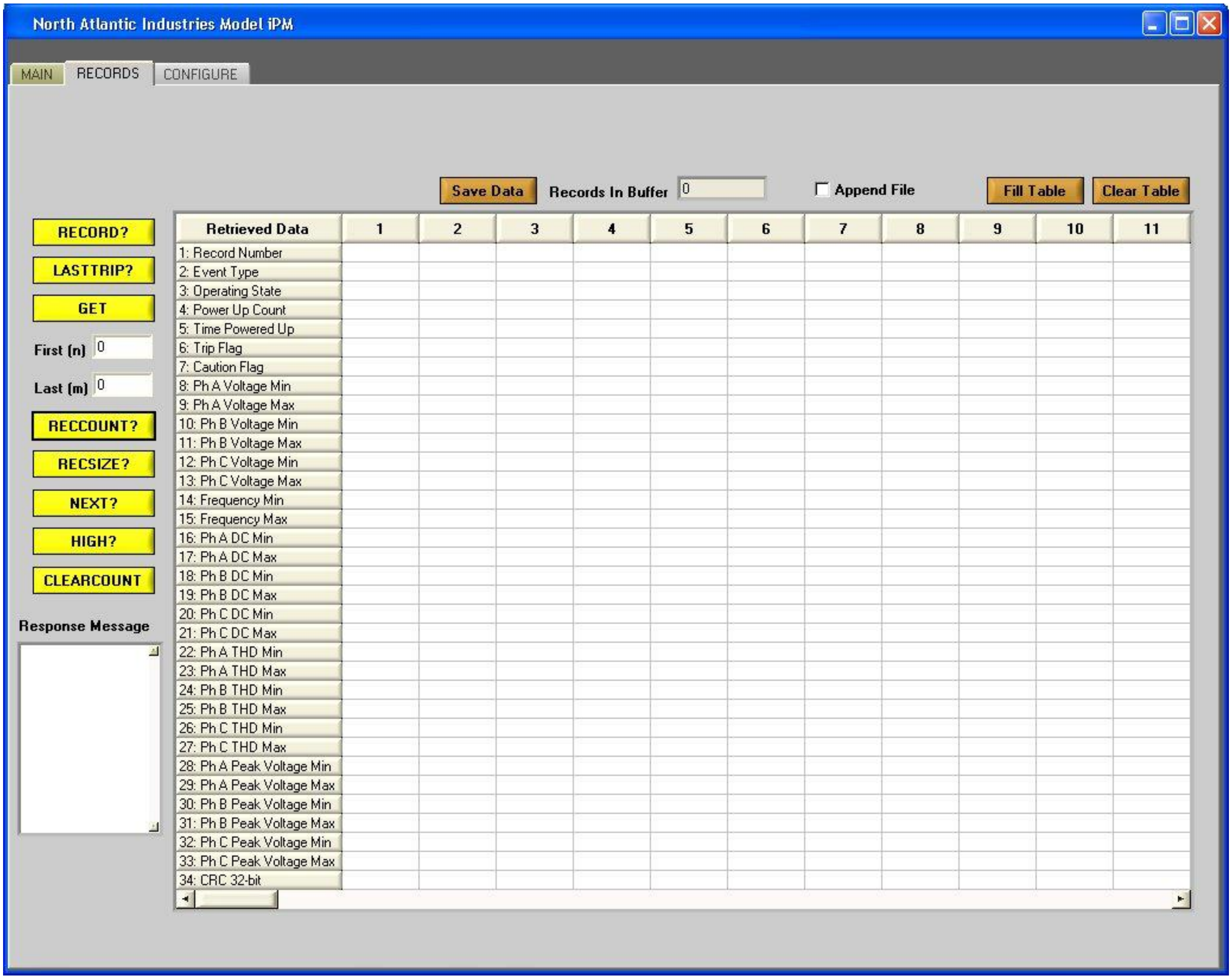
TEST: Initiated execution of internal self-test and updates the BIT result data internal to the iPM.

Click other controls.
ABOUT: Popup displays driver and soft panel version.

AUTO UPDATE: When switch is in ON position, panel will execute the STATUS?, MEASURE? and BITRESULT? commands and fill in all applicable tables and LED's on the panel. This will repeat continuously every 1 second until switch is set to OFF position.

QUIT: Closes driver and Soft Panel.

f. Data log record tab page.
Click the RECORDS tab at top of Soft Panel. The Data Logging page will appear as shown below.



Records In Buffer 0

□ Append File

First (n) 0

Last (m) 0

Response Message

Retrieved Data	1	2	3	4	5	6	7	8	9	10	11
1: Record Number											
2: Event Type											
3: Operating State											
4: Power Up Count											
5: Time Powered Up											
6: Trip Flag											
7: Caution Flag											
8: Ph A Voltage Min											
9: Ph A Voltage Max											
10: Ph B Voltage Min											
11: Ph B Voltage Max											
12: Ph C Voltage Min											
13: Ph C Voltage Max											
14: Frequency Min											
15: Frequency Max											
16: Ph A DC Min											
17: Ph A DC Max											
18: Ph B DC Min											
19: Ph B DC Max											
20: Ph C DC Min											
21: Ph C DC Max											
22: Ph A THD Min											
23: Ph A THD Max											
24: Ph B THD Min											
25: Ph B THD Max											
26: Ph C THD Min											
27: Ph C THD Max											
28: Ph A Peak Voltage Min											
29: Ph A Peak Voltage Max											
30: Ph B Peak Voltage Min											
31: Ph B Peak Voltage Max											
32: Ph C Peak Voltage Min											
33: Ph C Peak Voltage Max											
34: CRC 32-bit											

Functions performed by command buttons on this panel are as follows.
RECORD?: Reads the current data log record from the iPM and displays it in the RETRIEVED DATA table.

LASTTRIP?: Reads the data log record for the most recent TRIP event from the iPM and displays it in the RETRIEVED DATA table.

GET: Reads data log records numbered from FIRST to LAST as indicated by the boxes on the panel. Retrieved records may be displayed using FILL TABLE button and/or saved to a file using the SAVE DATA button.

RECCOUNT?: Reads the number of data log records stored from the iPM and shows result in RESPONSE MESSAGE box.

RECSIZE?: Reads the size of stored records (in bytes) that is stored in the iPM and displays this data in the RESPONSE MESSAGE box.

