instruNet®
Users Manual

July 9, 2003
Manual Version 2.0

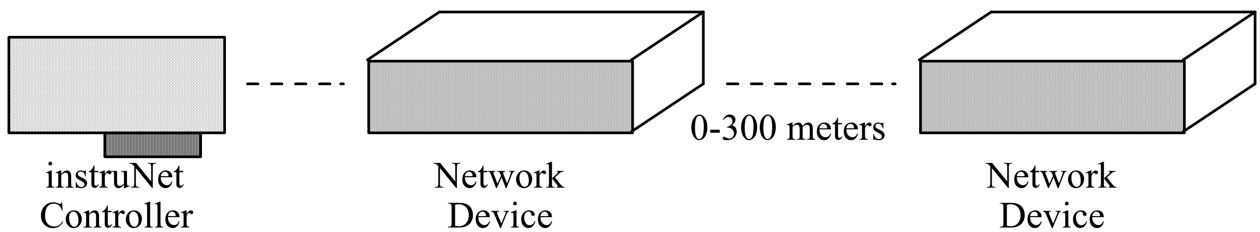
Table of Contents

F9 to update TOC

Chapter 1, Installation

This chapter explains how to install instruNet hardware and software onto your computer, and how to verify that your instruNet System is operating properly. With new systems it is recommend that you first do the Chapter 1 Installation, and then proceed to the Chapter 2 Tutorial.

instruNet Family Overview



Overview

instruNet is a hardware and software product family that enables one to interface computers such as the Apple Macintosh and Windows computers to common laboratory and factory equipment for purposes of data acquisition and control. instruNet utilizes a high speed network approach that is both low cost and flexible for providing voltage inputs, voltage outputs, digital inputs, digital outputs, and timer I/O to the computer.

Controllers

Each instruNet Network is controlled by a Network Controller board that installs into a computer. A different controller board is used with each common bus interface (e.g. PCI, PC-Card), yet they are all very similar internally. Each Controller is an independent computer in itself that utilizes a powerful 32-bit microprocessor and onboard RAM to control all aspects of data acquisition along its network. One can install as many Controllers as desired, space permitting, since each controller operates independently. Each network supports up to 32 Network Devices. Each Device is a small box (e.g. 10cm x 12cm x 25cm) that is connected in a daisy-chain configuration to form a chain of Devices. Each network can be up to 300 meters long. All networks are anchored with an instruNet Terminator at the far end, and an instruNet Controller at the near end. This makes instruNet a cost effective method for designing large scale, high speed, multi-channel data acquisition systems. The following table lists the instruNet Controllers described in this manual.

Model	Controller	Bus	Size	Computer Required
200	PCI Controller	PCI	7" x 4"	Windows 95/98/Me/Nt/2k/Xp Computer with PCI Rev ≥2.0 compliant, 32-bit, +5V slot; or Macintosh OS >= 8 (must boot OS 9 if on OS >= 10.2 Mac) with PCI Slot.
230	PC-Card Controller	PC-Card	2" x 3"	Windows 95/98/Me/2k/Xp (not NT) Computer or Macintosh OS >= 8 (must boot OS 9 if on OS >= 10.2 Mac) PPC/G3/G4/> Powerbook computer (i.e. Models 1400, 2400, 3400, 5300, >=G3). Requires Type II PCMCIA compliant ≥ v2.1 (or ≥ PC-Card 95) PC-Card slot.

Table 1.1 instruNet Controllers described in this Manual

Network Devices

Network Devices typically provide voltage input channels, voltage output channels, digital inputs and digital outputs. The following Network Devices are described in this manual:

	Voltage Inputs			Voltage Outputs			
Model	# of Channels	Range	Absolute Accuracy	# of Channels	Write Accuracy	Read-back Accuracy	Digital I/O
100	16ch w/ screw terminal access	+/- 5V +/-.6V +/-80mV +/-8mV	+/-1500µV +/-150µV +/-45µV +/-30µV	8ch with 4mA/1KpF Drive Capability	+/- 40 mV	+/- 3mV	8 Bidirectional I/O Bits
100B	Same as #iNet-100, yet with 16 additional BNC connectors for 16se voltage inputs.						
100HC	Same as #iNet-100, yet with voltage outputs that have 15mA/.01uF drive capability.						

Table 1.2 instruNet Network Devices described in this Manual

instruNet Software

instruNet includes software to interrogate, test, configure, and do I/O with all network channels. This includes an application program called "instruNet World"; drivers; interfaces to C, and Visual BASIC. instruNet World and the instruNet Driver can configure all I/O channels, store your settings, view digitized data in real time, stream data to disk, and scroll through your waveform post-acquisition. instruNet software runs on both a PC and a Macintosh OS >= 8 (must boot OS 9 if on OS >= 10.2 Mac).

Free Updates

Free software and manual updates are available on the web at:

www.instrunet.com

Computer Requirements

The following table summarizes the computer required to run instruNet:

Model	Controller	Computer Required	OS Required	RAM Required	HD Required	Slot Required
200	PCI Controller	IBM PC or Compatible ≥ 80486	Windows 95/98/Me Nt/2k/Xp	≥ 4MB	≥ 6MB free	PCI Slot, ≥6.8", Rev ≥2.0 compliant, 32-bit, +5V
230	PC-Card Controller	IBM PC or Compatible ≥ 80486	Windows 95/98/Me/2k/ Xp (not NT)	≥ 4MB	≥ 6MB free	Type II PC-Card Slot with ≥ v2.1 PCMCIA compliant card services
230	PC-Card Controller	Macintosh PPC/G3/G4/> Powerbook	OS >= 8 (must boot OS 9 if on OS >= 10.2 Mac)	≥ 4MB	≥ 6MB free	Type II PC-Card Slot with ≥ v2.1 PCMCIA compliant card services

Table 1.3 Computer Requirements for instruNet Controllers

Constructing Your Network

instruNet hardware is 100% plug and play for all computers. instruNet does not use dip switches, DMA, low memory, interrupts, and I/O addresses. All you need to do is plug the Controller board into your computer, connect your network devices, slap a terminator onto the end of your network and run the instruNet World software. The instruNet driver automatically determines the physical locations of all installed Controllers and Network Devices.

Please keep in mind the following when designing and constructing your network:

1. **Install as many controllers as desired**
   The number of available slots determines the number of controllers (i.e. networks) that can be installed, and simultaneously run, on one computer. The software numbers each controller in the order that they are found in the computer ("netNum" ranges from 1 to # of Controllers). Each controller manages it's own network of devices. In most cases, only one controller is necessary. The advantage of multiple controllers is that each is its own real-time machine, and more controllers can do more things simultaneously.

