

PNI Application Note:

Using RealTerm to Communicate with PNI Modules

Introduction

PNI's compassing and AHRS modules, including the Prime, TCM, SeaTrax, and Trax, communicate using PNI's binary protocol. It is possible to communicate with these modules using this protocol and a terminal emulation program.

In this application note we provide specific examples for operating PNI's TRAX AHRS with the RealTerm terminal emulation program. PNI's compassing modules would work the same, with the exception being the "SetDataComponents" and "GetData" examples where kHeadingStatus is only applicable for TRAX. The user should reference the specific device's user manual for the appropriate command set.

RealTerm is open source software, and can be downloaded at: <http://realterm.sourceforge.net/>. Other terminal emulation programs should work similarly.

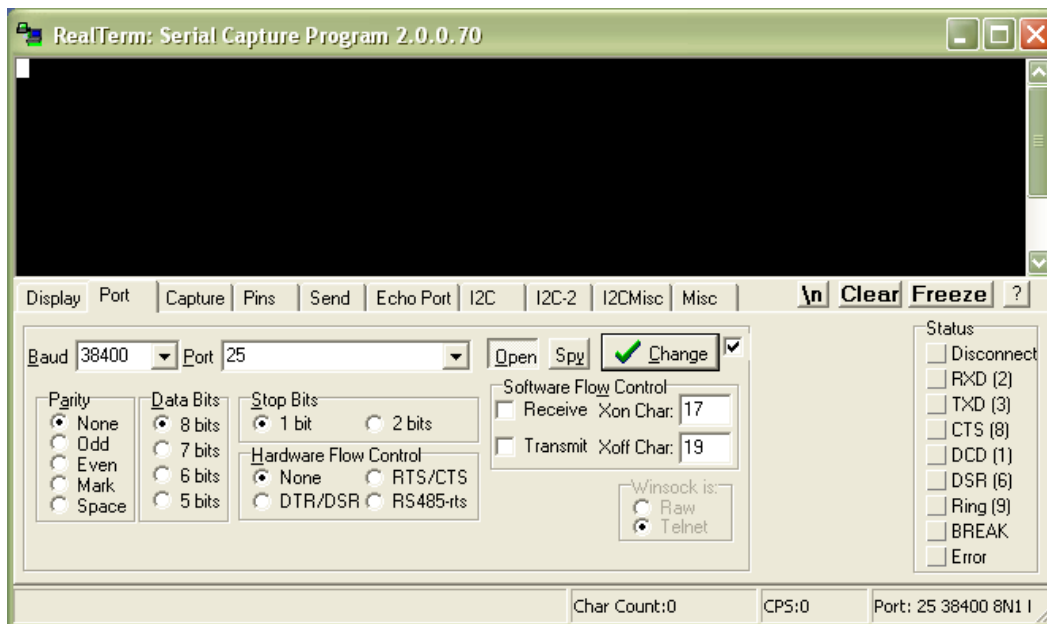
Set-Up

- Install RealTerm on a Windows PC.
- Connect the module to the PC's RS232 serial port or a USB virtual communication port (VCP). The picture below shows the enclosed TRAX connected to a PC's USB port using the TRAX's ODU-to-USB Cable.