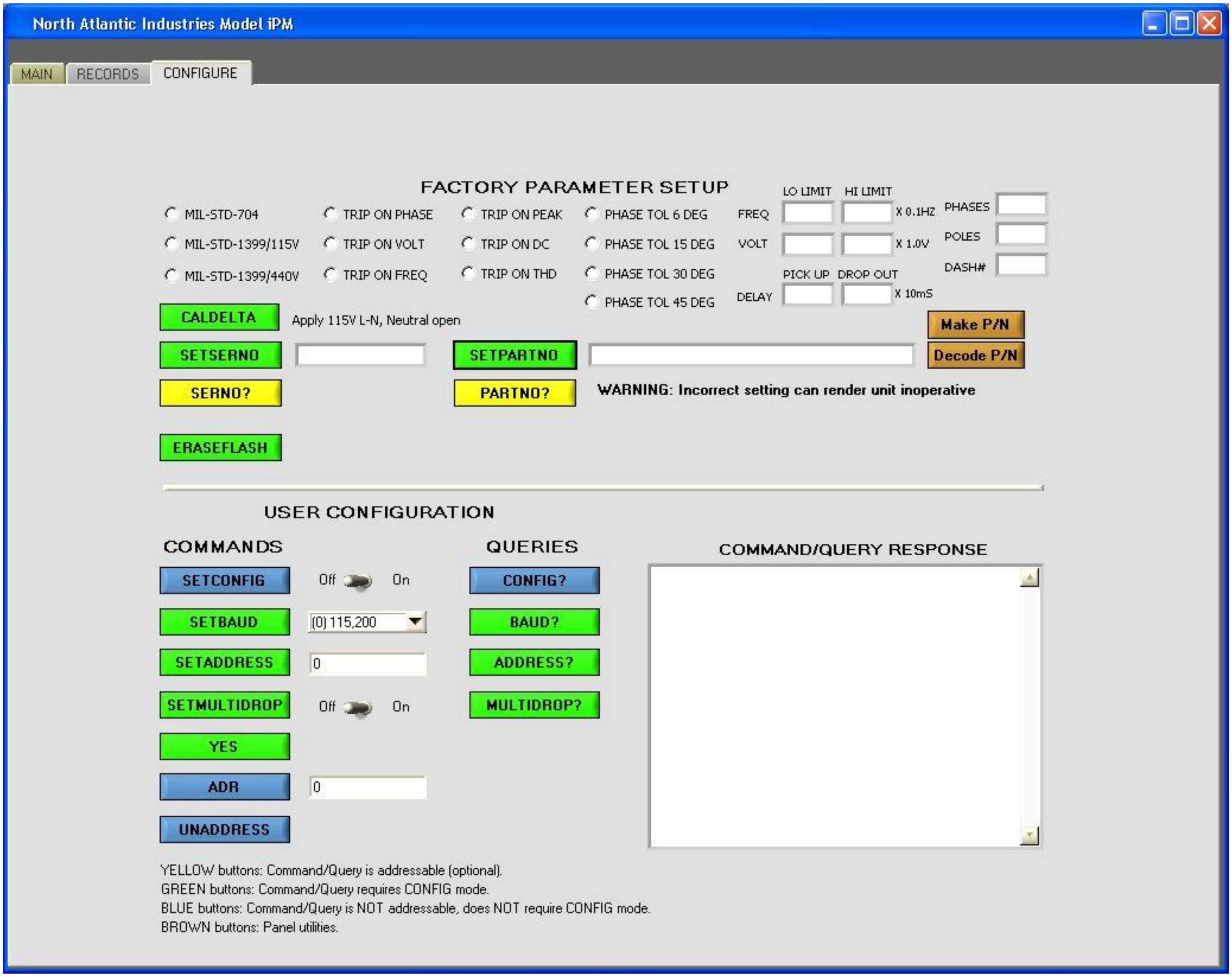
NEXT?: Reads the record number of the next data log record to be written from the iPM and displays it in the RESPONSE MESSAGE box.

HIGH?: Reads the record number of the highest data log record stored from the iPM and displays it in the RESPONSE MESSAGE box.

CLEARCOUNT: Clears the number of data log records (RECCOUNT) to zero.

CLEAR TABLE: Clears the displayed data from the RETRIEVED DATA table.

g. Configuration tab page.
Click the CONFIGURE tab at top of Soft Panel. The Configuration page will appear as shown below.



Apply 115V L-N, Neutral open

WARNING: Incorrect setting can render unit inoperative

SETCONFIG Off On

SETBAUD (0) 115,200

SETADDRESS 0

SETMULTIDROP Off On

ADR 0

YELLOW buttons: Command/Query is addressable (optional).
GREEN buttons: Command/Query requires CONFIG mode.
BLUE buttons: Command/Query is NOT addressable, does NOT require CONFIG mode.
BROWN buttons: Panel utilities.

All of the GREEN buttons on this panel require Configuration Mode to be ON to use them. YELLOW buttons provide query of part number and serial number and are same as those on MAIN tab page. BLUE buttons provide other

configuration and control functions. BROWN buttons provide utility functions. Function of command buttons on this panel are as follows.
SERNO?: Reads serial number from iPM and displays value COMMAND/QUERY RESPONSE box.

PARTNO?: Reads part number from iPM and displays value in COMMAND/QUERY RESPONSE box.

SETCONFIG: Sets Configuration Mode on or off according the setting of switch on Soft Panel. Configuration mode must be on for all GREEN buttons on configuration tab panel. Configuration mode defaults to OFF when iPM is powered up

CONFIG?: Reads the current state of configuration mode in the iPM and displays it in the COMMAND/QUERY RESPONSE box.

SETSERNO: Sets iPM serial number according to value in box next to button. This is a factory setting that should not be changed by end user. New serial number will take effect at next power up.

SETPARTNO: Sets iPM part number which controls functions and trip limits for monitoring applied AC power. This is set at factory but may be modified by the end user. Use great caution since this setting can drastically change the operation of the iPM. There are 2 convenient ways to use this configuration function.

1) Enter the desired settings using the boxes and radio buttons shown beneath FACTORY PARAMETER SETUP label. Once desired settings have been selected, click the MAKE P/N button and the part number box will be filled with a properly formatted part number. Click the SETPARTNO button to send command to iPM. New part number settings will take effect next time unit is powered on.
2) Click the PARTNO? button to read the present part number from the iPM. Cut and paste this text string into the box next to the SETPARTNO button. Edit this string to make desired changes to the iPM setup. When done, click the READ P/N to verify that the part number is valid and populate the parameter boxes and radio buttons to indicate what the settings are. Click the SETPARTNO button to send command to iPM. New part number settings will take effect next time unit is powered on.

CALDELTA: Calibrates iPM measurement system for 3-wire delta configuration. First apply 400Hz, 115V line-to-neutral 4-wire wye configured AC power to the iPM. Next, remove the neutral connection from the iPM. Finally, click the CALDELTA button to execute and store the delta configuration calibration value. Power unit off and back on for the calibration to take effect.

<u>Baud rate setup</u>
SETBAUD: Use the drop-down box to select the desired baud rate. Click the SETBAUD button. The new baud rate value will take effect the next time iPM is powered up. Note: factory default baud rate is 115.2K baud. **Use this command cautiously because it can make it impossible to communicate with the iPM if not used properly.**

BAUD? Reads the programmed baud rate value from the iPM and displays it in the COMMAND/QUERY RESPONSE box.

<u>Multi-drop address value setup</u>
SETADDRESS: Enter the desired address in the box provided. Permitted values are 0 to 255. Click the SETADDRESS button. The new address value will take effect the next time iPM is powered up. **Use this command cautiously because it can make it impossible to communicate with the iPM if not used properly.**

ADDRESS? Reads the programmed address value from the iPM and displays it in the COMMAND/QUERY RESPONSE box.

<u>Multi-drop addressability state setup</u>
SETMULTIDROP: Set multi-drop state toggle switch to ON or OFF as required. Click SETMULTIDROP button. The new multi-drop addressability state will take effect the next time iPM is powered up. **Use this command cautiously because it can make it impossible to communicate with the iPM if not used properly.**

MULTIDROP?: Reads the programmed multi-drop addressability state from the iPM and displays it in the COMMAND/QUERY RESPONSE box.