2. **Install up to 16 Devices on each Network**
   One can attach, in daisy-chain configuration, up to 16 Network Devices to each instruNet Controller. Each Network Device has two DB-25 connectors, one for network input (male), and another for network output (female). To connect a chain of Network Devices, one must connect each input connector to each output connector via a DB-25 Male/Female cable. The Controller is attached to the first device in the chain, and an instruNet Terminator is attached to the far

end of the chain. Due to the male/female polarization, the network cannot be installed incorrectly with instruNet Male-Female cables.

3. **Each Controller includes Timer I/O Channels**
   Each Controller (except iNet-230 PC-Card) provides 10 Timer I/O channels. Each channel can be programmed as a digital input, digital output (0V/4V TTL compatible), clock output, or period measurement input.

4. **Each Controller includes one Terminator**
   One instruNet Terminator must be installed at the end of each network chain. This terminator mates with the output connector of the last device. instruNet Controllers include an instruNet Terminator, therefore they do not need to be purchased separately. *Caution:* Do not use a SCSI Terminator in place of an instruNet Terminator -- they are different.

5. **Each Network Device includes one cable**
   Each instruNet Network Device is shipped with one 10foot DB-25 Cable Male/Female cable for purposes of configuring your network.

6. **You can purchasing your own cables**
   If you want a specific cable length, you can purchase your own DB-25 male to DB-25 female, shielded, wired point-to-point (i.e. pin X to pin X) cables. We recommend 24 gauge wire for > 4 meters; however 28 gauge is fine with ≤4m. Twisted pairs are recommend for >4 meters with the following wires twisted: 1 & 14, 2 & 15, 3 & 16, 4 & 17, 5 & 18, 6 & 19, 7 & 20, 8 & 21, 9 & 22, 10 & 23, 11 & 24, 12 & 25 (these are physically next to each other in the connector). Also, it is recommended that the drain wire (which is attached to the shield) be connected to the housing and pin #1 (GND) on both connectors. For more information on cables, please refer to Application Note #22, Third Party instruNet Cable Suppliers; and Application Note #39, Use Low Capacitance Cable to Maximize your instruNet Network Speed.

   A supplier of high quality twisted pair, low capacitance, double-shielded cable without connectors is Belden, Inc. The Belden Cable Part #8112 Low Capacitance RS-485/RS-232 cable is available in 100ft, 500ft, and 1000ft lengths with 12.5 Pairs and a copper braided shield. This cable features 24 gage wire, 41pF/meter between pairs, 72pF/meter between a wire and the shield, and 78ohms/killometer wire resistance

7. **Minimum Base System**
   One Controller and one Network Device is all you need to purchase to digitize waveforms, save them to disk, and view them.

8. **Maximum Sample Rate**
   As the physical length of the network increases, the maximum aggregate data acquisition sample rate decreases from 166Ks/sec maximum with a short network (e.g. 5 meters) to 4.15Ks/second aggregate with a long network (e.g. 300 meters).

This maximum aggregate rate applies to a batch of input channels. For example one instruNet network can support 4 voltage input channels at a maximum rate of 41.5Ksamples/sec for each channel (i.e. 166Ks/sec throughput). The same network would allow 8 channels of voltage input to be acquired at 20.075Ks/sec per channel. The maximum aggregate rate can be increased by installing additional instruNet networks and controllers. For example, two controllers could support 332Ks/sec aggregate throughput if run simultaneously.

When the instruNet powers up, it empirically tests (i.e. it test the cable impedance) of the network to determine its maximum throughput rate (i.e. 4.15K/sec to 166Ks/sec). The maximum rate is decreased by: additional network devices, longer aggregate network cable length, non-twisted pair cables, and thinner cable wire (e.g. 28 gauge instead of 24 gauge).

For more information on sample rates, please refer to Application Note #58 and #117 (URL's www.instrunet.com#58, www.instrunet.com#117).

9. **Turn power OFF when cabling**
   Always turn the computer and powered Network Devices Off before adjusting network cables.

10. **Large networks require external power supplies**
    The instruNet network cable provides power from the computer to the external Network Devices. As the number of devices increases (more current drawn), and the cable lengths increases (more voltage drops), it becomes increasingly necessary to add an external power supply for the Network Devices. We recommend adding an external power supply if your cable is > 50 meters, or for every 4 Network Devices after your 3rd Device. In other words, only add an external power supply if you have more than 3 network devices, or if your network is longer than 50 meters.

Software Installation

To install instruNet SOFTWARE on a Windows 95/98/Me/Nt/2k/Xp/> Computer:

1) Before installing instruNet Software >= v2.0 (which includes iW+) from CD or from Internet URL www.instrunet.com/d: Turn OFF anti-virus (AV) software, Disable any Anti-Virus "Automatically Launch on Startup" options, Restart your computer, login as ADMINISTRATOR if on Windows Nt/2k/Xp/>, Check that AV software is Off, and Make sure that other application programs are Off. AV must be OFF after Restart since Installer may Restart several times. This is Crucial, especially with older AV software that conflicts w/ modern Windows XP installers.

2) Install instruNet software Before installing physical i2x0 Controller card by inserting instruNet CD into your computer (which automatically runs "Setup.exe"), or by downloading Installer from "www.instrunet.com/d" (which downloads "instrunet_web_setup.cab", opens it via WinZip, and then runs the "Setup.exe" inside). If it asks you if you want to install the .Net Framework, say Yes if running under Windows >=98 and <= 2K (not 95, not XP); and you want to run programs VB Instrument, VB Scope or Direct To Excel; and you don't mind waiting another 2 to 30min for the installer to run. Installation might take up to 1 hour, especially with older computers that require more files. If you become concerned your computer has frozen, please check your hard disk light -- if it is blinking, you are Ok. When the instruNet installation is complete, an "instruNet World" Icon appears on your desktop and an alert notifies you of the success. At this time, it is safe to turn ON anti-virus software. For information on installing instruNet hardware & drivers, please continue with the below instructions.

To enable instruNet World+ (iW+) SOFTWARE on a Windows 98/ME/Nt/2k/Xp/> Computer:

For information on how to enable instrunet World+ (iW+) software, please see the "instruNet World+ Software License Installation" discussion at the end of this chapter, or refer to your "instruNet World+ (iW+) License x For Serial # y" sheet, included with iW+.