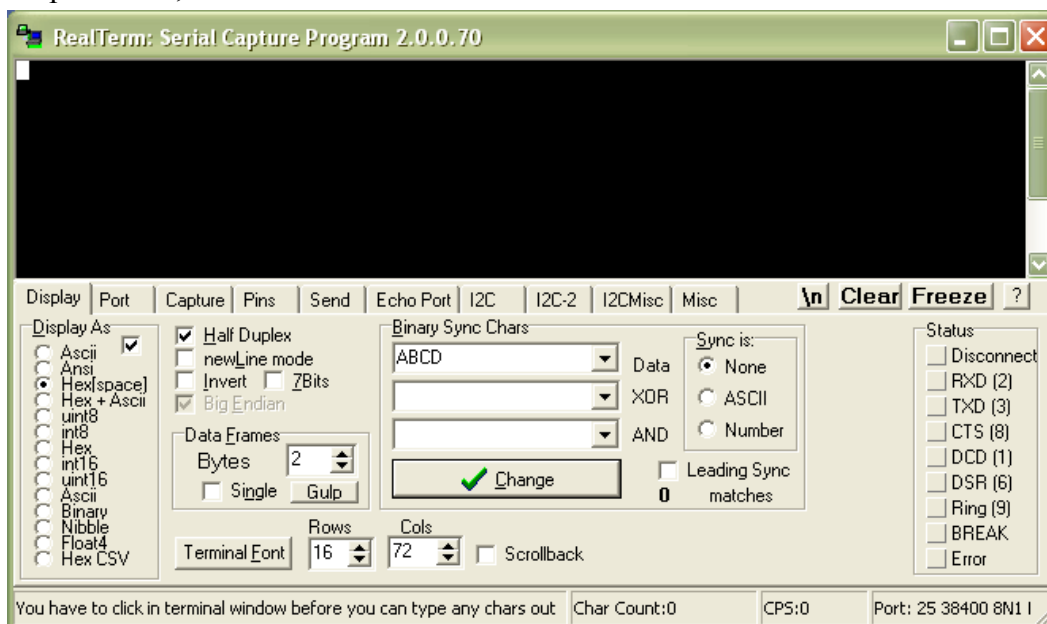


Establish Communication with the PNI Module

- Open RealTerm and click the “Port” tab. On this tab, set “Baud” to “38400” and set “Port” to whichever port is assigned to the PNI module. In the case below, VCP port 23 is assigned to TRAX. Click <Change> to implement these changes.



- Select the “Display” tab. Choose “Display As” “Hex[space]”, and check the “Half Duplex” box, as shown below.



Sending Commands to the PNI Module

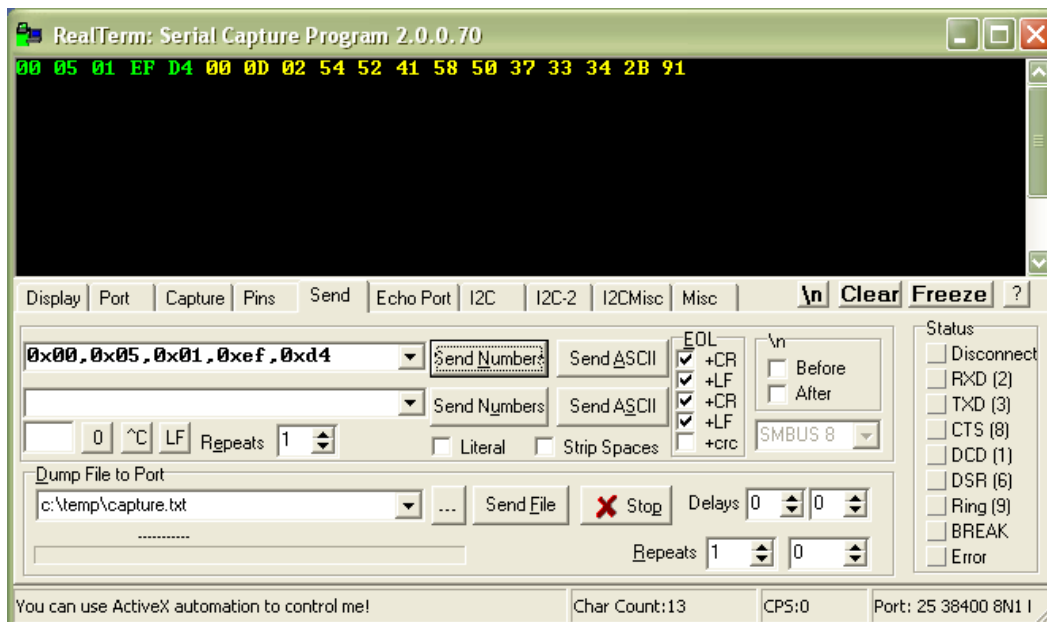
Select the “Send” tab to send commands to the PNI module. Below is a sample command set to first query the model and revision of the device, then to set the data components to be output, and then to retrieve data. The first sample command is explained in byte-by-byte detail, while subsequent samples do not have this level of detail. Note that sent commands are in green text while responses are in yellow text. For the sake of clarity, the screen is cleared prior to each sent command, but normally this would not be done.

Read the model and version of the device.