ADR: Sends broadcast command to request iPM with matching address to listen to all address restricted commands. The address is entered in the text box next to the button. Command has no effect on units that do not have a matching address or are not setup for multi-drop addressability.

UNADDRESS: Sends broadcast command to request all iPM’s on RS485 to stop listening to address restricted commands, if they are configured for multi-drop addressability.

14. COMMAND LIBRARY DLL

Appendix A contains content of the LabWindows/CVI dll documentation file for all commands except configuration commands. This is derived from naIPM.doc file which extracts data from naIPM.fp function panel file and naIPM.dll library file.

Appendix B contains content of the LabWindows/CVI dll documentation file for configuration commands only. This is derived from naIPMdiag.doc file which extracts data from naIPMdiag.fp function panel file and naIPMdiag.dll library file.

Content of Appendix A and Appendix B provide details needed for using library functions contained in the driver.

APPENDIX A. LabWindows/CVI Driver Documentation File for naIPM.dll (general communication)

North Atlantic Industries IPM

Introduction:

This instrument driver provides programming support for North Atlantic Industries IPM. It contains functions for opening, configuring, taking measurements from, and closing the instrument.

Assumptions:

To successfully use this module, the following conditions must be met:

For GPIB instrument drivers:

- the instrument is connected to the GPIB.
- the GPIB address supplied to the initialize function must match the GPIB address of the instrument.

For VXI instrument drivers:

- the instrument is installed in the VXI mainframe and you are using one of the following controller options:
  Embedded controller
  MXI
  MXI2
  GPIB-VXI
- the logical address supplied to the initialize function must match the logical address of the instrument.

For RS-232 instrument drivers:

- the instrument is connected to the RS-232 interface.
- the COM port, baud rate, parity, and timeout supplied to the initialize function must match the settings of the instrument.

Error and Status Information:

Each function in this instrument driver returns a status code that either indicates success or describes an error or warning condition. Your program should examine the status code from each call to an instrument driver function to determine if an error occurred. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

The description of each instrument driver function lists possible error codes and their meanings

---

How To Use This Document:

Use this document as a programming reference manual. It describes each function in the

North Atlantic Industries IPM

instrument. The functions appear in alphabetical order, with a description of the function and its C syntax, a description of each parameter, and a list of possible error codes.

```
________________________________________________________________________________

Function Tree Layout:

   Class/Panel Name:                        Function Name:

      Find First                           naIPM_find_first
      Initialize                           naIPM_init
      Set Initialization Options           naIPM_set_init_options
      Query Initialization Options         naIPM_query_init_options
      Query Functions
         MANUF (Mfr ID) Query              naIPM_query_MANUF
         PARTNO(Part Number)  Query        naIPM_query_PARTNO
         SERNO (Serial Number) Query       naIPM_query_SERNO
         VER (Firmware Version) Query      naIPM_query_VER
         POWER (Power Quality) Query       naIPM_query_POWER
         OPSTATE (Operating State) Query   naIPM_query_OPSTATE
         STATUS (Monitor Summary) Query    naIPM_query_STATUS
         MEASURE (Parameter Data) Query    naIPM_query_MEASURE
         BITRESULT (Test Result)Query      naIPM_query_BITRESULT
      Command Functions
         OFF (Force Off State) Command     naIPM_cmd_OFF
         RESET (Normal Monitor) Command    naIPM_cmd_RESET
         TEST (Self-Test) Command          naIPM_cmd_TEST
         ADR (Active Address) Command      naIPM_cmd_ADR
         UNADDRESS (Unlisten) Command      naIPM_cmd_UNADDRESS
      Data Logging Funcitons
         RECCOUNT (Record Count) Query     naIPM_query_RECCOUNT
         RECSIZE (Record Size) Query       naIPM_query_RECSIZE
         HIGH (High Record Number) Query   naIPM_query_HIGH
         NEXT (Next Record Number) Query   naIPM_query_NEXT
         LASTTRIP (Trip Record) Query      naIPM_query_LASTTRIP
         RECORD (Current Record) Query     naIPM_query_RECORD
         GET (Read Records) Query          naIPM_query_GET
         CLEARCOUNT (Zero Count) Command   naIPM_query_CLEARCOUNT
      Utility Functions
         Reset                             naIPM_reset
         Self-Test                         naIPM_self_test
         Error-Query                       naIPM_error_query
         Error Message                     naIPM_error_message
         Revision Query                    naIPM_revision_query
      Close                                naIPM_close
________________________________________________________________________________
```

North Atlantic Industries IPM

This instrument module provides programming support for the <Instrument Name>. The module is divided into the following functions:

Functions/Classes:

(1) Initialize:
This function initializes the instrument and sets it to a default configuration.

(2) Application Functions: (Class)
This class contains high-level, test and measurement routines. These examples call other instrument driver functions to configure, start, and read from the instrument.

(3) Configuration Functions: (Class)
This class of functions configures the instrument by setting acquisition and system configuration parameters.

(4) Action/Status Functions: (Class)
This class of functions begins or terminates an acquisition. It also provides functions which allow the user to determine the current status of the instrument.

(5) Data Functions: (Class)
This class of functions transfers data to or from the instrument.

(6) Utility Functions: (Class)
This class of functions provides lower level functions to communicate with the instrument, and change instrument parameters.

(7) Close:
This function takes the instrument offline.

The following functions are in alphabetical order.

---

naIPM_close

```
    ViStatus naIPM_close (ViSession instrumentHandle);
```

Purpose

This function performs the following operations:
viClose (instrSession) and viClose (rmSession).

Notes:

(1) The instrument must be reinitialized to use it again.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

Return Value

TThe return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_ADR

```
    ViStatus naIPM_cmd_ADR (ViSession instrumentHandle,
                            ViInt16 listenAddress);
```

Purpose

This function sets the intended listener address for optional multi-drop addressing. If this value matches the previously configured multi-drop address within a unit, it will listen to all address-restricted commands until either the UNADDRESS command is sent or a non-matching ADR address is sent.

Units will always listen to this command regardless of multi-drop addressability mode, so it can be broadcast to all units on a multi-drop bus.

THIS COMMAND IS NOT NEEDED FOR A SINGLE LISTENER SETUP.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

listenAddress

Variable Type   ViInt16

This parameter accepts listen address used for multi-drop addressable mode.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_OFF

```
ViStatus naIPM_cmd_OFF (ViSession instrumentHandle,
                        ViChar _VI_FAR response[]);
```

Purpose

This routine sets monitor output uconditionally to OFF state.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "OK" if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_RESET

```
    ViStatus naIPM_cmd_RESET (ViSession instrumentHandle,
                              ViChar _VI_FAR response[]);
```

Purpose

This function sets monitor to RESET operating state. In this state normal monitoring of input parameters and normal trip functions are enabled. RESET state cannot be set if monitor is in the FAIL operationg state.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "OK" if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_TEST

```
    ViStatus naIPM_cmd_TEST (ViSession instrumentHandle,
                             ViChar _VI_FAR response[]);
```

Purpose

This function initiates instrument self-test. Use the naIPM_Query-BITRESULT funciton to read selt-test result.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "OK" if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_UNADDRESS

```
ViStatus naIPM_cmd_UNADDRESS (ViSession instrumentHandle);
```

Purpose

This function cancels the listener address for optional multi-drop addressing. All units receiving this command will not longer listen to address-restricted commands until a new ADR command is received.