To install instruNet i200 PCI or i230 Pcmcia HARDWARE on a Windows 95/98/Me Computer:

1) Make sure you are on a Windows 95/98/Me computer, Install the instruNet software, as noted above, Turn computer POWER off, Install the physical instruNet PCI or PCMCIA card, Attach the instruNet network devices, Attach the instruNet terminator, Tighten the thumbscrews, and Turn the computer on.
2 It may ask you for a PCI or PCMCIA Card Driver on boot-up. Navigate via the Browse/Other Locations button to the .inf/.vxd files on your computer. For PCI, go to "Program Files \ instruNet \ internal \ PCI or PCMCIA Win95 Driver \". Press Next to move through the installation process. It might say "Please insert disk labeled 'Program files \ instruNet \ internal \ Pcmcia or Pci Win95 Driver \' directory". At this point, you must navigate to "Progra~1 \ instru~1 \ internal \ Pcmcia~1 \ iNt95Pcm.vxd" for Pcmcia or "Progra~1 \ instru~1 \ internal \ PciWin~1 \ iNet95.vxd" for PCI. You might need to type "iNt95Pcm.vxd" for Pcmcia or "iNet95.vxd" for PCI into the file name field, while "Progra~1 \ instru~1 \ internal \ Pcmcia~1 or PciWin~1 \" is selected in the Directory area. When done, the card should be listed under "Data Acquisition Cards" in the System "Device Manager". Reboot your computer and then proceed to "To Verify Your System...", below.

To install instruNet i200 PCI or i230 Pcmcia HARDWARE on a Windows Nt/2k/Xp/> Computer:

Make sure you are on a Windows Nt/2k/Xp computer (or later), Install the instruNet software, as noted above, Turn computer POWER off, Install the physical instruNet PCI or PCMCIA card, Attach the instruNet network devices, Attach the instruNet terminator, Tighten the thumbscrews, and Turn the computer on. If it asks for a .sys/.inf driver, navigate to "\ Program Files \ instruNet \ Internal \ PCI or PCMCIA Win 2k-Xp Driver \" for PCI "inet.sys, inetpci.inf" or PCMCIA "inetpcm.sys, inetpcm.inf". The i230 does not run under Win NT. Proceed to "To Verify Your System...", below.

To install instruNet i200 PCI or i230 Pcmcia HARDWARE/SOFTWARE on a Macintosh computer:

1) Make sure you are on an OS >= 8 Macintosh (must boot OS 9 if on OS >= 10.2). The i200 requires a PowerPC/G3/G4/> Mac w/ 1 free PCI slot. The i230 requires a Powerbook 1400/5300/2400/3400/G3/G4/> Mac with at least one free Type II pcmcia slot.
2) Copy the Macintosh instruNet software to your computer (from the included instruNet CD in the "Macintosh" directory, or from www.instruNet.com/d). If the files are compressed (i.e. the file has an ".sea" or ".sit" suffix), uncompress them with Stuffit Expander >= 5.1.13 from Aladdin Systems. Copy into your Extensions folder (inside your System folder) "instruNet Driver (ppc)". If working with SuperScope II, use the version of "instruNet Driver (ppc)" included with instruNet.
3) Turn computer POWER off, install the instruNet PCI/Pcmcia card, attach instruNet network devices, attach instruNet terminator, tighten thumbscrews, turn cpu on, and proceed to "To Verify Your System...", below. On a Powerbook 1400/5300 with an i230, it might say "please install driver". Please ignore this alert, since the driver is inside instruNet World.

To Quickly Verify That Your System Is Working Properly:

To test instruNet, run "Start > Programs > instruNet > instruNet World" if on Windows or run "instruNet World Mac" if on Macintosh, press the TEST tab at the bottom of the window, press the SEARCH button and make sure your hardware is listed (e.g. i200 card, i100 box). If the list is not correct, please see Note #69, "Troubleshooting" (www.instrunet.com\trouble). Otherwise, if the list is correct, please see the 1 page Note #209 "Getting Started Quickly with Windows" (www.instrunet.com\start) document and then proceed to the instruNet User's Manual Ch 2.

For details on verifying proper operation, please see the "To Verify …" discussion later in this chapter.

To copy instruNet software From www.instrunet.com To non-networked computers:

Surf to "www.instrunet.com/s" via Internet Explorer. Drag file "instrunet_web_setup.cab" from your browser window to your hard disk -- to download this file to your computer. Open this .cab file with decompression software such as WinZip (a popular program for opening .zip files) and copy it's contents (i.e. a "setup.exe" file) to the computer(s) on which you want to install instruNet Software (i.e. via burning a CD, or via a local area network), and Install per the above instructions.

Hardware Installation

To install an instruNet network, please:

1. Read the previous "Constructing your Network" discussion. Before installing hardware, please install software per the pervious instructions.
2. Turn OFF your computer.
3. Turn OFF all powered devices connected to your network.
4. Touch bare metal on your computer to discharge personal static electricity.
5. Remove the cover from your computer to gain access to the card slots, as needed.
6. Remove the small I/O fence cover from the back of your computer, as needed.
7. If you require access to an available Controller Digital/Timer I/O Channel, run a 34-wire ribbon cable from the Controller's 34pin header connector, out of the computer, any way you can (e.g. through another slot opening), to the breakout of your choice (e.g. a screw terminal block), as illustrated below:

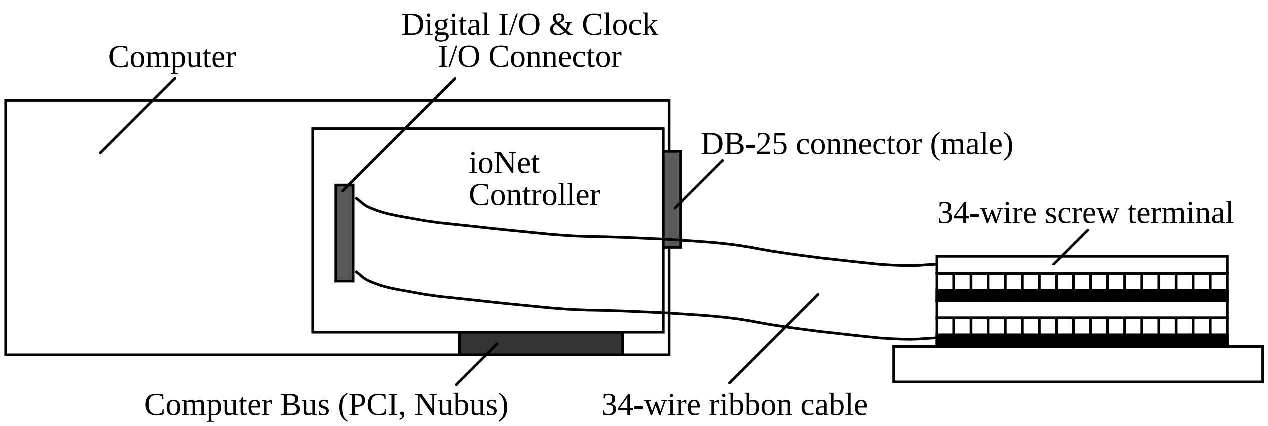


Figure 1.4, Installing an instruNet Controller into a Computer

8. Install the instruNet controller(s) into the computer's expansion slot(s). If working with a PCI Card, make sure the Controller connector is well seated and inspect this connection with a strong light to make sure the printed circuit board fingers are aligned with their mating connector pins.
9. Bolt the board metal I/O fence to the computer, as needed (some computers do this). Please skip this step if tightening this bolt causes the card to not seat well in its connector.
10. Put the cover back onto the computer, as needed.
11. Attach the external instruNet Network Devices in a daisy chain configuration, as illustrated below.