- Enter “0x00,0x05,0x01,0xef,0xd4” in the field next to <Send Numbers>. The first two bytes represent the byte count, the third byte represents the frame ID, and the last two bytes are the CRC checksum. For this case the byte count is 5 (0x0005), the frame ID is 0x01, representing kGetModInfo per the user manual, and the CRC checksum is 0xefd4. The CRC checksum should comply with the XModem standard and can be determined with an online calculator, such as that found at:

<http://www.lammertbies.nl/comm/info/crc-calculation.html?crc=000515&method=hex>.

- Click <Send Numbers>. The screen shot below depicts what the screen looks like after selecting <Send Numbers>. Green text in the window represents the command that was sent, while yellow text represents the response.

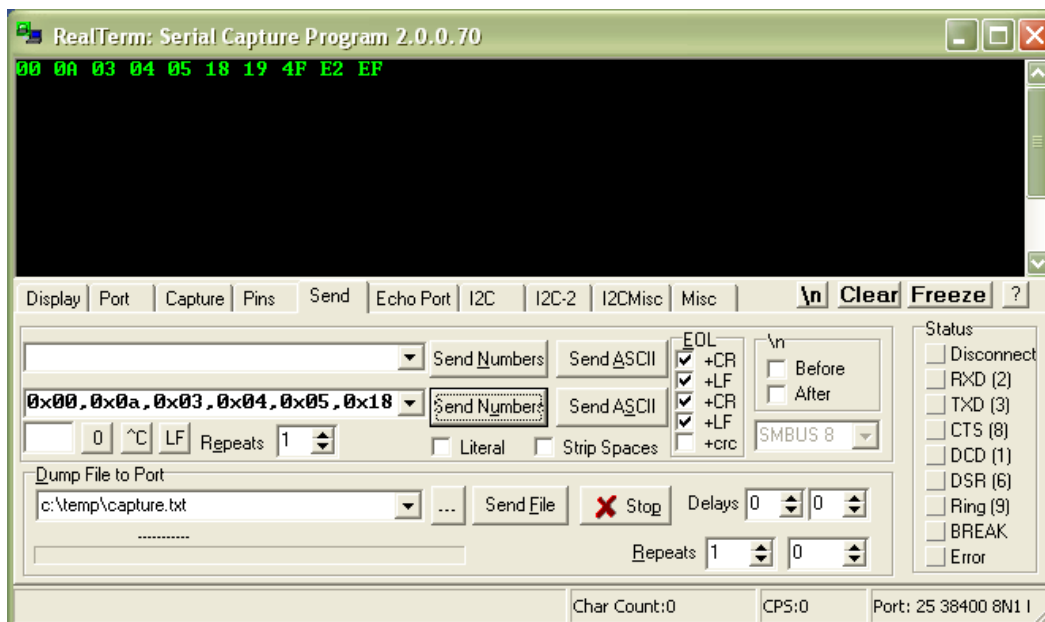


- Interpret the response of “00 0D 02 54 52 41 58 50 37 33 34 2B 91”. “00 0D” represents the byte count of the response, which is 13 bytes. (You can count the yellow bytes to confirm this.) “02” represents the frame ID, which is kGetModInfoResp per the user manual. The next 8 bytes (13 bytes minus 5 fixed bytes) is the payload. The last two bytes, “2B 91”, are the CRC checksum.
- Decipher the response payload, which is “54 52 41 58 50 37 33 34”. Per the user manual, the first 4 bytes represent the device model and the last 4 bytes represent the revision. Converting the payload to ASCII yields “TRAX” for the device model and “P734” for the revision. A HEX-to-ASCII converter can be found at:

<http://www.dolcevie.com/js/converter.html>.