Units will always listen to this command regardless of multi-drop addressability mode, so it can be broadcast to all units on a multi-drop bus.

THIS COMMAND IS NOT NEEDED FOR A SINGLE LISTENER SETUP.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_error_message

```
ViStatus naIPM_error_message (ViSession instrumentHandle,
                              ViStatus errorCode,
                              ViChar _VI_FAR errorMessage[]);
```

Purpose

This function takes the Status Code returned by the instrument driver functions, interprets it and returns it as a user readable string.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: VI_NULL

errorCode

Variable Type   ViStatus

This control accepts the Status Code returned from the instrument driver functions.

Default Value: 0 - VI_SUCCESS

errorMessage

Variable Type   ViChar[]

This control returns the interpreted Status Code as a user readable message string.

Notes:

(1) The array must contain at least 256 elements ViChar[256].

Status Codes:

Status	Description
0	No error (the call was successful).
3FFF0005	The specified termination character was read.
3FFF0006	The specified number of bytes was read.
BFFF0000	Miscellaneous or system error occurred.
BFFF000E	Invalid session handle.
BFFF0015	Timeout occurred before operation could complete.
BFFF0034	Violation of raw write protocol occurred.
BFFF0035	Violation of raw read protocol occurred.
BFFF0036	Device reported an output protocol error.
BFFF0037	Device reported an input protocol error.
BFFF0038	Bus error occurred during transfer.
BFFF003A	Invalid setup (attributes are not consistent).
BFFF005F	No listeners condition was detected.
BFFF0060	This interface is not the controller in charge.
BFFF0067	Operation is not supported on this session.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_error_query

```
ViStatus naIPM_error_query (ViSession instrumentHandle,
                            ViPInt32 errorCode,
                            ViChar _VI_FAR errorMessage[]);
```

Purpose

This function reads an error code from the instrument's error queue.

Notes:

(1) If this instrument does not support an Error Query, this function should return the Warning Code 0x3FFC0104 - VI_WARN_NSUP_ERROR_QUERY

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

errorCode

Variable Type   ViInt32 (passed by reference)

This control returns the error code read from the instrument's error queue.

errorMessage

Variable Type   ViChar[]

This control returns the error message string read from the instrument's error message queue.

Notes:

(1) The array must contain at least 256 elements ViChar[256].

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_find_first

```
ViStatus naIPM_find_first (ViChar _VI_FAR description[],
                           ViChar _VI_FAR partNumberString[]);
```

Purpose

This function searches for North Atlantic Model IPMs and returns information for the first one it finds.

- Information returned is Description String and Part Number.

- It searches the system for Serial instruments using string queries.

In addition, information on baud rate, multi-drop enable status and multi-drop address is stored in an internal data structure (for unit with these configuration features).

These configuration parameters may be accessed using the naIPM_query_init_optins function.

These configuration parameters may be set to new values using the naIPM_set_init_options function.

Parameter List

description

Variable Type   ViChar[]

This variable stores a string representing the device descriptor.

Example syntax: ASRL1::INSTR

partNumberString

Variable Type   ViChar[]

This variable stores a string representing the device part number.

Example syntax: iPM33A0000(360-440)(110-130)(005)(005)00

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_init

```
ViStatus naIPM_init (ViRsrc resourceName, ViBoolean IDQuery,
                     ViBoolean resetDevice,
                     ViPSession instrumentHandle);
```

Purpose

This function performs the following initialization actions:

- Opens a session to the Default Resource Manager resource and a session to the specified device using the interface and address specified in the Resource_Name control.

- Performs an identification query on the Instrument.

- Resets the instrument to a known state.

- Sends initialization commands to the instrument that set any necessary programmatic variables such as Headers Off, Short Command form, and Data Transfer Binary to the state necessary for the operation of the instrument driver.

- Returns an Instrument Handle which is used to differentiate between different sessions of this instrument driver.

- Each time this function is invoked a Unique Session is opened. It is possible to have more than one session open for the same resource.

Notes:

(1) If this instrument does not support a Reset, and the Reset control is set to "Reset Device" then this function should return the Warning Code 0x3FFC0102 - VI_WARN_NSUP_RESET.

Parameter List

resourceName

Variable Type   ViRsrc

This control specifies the interface and address of the device that is to be initialized (Instrument Descriptor). The exact grammar to be used in this control is shown in the note below.

Default Value: "GPIB-VXI::1"

Notes:

(1) Based on the Instrument Descriptor, this operation establishes a communication session with a device. The grammar for the Instrument Descriptor is shown below. Optional parameters are shown in square brackets ([]).

Interface	Grammar
VXI	VXI[board]::VXI logical address[::INSTR]
GPIB-VXI	GPIB-VXI[board][::GPIB-VXI primary address] ::VXI logical address[::INSTR]

The VXI keyword is used for VXI instruments via either embedded or MXIbus controllers. The GPIB-VXI keyword is used for a GPIB-VXI controller.

The default value for optional parameters are shown below.

Optional Parameter	Default Value
board	0
GPIB-VXI primary address	1

IDQuery

Variable Type   ViBoolean

This control specifies if an ID Query is sent to the instrument during the initialization procedure.

Valid Range:
VI_OFF (0) - Skip Query
VI_ON (1) - Do Query (Default Value)

Notes:

(1) Under normal circumstances the ID Query ensures that the instrument initialized is the type supported by this driver. However circumstances may arise where it is undesirable to send an ID Query to the instrument. In those cases; set this control to "Skip Query" and this function will initialize the selected interface, without doing an ID Query.

resetDevice

Variable Type   ViBoolean

This control specifies if the instrument is to be reset to its power-on settings during the initialization procedure.

Valid Range:
VI_OFF (0) - Don't Reset
VI_ON (1) - Reset Device (Default Value)

Notes:

(1) If this instrument does not support a Reset, and this control is set to "Reset Device" then this function should return the Warning Code 0x3FFC0102 - VI_WARN_NSUP_RESET.

(2) If you do not want the instrument reset. Set this control to "Don't Reset" while initializing the instrument.

instrumentHandle

Variable Type   ViSession (passed by reference)

This control returns an Instrument Handle that is used in all subsequent function calls to differentiate between different sessions of this instrument driver.

Notes:

(1) Each time this function is invoked a Unique Session is opened. It is possible to have more than one session open for the same resource.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_BITRESULT

```
ViStatus naIPM_query_BITRESULT (ViSession instrumentHandle,
                                ViPInt32 byteCount,
                                ViChar _VI_FAR builtInTestMeasuredValues[]);
```

Purpose

This function queries results of background and commanded self-test.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

byteCount

Variable Type   ViInt32 (passed by reference)

This parameter accepts the number of bytes in the status message. Nominal value is 24.

builtInTestMeasuredValues

Variable Type   ViChar[]

This parameter accepts the returned character (byte) array representing background and commanded self-test measurement result data structure.