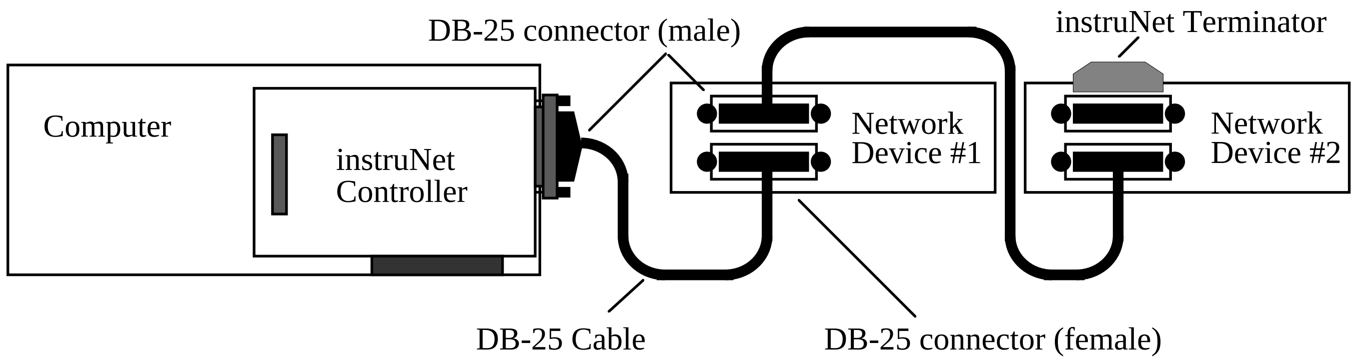


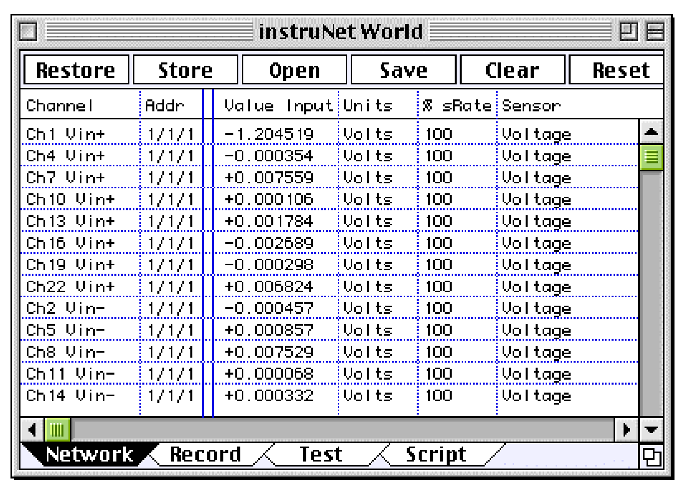
Figure 1.5, instruNet Network

12. Tighten all DB-25 thumbscrews until lightly snug.
13. Install an instruNet Terminator on the end of each network.
14. It is recommended that the user attach the instruNet instrumentation ground (i.e. instruNet box and instruNet GND screw terminals) directly to Earth ground with a 16gauge wire from the left-most GND screw on the instruNet box, to the closest Earth ground (e.g. screw next to power socket). This will reduce the chance of RFI coupling into the instruNet ground, and is required if the user wants to meet EC and FCC RFI guidelines.
15. If working with an iNet-230 PC-Card, attach an external power supply (e.g. iNet-311.x, 312.x, 322.x) to the PC-Card 5pin DIN connector. This supplies power to the external network devices (i.e. not the PC-Card itself).
16. Turn the computer power ON and then Turn ON all powered devices attached to the instruNet network.

Verifying That Your System Is Working Properly

Verify that your hardware and software is working in 5 easy steps:

1. Run the "`instruNet World`" application program ("`instruNet World Mac`" on Macintosh, and "`instruNet World Win32.exe`" on a Microsoft Windows computer), in the instruNet directory. A window will open, similar to what is pictured to the right. If necessary, you might need to click on the Network tab at the bottom of the window to select the Network page. If this window opens, then you know your instruNet driver



file is installed and working correctly. The list of channels shown in the window's table will vary, depending on what instruNet hardware is connected. If the instruNet window does not appear, then check the Software Installation section at the beginning of this chapter to make sure that the software has been installed correctly. If it appears that the software is installed correctly but not functioning properly then see Application Note #69 Troubleshooting (www.instrunet.com\trouble) and instruNet User's Manual *Appendix I Trouble Shooting*.

2. Press the **Test** tab at the bottom of the window to select the Test page, and then press the **Search** button at the top of the window. A report will print that lists the controllers and network devices that are currently installed on your computer. For example, the report below shows one Controller, and one Model 100 Network Device.

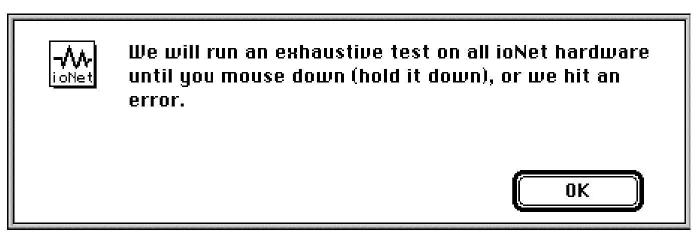
```
instruNet HARDWARE SEARCH RESULTS:  Date & Time: 10/2/1997, 12:12:41
Net Dev Mod  Device
---------------------------------------
 0   0   0    Mac OS Ver 7.5.3
 0   0   1      instruNet Driver Version 1.24
 0   0   1      instruNet World+ license Not Installed (costs extra)
 1   0   1         PCI Controller #iNet-200 (slot #13, 4000KBD, 6us, 94%)
 1   1   1           Device #iNet-100 (SN37532, Cal 9/28/97, Rev4, 30.99C, ...
```

If this does not match what you believe is installed on your network, then check your hardware installation (cables, power, etc). Also, it is possible that the instruNet Driver is older than the devices on your network, which means you need the latest driver that recognizes these new devices. The instruNet Driver is always listed as the 1st installed item.