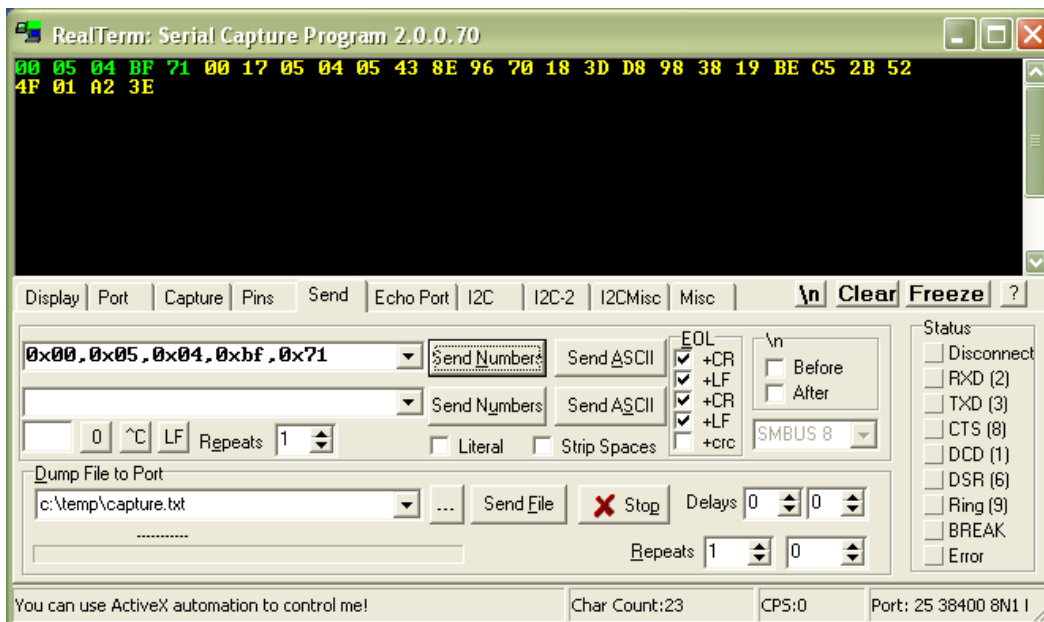
Set the data components to be output.

- Enter “0x00,0x0a,0x03,0x04,0x05,0x18,0x19,0x4f,0xe2,0xef” in the field next to <Send Numbers>. This writes ksetdatacomponents (frame ID 0x03), as discussed in the user manual. The payload indicates 0x04 components are to be set, with the components being kHeading (0x05), kPitch (0x18), kRoll (0x19), and kHeadingStatus (0x4f). The CRC checksum is 0xf0df.
- Click <Send Numbers>. The screen shot below depicts what the screen looks like after selecting <Send Numbers>. Green text in the window represents the command that was sent, and there is no response.



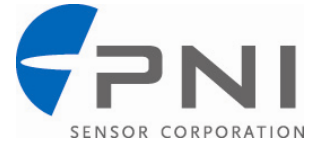
Get Data.

- Enter “0x00,0x05,0x04,0xbf,0x71” in the field next to <Send Numbers>. This writes kGetDataResponse (frame ID 0x04). There is no payload
- Click <Send Numbers>. The screen shot below depicts what the screen looks like after selecting <Send Numbers>. Green text in the window represents the command that was sent, while yellow text represents the response.



- Interpret the response, which is “00 17 05 04 05 43 8E 96 70 18 3D D8 98 38 19 BE C5 2B 52 4F 01 A2 3E”. The byte count is 0x0017, which represents 23 bytes. The frame ID is 0x05, representing kGetDataResp. The number of data components is 4. The first component is 0x05, kHeading, and its float value is 0x438E9670 representing 285.2°. Similarly, kPitch (0x18) is float 0x3DD89838 representing 0.1°, kRoll (0x19) is float 0xBEC52B52 representing -0.4°, and kHeadingStatus (0x4F) is 0x01 representing a heading uncertainty of <2°. A HEX-to-FLOAT converter can be found at: <http://babbage.cs.qc.cuny.edu/IEEE-754.old/32bit.html>.

Note: The default kSetDataComponents is for kHeading, kPitch, and kRoll (HPR). If something different than HPR is desired, but HPR is returned, then something is likely wrong with the kSetDataComponents packet. Frequently errors are in the byte count or the CRC checksum.



©2014 PNI Sensor Corporation. All Rights Reserved. Revised March 2014.

For the most recent version visit our website at www.pnicorp.com or contact PNI.

Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under copyright laws.

PNI Sensor Corporation
2331 Circadian Way
Santa Rosa, CA 95407, USA
Tel: (707) 566-2260
Fax: (707) 566-2261

PNI Sensor Corporation (PNI) reserves the right to correct, change, or modify this document and the products associated with this document solely at PNI's discretion.

No license, express or implied, to any intellectual property rights is granted under this document. If any part of this document refers to third party products or services it shall not be deemed a license grant by PNI for the use of such third party products or services.

PNI assumes no liability relating to the choice, selection or use of PNI products or services described herein.

All PNI products are sold pursuant to PNI's terms and conditions of sale.

PNI Sensor Corporation, PNI, and the PNI logo are registered trademarks of PNI.