Response:

Byte	Data
1-2	bitStatus (16-bit flag register)
3-4	Half-Ref voltage (16-bit, lsb = 4.89mV)
5-6	VREF voltage (16-bit, lsb = 4.89mV)
7-8	+5V voltage (16-bit, lsb = 9.78mV)
9-10	+5VA voltage (16-bit, lsb = 9.78mV)
11-12	Relay Drive voltage (16-bit, lsb = 53.76mV)
13-14	Reserved (16-bit)
15-16	Reserved (16-bit
17-18	Phase A input test voltage (16-bit, lsb = 4.89mV)
19-20	Phase B input test voltage (16-bit, lsb = 4.89mV)
21-22	Phase C input test voltage (16-bit, lsb = 4.89mV)
23-24	Temperature (16-bit signed, lsb = 0.1 deg. C)

Built-In-Test Status (bitStatus) flag register values

```
  Bit     Data
  0 - Half-Ref voltage HI or LO
  1 - VREF voltage HI or LO
  2 - +5V voltage HI or LO
  3 - +5VA voltage HI or LO
  4 - Relay Drive voltage HI or LO
  5 - Reserved
  6 - Reserved
  7 - Phase A Input Test failed
  8 - Phase B Input Test failed
  9 - Phase C Input Test failed
  10 - Temperature too high
  11 - 15 - Reserved
```

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_CLEARCOUNT

```
ViStatus naIPM_query_CLEARCOUNT (ViSession instrumentHandle,
                                 ViChar _VI_FAR response[]);
```

Purpose

This function commands Record Count value to be set to zero. Data records are not erased but indext to records is cleared.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "OK" if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_GET

```
ViStatus naIPM_query_GET (ViSession instrumentHandle,
                          ViInt32 firstRecordNumber,
                          ViInt32 lastRecordNumber, ViPInt32 byteCount,
                          ViChar _VI_FAR dataRecordStructure[]);
```

Purpose

This function queries the contents of a group data records from First Record (n) to Last Record (m). The number of bytes comprising the data as well as the data itself are returned.

If the First Record number and the Last Record number are the same, only 1 record is returned.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

firstRecordNumber

Variable Type   ViInt32

This input parameter specifies the record number of the first record to be read.

lastRecordNumber

Variable Type   ViInt32

This input parameter specifies the record number of the last record to be read.

byteCount

Variable Type   ViInt32 (passed by reference)

This parameter returns the number of bytes contained in the set of data records.

dataRecordStructure

Variable Type   ViChar[]

This parameter contains the character (byte) array containing the requested data records. Depending on the number of records requested, this may be as large as 4M bytes.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_HIGH

```
    ViStatus naIPM_query_HIGH (ViSession instrumentHandle,
                               ViChar _VI_FAR highestRecord[]);
```

Purpose

This function queries the record number of the highest data record already stored.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type	ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

highestRecord

Variable Type	ViChar[]

This parameter returns the number of the highest data record already stored.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naINT.h" and "visa.h" headers.)

---

naIPM_query_init_options

```
    ViStatus naIPM_query_init_options (ViPInt16 baudRateIndex,
                                       ViPInt16 multiDropAddress,
                                       ViPBoolean multiDropMode);
```

Purpose

This function queries additional parameters needed for initialization of baud rate and multi-drop addressability.

The naIPM_init function will use these parameters.

Parameter List

baudRateIndex

Variable Type   ViInt16 (passed by reference)

This parameter will set the baud rate index value used to establish baud rated during initialization.

Valid values:

INDEX	BAUD RATE
0	115,200 baud, -3.55%
1	not defined, default to 115,200
2	not defined, default to 115,200
3	not defined, default to 115,200
4	2400 baud, -.08%
5	4800 baud, +.16%
6	9600 baud, +.16%
7	14400 baud, +.64%
8	19200 baud, +.16%
9	28800 baud, -.79%
10	38400 baud, +.16%
11	57600 baud, -.79%
12	not defined, default to 115,200
13	not defined, default to 115,200
14	not defined, default to 115,200
15	115,200 baud, -3.55%

Default value: none

multiDropAddress

Variable Type   ViInt16 (passed by reference)

This parameter will set the optional multi-drop address value used during initialization.

Valid values: 0 - 255

multiDropMode

Variable Type   ViBoolean (passed by reference)

This parameter is used by initialization to enable or disable optional multi-drop addressability.

Valid Range:
VI_OFF (0) - Multi-drop addressability disabled
VI_ON (1) - Multi-drop addressability enabled

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_LASTTRIP

```
ViStatus naIPM_query_LASTTRIP (ViSession instrumentHandle,
                               ViPInt32 byteCount,
                               ViChar _VI_FAR lastTripRecord[]);
```

Purpose

This function queries the data record that was stored for the last trip event of the power monitor.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

byteCount

Variable Type   ViInt32 (passed by reference)

This parameter returns the number of byte read.

lastTripRecord

Variable Type   ViChar[]

This parameter returns the character (byte) array containing data record stored for last trip event.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_MANUF

```
ViStatus naIPM_query_MANUF (ViSession instrumentHandle,
                            ViChar _VI_FAR manuracturer[]);
```

Purpose

This function queries instrument manufacurer identification.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

manuracturer

Variable Type   ViChar[]

This parameter accepts the returned manufacturer name.

Response:
"North Atlantic Industries, naii.com"
or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_MEASURE

```
ViStatus naIPM_query_MEASURE (ViSession instrumentHandle,
                              ViPInt32 byteCount,
                              ViChar _VI_FAR measurementValues[]);
```

Purpose

This function queries data structure containing all measured parameter values.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

byteCount

Variable Type   ViInt32 (passed by reference)

This parameter accepts the number of bytes in the status message. Nominal value is 34.

measurementValues

Variable Type   ViChar[]

This parameter accepts the returned character (byte) array representing current instrument parameter measurement data structure.

Response:

Byte	Data
1-2	Frequency (16-bit, lsb = 0.1Hz)
3-4	Reserved
5-6	Temperature (16-bit signed, lsb = 0.1 deg. C)
7-8	Phase A voltage rms (16-bit, lsb = 0.1 volt)
9-10	Phase B voltage rms (16-bit, lsb = 0.1 volt)
11-12	Phase C voltage rms (16-bit, lsb = 0.1 volt)
13-14	Phase A voltage peak (16-bit, lsb = 0.1 volt)
15-16	Phase B voltage peak (16-bit, lsb = 0.1 volt)
17-18	Phase C voltage peak (16-bit, lsb = 0.1 volt)
19-20	Phase A DC component (16-bit, lsb = 1 mV DC)
21-22	Phase B DC component (16-bit, lsb = 1 mV DC)

23-24	Phase C DC component (16-bit, lsb = 1 mV DC)
25-26	Phase A Phase Angle (16-bit, lsb = 0.1 deg) rel to A
27-28	Phase B Phase Angle (16-bit, lsb = 0.1 deg) rel to A
29-30	Phase C Phase Angle (16-bit, lsb = 0.1 deg) rel to A
31	Phase A THD (8-bit, lsb = 0.1%)
32	Phase B THD (8-bit, lsb = 0.1%)
33	Phase C THD (8-bit, lsb = 0.1%)
34	PowerOK (8-bit, 1 = power good, 0 = no good)

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_NEXT

```
ViStatus naIPM_query_NEXT (ViSession instrumentHandle,
                           ViChar _VI_FAR nextRecord[]);
```

Purpose

This function queries the number of the next data record to be stored. This value will always be 1 more that the value returned by naIPM_query_HIGH function.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type	ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

nextRecord

Variable Type	ViChar[]

This parameter returns the number of the next data record to be stored.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_OPSTATE

```
ViStatus naIPM_query_OPSTATE (ViSession instrumentHandle,
                              ViChar _VI_FAR operatingState[]);
```

Purpose

This function queries operating state of the instrument.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

operatingState

Variable Type   ViChar[]

This parameter accepts the returned operating state value.