3. Press the **Test** button at the top of the window to test your controllers and devices, and to report any problems if found. The duration of the test can vary from 1 to 120 seconds depending on the size of the networks, and the speed of the computer. If no problems are found, a report similar to the one below is printed in the window:

```
INSTRUNET QUICK TEST RESULTS:
Date & Time: 10/12/1995, 12:53:23
We ran 0.510029 million tests and did NOT hit 1 error.
```

4. Press the **Big Test** button to run an exhaustive test. An alert similar to the one on the right will appear to communicate that the computer will test all instruNet hardware until there is an error, or until you press the mouse button down. It will test your network(s) all night long if you let it. Press OK to begin the Big Test. Wait 20 seconds or longer. If an error occurs while testing, an alert will appear and the error message will also be printed in the window. Refer to *Appendix II instruNet Error Codes* for more information on error codes, and *Appendix I Trouble Shooting* for tips on de-



bugging if necessary. If no error alerts appear, and you want to stop the test, press the mouse button down and hold it down until the an alert appears announcing the end of the test. Click OK to exit this alert. The test results are printed in the window, in a format similar to what is shown below:

```
We ran 2.170272 million tests and did NOT hit 1 error.
```

Big Test is identical to **Test**, except it runs for a longer period of time and is useful at finding intermittent problems that only occur once every minute, hour, or day. Big Test can be run overnight for extensive testing of all hardware.

5. You are done! Your instruNet hardware and software is installed correctly and running beautifully. Please see the 1 page Note #209 "Getting Started Quickly with Windows" document (www.instrunet.com\start) and then proceed to the instruNet User's Manual Chapter 2, Tutorial to learn more.

instruNet World+ Software License Installation

Overview

instruNet World (not +) software is available free of charge and included on the instruNet CD with each i2x0 controller card (and at "www.instrunet.com/d"). The "PLUS" version (referred to as "instruNet World+" or "iW+"), with more features, is available at an additional charge under the following product numbers:

- #iNet-350-*SerialNumberOfController*, instruNet World+ for pre-existing i2x0 controller card. One must specify the serial number of that controller on their Purchase Order, since license codes are keyed to controller serial numbers.
- #iNet-200P, instruNet PCI card and instruNet World+
- #iNet-230P, instruNet PCMCIA card and instruNet World+

Since the PLUS software is included inside the NOT plus software, one only needs to register the iW+ license code, described below, to enable the PLUS software after installing the NOT plus software (i.e. after running the instruNet CD).

Licenses are issued for each individual controller cards and are keyed to the controller's serial number. Subsequently, the PLUS features are only enabled in the following cases:

- A valid PLUS license code is installed, yet no controller card is installed.
- A valid PLUS license code is installed, along with it's corresponding controller card.
- instruNet software is running in Demo mode and simulating a controller card.

iW+ Installation

To install an iW+ license on your computer and enable its powerful features on Windows 98/Me/Nt/2k/Xp/> computers (iW+ is not supported on Macintosh):

1. In order to proceed, you must have purchased instruNet World+ (iW+) software and have received an iW+ license code for your controller card. To learn more about PLUS, please select "instruNet World+ Manual" in the "Help" menu within instruNet World, or see Internet URL www.instrunet.com/plus.

2. Install instruNet software Version >= 2.0 onto your computer via the instruNet CD, or from web URL "www.instrunet.com/d".

3. Launch the instruNet World software by double-clicking on its icon, or by selecting START > Programs > instruNet > instruNet World.exe.

4. Select "Install instruNet World Script License…" in the Script menu.

5. Enter your license code (i.e. in the zz-ssssss-yyyyy format) into the license field (note that "0" refers to number 0).

6. Press Ok, and a confirmation alert will appear. To view the status of your license, press the Test tab at the bottom of the window and then press the Search button. From this point forward, whenever you run instruNet World on this computer (i.e. START > Programs > instruNet > instruNet World.exe), you will be running iW+. Since the license code is stored in a file within the System directory, installing newer versions of instruNet software will not require repeating this procedure. To learn more about PLUS, please select "instruNet World+ Manual" in the "Help" menu within instruNet World, or see Internet URL www.instrunet.com/plus.

Chapter 2, instruNet Tutorial

This chapter is a step-by-step tutorial that shows the user how to navigate within the world of instruNet. Controlling instruNet hardware can be done manually though the free instruNet World application program, or through the programming languages Visual Basic and C. This chapter deals exclusively with the easy-to-use application program instruNet World application program while Chapter 4 covers programming languages. instruNet World software allows you to set up and probe your network, record waveforms, save them to disk, load them from disk, and view them post acquisition.

This manual focuses on the buttons at the top of the instruNet World pages, yet these functions can *also* be accessed in the menubar. For documentation on the menubar and on instruNet World PLUS "iW+" (a version of instruNet World with more features), please select "instruNet World+ Manual" in the "Help" menu within instruNet World, or see Internet URL www.instrunet.com/plus.

Record Waveforms in 7 Easy Steps

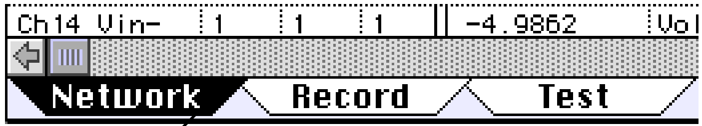
This section explains how to record waveforms in several easy steps.

1. **Install your hardware and software per instructions in Chapter 1**
   If your instruNet World hardware and software is not installed, please install it now, as described in *Chapter 1*.

2. **Run the instruNet World application program.**
   Locate the instruNet world application program within the instruNet folder on your hard disk and then:

   Windows: Double-click on "`instruNet World Win32.exe`",
   or run "Start > Programs > instruNet > instruNet World".
   Macintosh: Double-click on the "`instruNet World Mac`" icon.

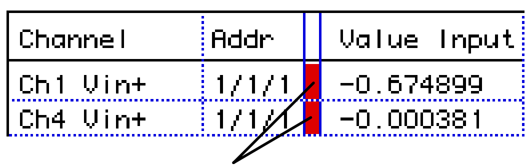
3. **Select the Network Page.**
   instruNet World offers several Pages: Record, Network, and Test. Click on the Network tab at the bottom of the window to select the Network page. The Network tab will inverse black to indicate the Network page is selected.