Response:
"0" - OFF state
"1" - reserved
"2" - RESET state
"3" - TRIPPED state
"4" - FAILED state
or an error message

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_PARTNO

```
ViStatus naIPM_query_PARTNO (ViSession instrumentHandle,
                             ViChar _VI_FAR partNumber[]);
```

Purpose

This function queries exact product part number.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

partNumber

Variable Type   ViChar[]

This parameter accepts the returned instrument part number.

Response:
similar to "iPM33A0000(360-440)(110-130)(005-005)00"
or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_POWER

```
    ViStatus naIPM_query_POWER (ViSession instrumentHandle,
                                ViChar _VI_FAR powerStatus[]);
```

Purpose

This function queries status of power that is being monitored by the instrument.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

powerStatus

Variable Type   ViChar[]

This parameter accepts the returned power status value.

Response:
"1" - Power OK
"0" - Power no good
or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_RECCOUNT

```
    ViStatus naIPM_query_RECCOUNT (ViSession instrumentHandle,
                                   ViChar _VI_FAR recordCount[]);
```

Purpose

This function queries number of data log records available.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

recordCount

Variable Type   ViChar[]

This parameter contains number of records in data log.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_RECORD

```
ViStatus naIPM_query_RECORD (ViSession instrumentHandle,
                             ViPInt32 byteCount,
                             ViChar _VI_FAR currentRecord[]);
```

Purpose

This function queries the most current data record that was stored for the power monitor.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

byteCount

Variable Type   ViInt32 (passed by reference)

This parameter returns the number of byte read.

currentRecord

Variable Type   ViChar[]

This parameter returns the character (byte) array containing the most current data record stored.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_RECSIZE

```
    ViStatus naIPM_query_RECSIZE (ViSession instrumentHandle,
                                  ViChar _VI_FAR recordSize[]);
```

Purpose

This function queries the size of one data log record, in bytes.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

recordSize

Variable Type   ViChar[]

This parameter contains a value representing the size (in bytes) of one data record.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_SERNO

```
    ViStatus naIPM_query_SERNO (ViSession instrumentHandle,
                                ViChar _VI_FAR serialNumber[]);
```

Purpose

This function queries instrument specific serial number.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

serialNumber

Variable Type ViChar[]

This parameter accepts the returned instrument serial number.

Response:
similar to "123456-78"
or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_STATUS

```
ViStatus naIPM_query_STATUS (ViSession instrumentHandle,
                             unsigned long *byteCount,
                             ViChar _VI_FAR monitorStatus[]);
```

Purpose

This function queries monitor overall status data structure.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

byteCount

Variable Type unsigned long (passed by reference)

This parameter accepts the number of bytes in the status message. Nominal value is 12.

monitorStatus

Variable Type ViChar[]

This parameter accepts the returned character (byte) array representing current instrument status data structure.

Response:

Byte	Data
1	OpState (8-bit)
2	PowerOK (8-bit)
3-6	Trip Flag (32-bit)
7-10	CautionFlag (32-bit)
11-12	bitStatus (16-bit)

```
Data Details:
Opstate
  0 = OFF, 1 = (reserved), 2 = RESET, 3 = TRIPPED, 4 = FAILED
Power OK
  0 = power no good, 1 = power good
Trip Flag and Caution Flag
  Bit     Data
  0 - Improper phase Sequence
  1 - High Steady-State Frequency
  2 - Low Steady-State Frequency
  3 - Phase A High Steady-State Voltage
  4 - Phase A Low Steady-State Voltage
  5 - Phase B High Steady-State Voltage
  6 - Phase B Low Steady-State Voltage
  7 - Phase C High Steady-State Voltage
  8 - Phase C Low Steady-State Voltage
  9 - Phase A High DC component
  10 - Phase B High DC component
  11 - Phase C High DC component
  12 - Phase A High THD
  13 - Phase B High THD
  14 - Phase C High THD
  15 - Phase A High Peak Voltage
  16 - Phase B High Peak Voltage
  17 - Phase C High Peak Voltage
  18 - High Transient Frequency
  19 - Low Transient Frequency
  20 - Phase A High Transient Voltage
  21 - Phase A Low Transient Voltage
  22 - Phase B High Transient Voltage
  23 - Phase B Low Transient Voltage
  24 - Phase C High Transient Voltage
  25 - Phase C Low Transient Voltage
  26 - 31 - Reserved

Built-In-Test Status
  Bit     Data
  0 - Half-Ref voltage HI or LO
  1 - VREF voltage HI or LO
  2 - +5V voltage HI or LO
  3 - +5VA voltage HI or LO
  4 - Relay Drive voltage HI or LO
  5 - Reserved
  6 - Reserved
  7 - Phase A Input Test failed
  8 - Phase B Input Test failed
  9 - Phase C Input Test failed
  10 - Temperature too high
  11 - 15 - Reserved
```

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_VER

```
    ViStatus naIPM_query_VER (ViSession instrumentHandle,
                              ViChar _VI_FAR firmwareVersion[]);
```

Purpose

This function queries firmware version of the instrument.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

firmwareVersion

Variable Type   ViChar[]

This parameter accepts the returned instrument firmware version.

Response:
similar to "VER 002 2009-07-14"
or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_reset

```
ViStatus naIPM_reset (ViSession instrumentHandle);
```

Purpose

This function resets the instrument to a known state with normal monitor functions enabled.

Notes:

(1) If this instrument does not support a Reset, this function should return the Warning Code 0x3FFC0102 - VI_WARN_NSUP_RESET

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_revision_query

```
ViStatus naIPM_revision_query (ViSession instrumentHandle,
                               ViChar _VI_FAR instrumentDriverRevision[],
                               ViChar _VI_FAR firmwareRevision[]);
```

Purpose

This function returns the revision numbers of the instrument driver and instrument firmware.
Notes:

(1) If this instrument does not support a Firmware Revision Query, this function should return the Warning Code 0x3FFC0105 - VI_WARN_NSUP_REV_QUERY.

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

```
Variable Type       ViSession
```

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

instrumentDriverRevision

```
Variable Type       ViChar[]
```

This control returns the Instrument Driver Software Revision.

Notes:

(1) The array must contain at least 256 elements ViChar[256].

firmwareRevision

```
Variable Type       ViChar[]
```

This control returns the Instrument Firmware Revision.

Notes:

(1) The array must contain at least 256 elements ViChar[256].

Return Value

The return value will be 0 if the function terminates normally.
A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_self_test

```
ViStatus naIPM_self_test (ViSession instrumentHandle,
                          ViPInt16 selfTestResult,
                                 ViChar _VI_FAR selfTestMessage[]);
```

Purpose

This function runs the instrument's self test routine and returns the test result(s).

Notes:

(1) If this instrument does not support a Self-Test, this function should return the Warning Code 0x3FFC0103 - VI_WARN_NSUP_SELF_TEST

This command will be ignored if OPTIONAL multi-drop addressability is enabled and no matching OPTIONAL ADR command has been sent.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

selfTestResult

Variable Type   ViInt16 (passed by reference)

This control contains the value returned from the instrument self test. Zero means success. For any other code, see the device's operator's manual.

selfTestMessage

Variable Type   ViChar[]

This control contains the string returned from the self test. See the device's operation manual for an explanation of the string's contents.

Notes:

(1) The array must contain at least 256 elements ViChar[256].

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_set_init_options

```
    ViStatus naIPM_set_init_options (ViInt16 baudRateIndex, ViInt16 address,
                                     ViBoolean multiDropMode);
```

Purpose

This function provides a means to set additional parameters needed for initialization of baud rate and multi-drop addressability.

These parameters may also be set by naIPM_find_first function.

The naIPM_init function will use these parameters.

Parameter List

baudRateIndex

Variable Type   ViInt16

This parameter will set the baud rate index value used to establish baud rated during initialization.

Valid values:

INDEX	BAUD RATE
0	115,200 baud, -3.55%
1	not defined, default to 115,200
2	not defined, default to 115,200
3	not defined, default to 115,200
4	2400 baud, -.08%
5	4800 baud, +.16%
6	9600 baud, +.16%
7	14400 baud, +.64%
8	19200 baud, +.16%
9	28800 baud, -.79%
10	38400 baud, +.16%
11	57600 baud, -.79%
12	not defined, default to 115,200
13	not defined, default to 115,200
14	not defined, default to 115,200
15	115,200 baud, -3.55%

Default value: none

address

Variable Type   ViInt16

This parameter will set the optional multi-drop address value used during initialization.

Valid values: 0 - 255

multiDropMode

Variable Type   ViBoolean

This parameter is used by initialization to enable or disable optional multi-drop addressability.

Valid Range:
VI_OFF (0) - Multi-drop addressability disabled
VI_ON (1) - Multi-drop addressability enabled

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

APPENDIX B. LabWindows/CVI Driver Documentation File for naIPMdiag.dll (configuration commands)

North Atlantic Industries IPM Configure

Introduction:

This instrument driver provides programming support for North Atlantic Industries IPM Configure. It contains functions for opening, configuring, taking measurements from, and closing the instrument.

Assumptions:

To successfully use this module, the following conditions must be met:

For GPIB instrument drivers:

- the instrument is connected to the GPIB.
- the GPIB address supplied to the initialize function must match the GPIB address of the instrument.

For VXI instrument drivers:

- the instrument is installed in the VXI mainframe and you are using one of the following controller options:
  - Embedded controller
  - MXI
  - MXI2
  - GPIB-VXI
- the logical address supplied to the initialize function must match the logical address of the instrument.

For RS-232 instrument drivers:

- the instrument is connected to the RS-232 interface.
- the COM port, baud rate, parity, and timeout supplied to the initialize function must match the settings of the instrument.

Error and Status Information:

Each function in this instrument driver returns a status code that either indicates success or describes an error or warning condition. Your program should examine the status code from each call to an instrument driver function to determine if an error occurred. The general meaning of the status code is as follows:

Value	Meaning
0	Success
Positive Values	Warnings
Negative Values	Errors

The description of each instrument driver function lists possible error codes and their meanings

How To Use This Document:

Use this document as a programming reference manual. It describes each function in the

North Atlantic Industries IPM Configure

instrument. The functions appear in alphabetical order, with a description of the function and its C syntax, a description of each parameter, and a list of possible error codes.

Function Tree Layout:

```
   Class/Panel Name:                        Function Name:

      Configuration Command Functions
         SETCONFIG (Config Mode) Command    naIPM_cmd_SETCONFIG
         SETPARTNO (Part Number) Command    naIPM_cmd_SETPARTNO
         SETSERNO (Serial Num) Command      naIPM_cmd_SETSERNO
         CALDELTA (Calibrate) Command       naIPM_cmd_CALDELTA
         ERASEFLASH (clear log) Command     naIPM_cmd_ERASEFLASH
         SETBAUD ( Baud Rate) Command       naIPM_cmd_SETBAUD
         SETADDRESS ( Address) Command      naIPM_cmd_SETADDRESS
         SETMULTIDROP (Adr Mode) Command    naIPM_cmd_SETMULTIDROP
         YES (Execute  Setup) Command       naIPM_cmd_YES
      Configuration Query Functions
         CONFIG (Config Mode) Command       naIPM_query_CONFIG
         BAUD (Baud Rate) Command           naIPM_query_BAUD
         ADDRESS (Address) Command          naIPM_query_ADDRESS
         MULTIDROP (Adr Mode) Command       naIPM_query_MULTIDROP
```

North Atlantic Industries IPM Configure

This instrument module provides programming support for the <Instrument Name>. The module is divided into the following functions:

Functions/Classes:

(1) Initialize:
This function initializes the instrument and sets it to a default configuration.

(2) Application Functions: (Class)
This class contains high-level, test and measurement routines. These examples call other instrument driver functions to configure, start, and read from the instrument.

(3) Configuration Functions: (Class)
This class of functions configures the instrument by setting acquisition and system configuration parameters.

(4) Action/Status Functions: (Class)
This class of functions begins or terminates an acquisition. It also provides functions which allow the user to determine the current status of the instrument.

(5) Data Functions: (Class)
This class of functions transfers data to or from the instrument.

(6) Utility Functions: (Class)
This class of functions provides lower level functions to communicate with the instrument, and change instrument parameters.

(7) Close:
This function takes the instrument offline.

The following functions are in alphabetical order.

---

naIPM_cmd_CALDELTA

```
    ViStatus naIPM_cmd_CALDELTA (ViSession instrumentHandle,
                                 ViChar _VI_FAR response[]);
```

Purpose

This function initiates calibration of 3-wire delta configuration. Prior to execution of this command, monitor input should be set to 115.0 Vrms line-to-neutral, with no neutral connection to the monitor.

Configuration mode must be enabled to use this command.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "OK" if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_ERASEFLASH

```
ViStatus naIPM_cmd_ERASEFLASH (ViSession instrumentHandle,
                               ViChar _VI_FAR response[]);
```

Purpose

This function initiates erasure of internal flash memory that is used to store data log values.

Configuration mode must be enabled to use this command.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "OK" if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally.

A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_SETADDRESS

```
ViStatus naIPM_cmd_SETADDRESS (ViSession instrumentHandle,
                               ViInt16 address,
                               ViChar _VI_FAR response[]);
```

Purpose

This function sets the RS485 multi-drop device address. When multi-drop addressability is enabled, this allows multiple iPM units on the same RS485 bus to be addressed individually.

Valid response will be "ARE YOU SURE?"

Use the naIPM_cmd_YES function to confirm and execute the address setup. The new address will be activated the next time the unit is powered up.

Configuration mode must be enabled to use this command.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

address

Variable Type   ViInt16

This input parameter is an integer representing the RS485 multi-drop address.

Range: 0 to 255

Default value: none

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "ARE YOUR SURE?" if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_SETBAUD

```
ViStatus naIPM_cmd_SETBAUD (ViSession instrumentHandle,
                                ViInt16 baudRateIndex,
                                ViChar _VI_FAR response[]);
```

Purpose

This function sets the RS485 baud rate by selecting one of 16 programmable index values.