Click here to select the Network Page

4. **Enable a Channel for digitizing.**
A channel is enabled for digitizing by clicking on the small cell between the addr and Value Input columns within the Network page, as illustrated below. Once enabled, the channel will be digitized when the user presses the Start button on the Record page. To disable a channel, one must click the digitize on/off cell again. This digitize on/off cell is black or red when On, and white when Off. Any number of channels can be selected for digitizing.

Please enable two voltage input channels (e.g. "Ch1 Vin+" and "Ch4 Vin+" on the Model 100) for digitizing, as illustrated below. Voltage input channels are typically labeled ChX Vin+ or ChX Vin-. These work identically when doing single-ended voltage measurement (i.e. read a voltage between an input terminal and ground), and are used as a pair when doing differential voltage measurement (i.e. reading a voltage between 2 input channels). If instruNet voltage input hardware is not installed, you will not be able to digitize. Also, note that the contents of the Network page may vary depending on what is installed on your computer.



To Enable/Disable a Channel for digitizing

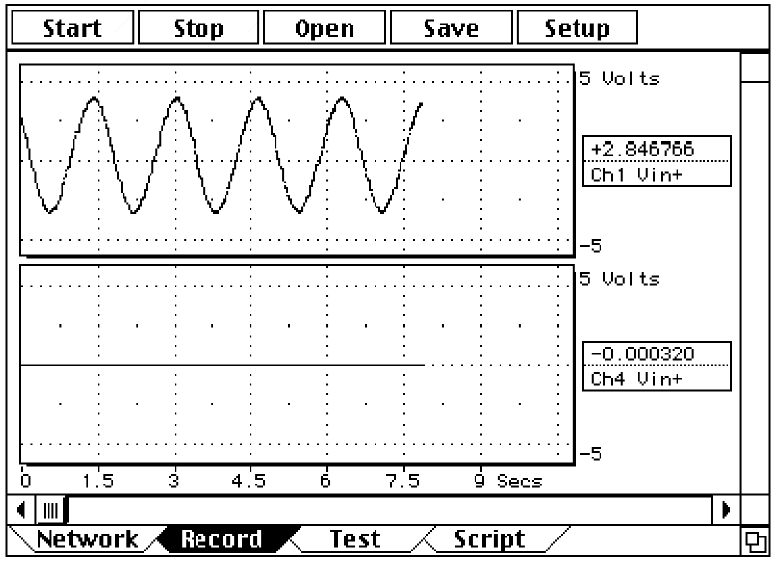
5. **Attach a signal source.**
If possible, attach a signal source to at least one channel's hardware input terminal. For example, one might attach a Function Generator output to the instruNet "Ch1 Vin+" input terminal, and the Function Generator's ground to the instruNet "AGND" terminal. It is not necessary to connect a signal source to do the tutorial, however, the displayed waveforms are more interesting if a signal is applied; otherwise, you get a flat line at 0Volts.

6. **Select the Record Page.**
Select the Record page by clicking on the Record tab at the base of the window, as illustrated below.

Click here to select Record Page

7. **Tell instruNet to start digitizing.**
Click the Start button at the top of the window to tell instruNet to record and display channels that have been enabled for digitizing (e.g. "Ch1 Vin+"). You should see your waveforms move across the screen as they are digitized in real-time, as illustrated below.

Click Start to start recording, Stop to stop recording



8. **Tell instruNet to stop digitizing**
Click the Stop button to stop the digitizing process.

9. **Save your waveforms to disk**
Click the Save button to save your waveforms to disk. When the save dialog appears, type a name and choose a convenient location to save the data. Saving does not specify a file name, but rather a folder name in which all acquired waveforms and a preferences file are saved. For example, if you digitized 2 waves and then clicked Save, 3 files would be stored in your: one named "instruNet.prf" that contains the Field settings, and two files that have the same name as the two channels, that contain the wave data.

10. **Record again**
Click the Start button to start recording again, and then click the Stop button after a few moments.

11. **Load your saved waves from disk**
Click the Open button to load in the previously saved waves from disk. A File open dialog will appear, and it is here that you must select one of your previously saved files (e.g. "instruNet.prf", "Ch1 Vin+" or "Ch4 Vin+"). After your waves are loaded in, they should appear in their displays .

Digitizing Directly To Excel

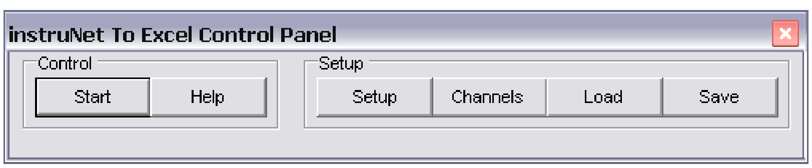
If you want to learn how to digitize in instruNet World, and then save a text file to be loaded by a spreadsheet program, post-acquisition, please refer to Appendix III. If you want to digitize into a spreadsheet, *while* data is being acquired, please consider the Direct To Excel program, described here. Note that Direct To Excel requires that an iW+ license be installed on your computer.

- In order to run Direct To Excel, *All* of the following must be installed on your computer. If this is not the case, please proceed to the next section.

    - Windows Excel version >= 8.0 (i.e. Microsoft Office >= 97)
    - Windows >= 98 (not Windows 95, not Macintosh)
    - iW+ license (see www.instrunet.com/plus for details)
    - Windows .Net Framework (which is included with the instruNet software, and is included automatically on Windows >= XP computers)
    - instruNet software >= 2.0 (7/1/2003), available at www.instrunet.com/d

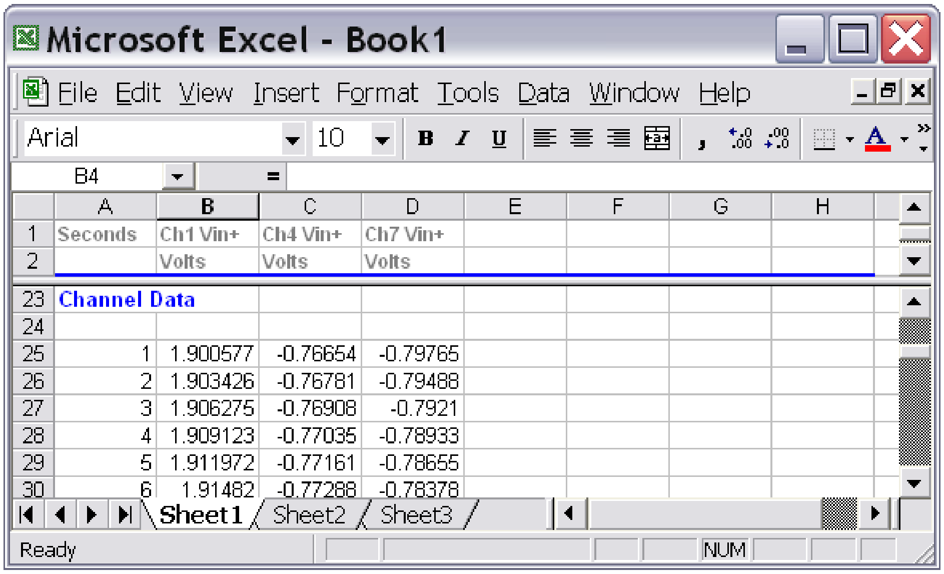
  If you are not sure these requirements have been satisfied, try running the Direct To Excel program, as described below, and if it runs ok, then you are probably ok.

- Exit instruNet World and select under the Windows START menu: Programs / instruNet / Application Software / Direct To Excel / Direct To Excel.exe. This will run a program (shown below) written in Visual Basic; the source code of which is currently installed on your hard disk. You are welcome to modify it.



- Press the Channels button, select 2 channels for digitizing (i.e. click in small rectangle after channel address, it will turn red), and close the instruNet Network window.

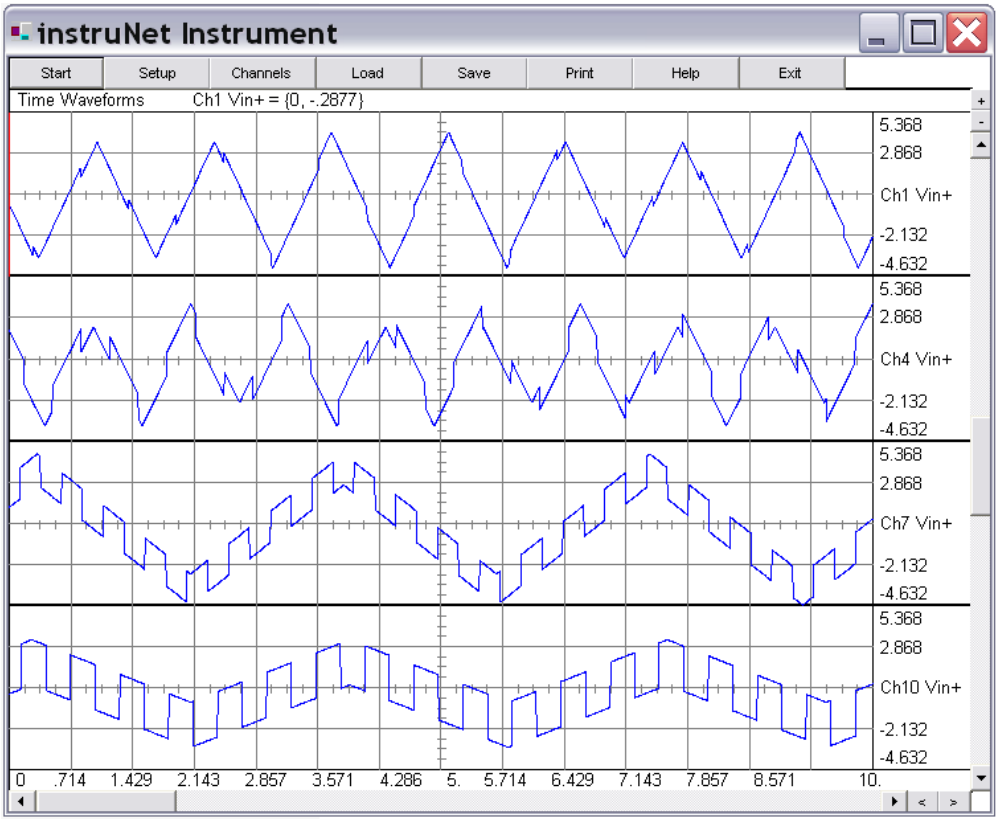
- Press the Start button and watch a new spreadsheet window appear and fill with instruNet data, as shown below. When finished, you can save or delete the Excel spreadsheet file.



- To learn more, please press the Help button.
- Close the Direct To Excel program by clicking in the upper right corner.

Working with VB Instrument

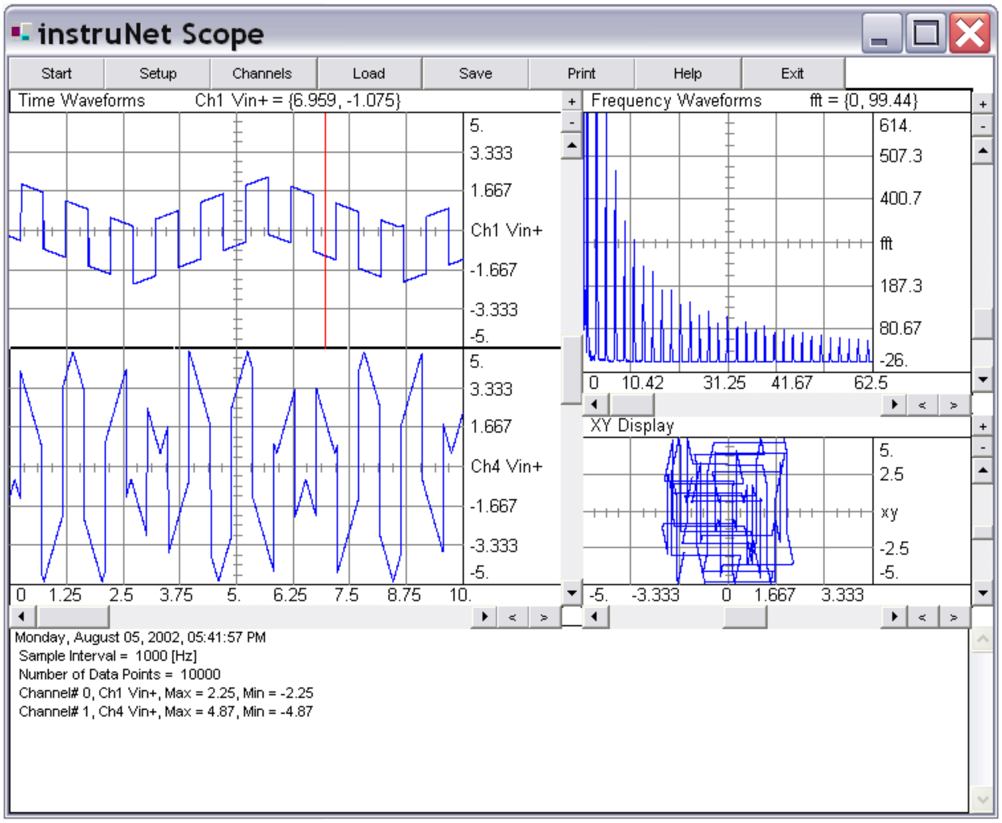
- We will now run a Windows program called "VB Instrument" that implements a strip chart/oscilloscope recorder for 1 to 16 channels. It is similar to instruNet World, yet is written in Visual Basic and the source code is installed on your computer. You are welcome to modify it. If you are on a Macintosh, please proceed to the next section.
- Exit any currently running instruNet software and select under the Windows START menu: Programs / instruNet / Application Software / VB Applications / VB Instrument.exe.
- Press the Channels button, select 4 channels for digitizing (i.e. click in small rectangle after channel address, it will turn red), and close the instruNet Network window.
- Press the Start button and watch the data appear in the window, as shown below.