Valid response will be "ARE YOU SURE?"

Use the naIPM_cmd_YES function to confirm and execute the baud rate setup. The new baud rate will be activated the next time the unit is powered up.

Configuration mode must be enabled to use this command.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

baudRateIndex

Variable Type   ViInt16

This input parameter is an integer representing the index of the programmable baud rate value.

INDEX	BAUD RATE
0	115,200 baud, -3.55%
1	not defined, default to 115,200
2	not defined, default to 115,200
3	not defined, default to 115,200
4	2400 baud, -.08%
5	4800 baud, +.16%
6	9600 baud, +.16%
7	14400 baud, +.64%
8	19200 baud, +.16%
9	28800 baud, -.79%
10	38400 baud, +.16%
11	57600 baud, -.79%
12	not defined, default to 115,200
13	not defined, default to 115,200
14	not defined, default to 115,200
15	115,200 baud, -3.55%

Default value: none

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "ARE YOUR SURE?" if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_SETCONFIG

```
ViStatus naIPM_cmd_SETCONFIG (ViSession instrumentHandle,
                              ViBoolean configurationState);
```

Purpose

This function sets the unit to configuration (setup) mode. This mode is requiered in order to use certain setup commands.

There is no response from the unit for this command.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

configurationState

Variable Type   ViBoolean

This input parameter is a boolean value representing the desired state of the configuration mode.

Value	State
0	OFF
1	ON

Default value: none

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_SETMULTIDROP

```
ViStatus naIPM_cmd_SETMULTIDROP (ViSession instrumentHandle,
                                 ViBoolean multidropMode,
                                 ViChar _VI_FAR response[]);
```

Purpose

This function enables or disables optional RS485 multi-drop addressability.

When multi-drop addressability is enabled, this allows multiple iPM units on the same RS485 bus to be addressed individually.

Valid response will be "ARE YOU SURE?"

Use the naIPM_cmd_YES function to confirm and execute the address setup. The new address will be activated the next time the unit is powered up.

Configuration mode must be enabled to use this command.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

multidropMode

Variable Type   ViBoolean

This input parameter is boolean value representing the RS485 multi-drop mode state.

0 = Disabled
1 = Enabled

Default value: none

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "ARE YOUR SURE?" if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_SETPARTNO

```
ViStatus naIPM_cmd_SETPARTNO (ViSession instrumentHandle,
                              ViChar _VI_FAR partNumber[],
                              ViChar _VI_FAR response[]);
```

Purpose

This function configures the factory-set part number of the power monitor. This part number is critical to the operation of the product because is sets all operational parameters including trip points and options enabled.

Configuration mode must be enabled to use this command.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

partNumber

Variable Type   ViChar[]

This input parameter contains part number to be programmed into the instrument. String must be exactly 39 characters.

Typical Value:
"iPM33A0000(360-440)(110-130)(005-005)00"

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "OK" if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_cmd_SETSERNO

```
ViStatus naIPM_cmd_SETSERNO (ViSession instrumentHandle,
                             ViChar _VI_FAR serialNumber[],
                             ViChar _VI_FAR response[]);
```

Purpose

This function configures the factory-set serial number of the power monitor.

Configuration mode must be enabled to use this command.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

serialNumber

Variable Type   ViChar[]

This input parameter contains serial number to be programmed into the instrument. String may be up to 32 characters.

Typical Value:
"30001210-12"

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "OK" if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally.
A negative value indicates either an instrument error or a
communications error. The error message function can be used to
convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

naIPM_cmd_YES

```
ViStatus naIPM_cmd_YES (ViSession instrumentHandle,
                        ViChar _VI_FAR response[]);
```

Purpose

This function is used to confirm and execute setup commands that return
the "ARE YOU SURE?" response.

Configuration mode must be enabled to use this command.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize
function to select the desired instrument driver session.

Default Value: None

response

Variable Type   ViChar[]

This parameter returns the insrument response. Response will be "OK"
if command completes successfully, or an error message.

Return Value

The return value will be 0 if the function terminates normally.
A negative value indicates either an instrument error or a
communications error. The error message function can be used to
convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

naIPM_query_ADDRESS

```
ViStatus naIPM_query_ADDRESS (ViSession instrumentHandle,
                              ViChar _VI_FAR address[]);
```

Purpose

This function queries the optional RS485 multi-drop address.

Configuration mode must be enabled to use this command.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

address

Variable Type ViChar[]

This parameter accepts the returned optional RS485 multi-drop address value.

Expected values: 0 to 255 or an error message.

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_BAUD

```
ViStatus naIPM_query_BAUD (ViSession instrumentHandle,
                           ViChar _VI_FAR baudRateIndex[]);
```

Purpose

This function queries the baud rate setup index.

Configuration mode must be enabled to use this command.

Parameter List

instrumentHandle

Variable Type ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

baudRateIndex

Variable Type ViChar[]

This parameter accepts the returned baud rate index value.

INDEX	BAUD RATE
0	115,200 baud, -3.55%
1	not defined, default to 115,200
2	not defined, default to 115,200
3	not defined, default to 115,200
4	2400 baud, -.08%
5	4800 baud, +.16%
6	9600 baud, +.16%
7	14400 baud, +.64%
8	19200 baud, +.16%
9	28800 baud, -.79%
10	38400 baud, +.16%
11	57600 baud, -.79%
12	not defined, default to 115,200
13	not defined, default to 115,200
14	not defined, default to 115,200

```
        15     115,200  baud, -3.55%

        or an error message.
```

Return Value

        The return value will be 0 if the function terminates normally.
        A negative value indicates either an instrument error or a
        communications error. The error message function can be used to
        convert the result value into an error message string.

        (See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_CONFIG

```
    ViStatus naIPM_query_CONFIG (ViSession instrumentHandle,
                                 ViChar _VI_FAR configurationState[]);
```

Purpose

This function queries the state of configuration (setup) mode.

```
    Response will be:
    ------------------------
    "ON"  Configuration mode is active
    "OFF" Configuration mode is inactive
    error message
```

Parameter List

instrumentHandle

```
        Variable Type       ViSession
```

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

configurationState

```
        Variable Type       ViChar[]
```

This parameter accepts the returned configuration mode state.

```
        Response:
        "ON"  = configuration mode enabled
        "OFF" = configuration mode disabled

        or an error message.
```

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)

---

naIPM_query_MULTIDROP

```
    ViStatus naIPM_query_MULTIDROP (ViSession instrumentHandle,
ViChar _VI_FAR address[]);
```

Purpose

This function queries the optional RS485 multi-drop mode.

Configuration mode must be enabled to use this command.

Parameter List

instrumentHandle

Variable Type   ViSession

This control accepts the Instrument Handle returned by the Initialize function to select the desired instrument driver session.

Default Value: None

address

Variable Type   ViChar[]

This parameter accepts the returned optional RS485 multi-drop mode.

Expected values:
"ON" = multi-drop addressability enabled
"OFF" = multi-drop addressability disabled

or an error message

Return Value

The return value will be 0 if the function terminates normally. A negative value indicates either an instrument error or a communications error. The error message function can be used to convert the result value into an error message string.

(See error codes in "naIPM.h" and "visa.h" headers.)