- To learn more, please press the Help button.
- Close the VB Instrument program by clicking in the upper right corner.

Working with VB Scope

- We will now run a similar Windows program (written in Visual Basic, source code included) called "VB Scope". It implements a 2 channel strip chart recorder / oscilloscope, XY Record, and Spectrum Analyzer. If you are on a Macintosh, please proceed to the next section.
- Exit any currently running instruNet software and select under the Windows START menu: Programs / instruNet / Application Software / VB Applications / VB Scope.exe.
- Press the Start button and watch the data appear in the window, as shown below. Ch1 and Ch2 timewaves are shown in the left-most display, the frequency spectrum of Ch1 is shown in the upper-right display, and an XY plot of Ch1 and Ch4 are shown in the lower-right display.,



- To learn more, please press the Help button.
- Close the VB Scope program by clicking in the upper right corner.

Digitizing Analog Signals into the Computer

The Setup button at the top of the Record page opens a dialog box that effects the manner in which waves are recorded.

- Click the Setup button to open the Setup dialog, as illustrated in Figure 2.1.

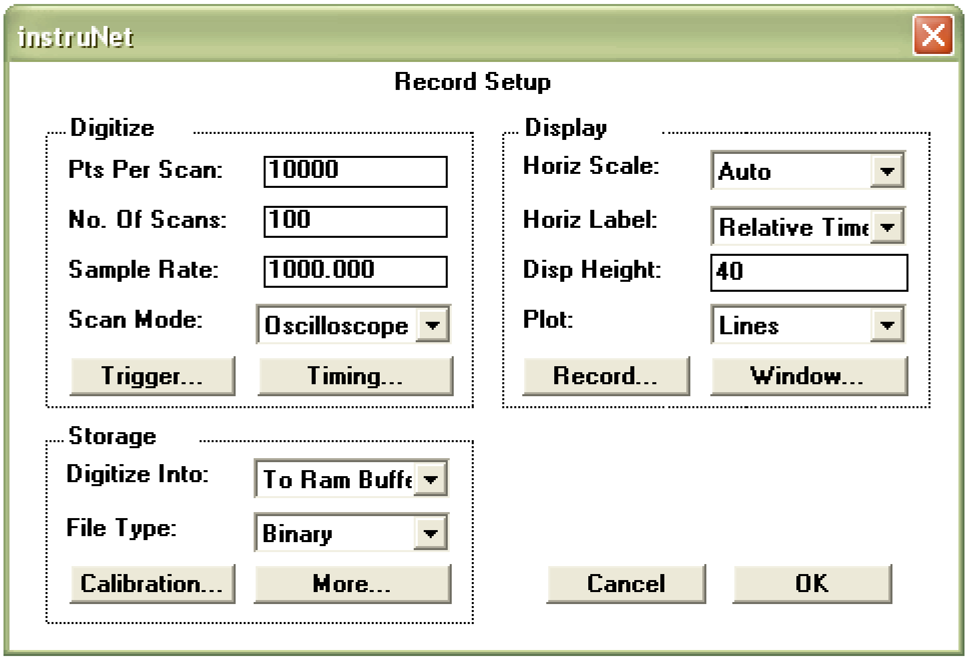


Figure 2.1 The Record Setup Dialog

The Setup dialog is used to set the base sample rate, the number of points to be acquired per Scan, the number of Scans to be acquired and the recording mode (i.e. oscilloscope or strip chart recorder). All instruNet Networks are set up with one base sample rate (i.e. number of points digitized per second) and individual channels can have sample rates less than or equal to the base sample rate. This allows, in effect, each channel to have its own sample rate.

The Sample Rate field sets the base sample rate. The Pts Per Scan field determines the amount of data to be collected in each Scan. The No. of Scans sets the number of Scans to be acquired. The Scan Mode popup has three choices: Strip Chart, Oscilloscope, and Oscillo Queued. Strip Chart is selected for continuous strip chart recorder mode and Oscilloscope or Oscillo Queued are selected for oscilloscope mode. Refer to the *Oscilloscope or Strip Chart* section of *Chapter 5* for a full description of these modes.

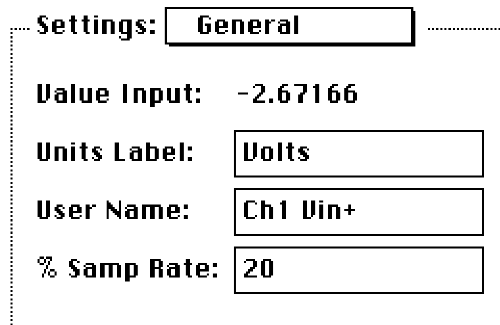
instruNet Networks are self-configuring and on startup determine the maximum rate at which data can be transferred. This rate is displayed after pressing the Timing button, in the Network BPS field, in units of bits per second. 4 million bits per seconds is the fastest, and 100Kbps is the slowest. This rate slows down with networks that have many Devices and long network cables (i.e. >100ft).

- Select Oscilloscope in the Scan Mode popup and set the Pts Per Scan field to 100. 100 points at 1000s/sec will take .1 seconds to acquire. Leave the rest of the Dialog in its default settings, and click OK to return to the Record Page.

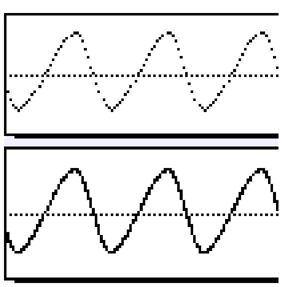
- Click the Start button to begin digitizing.

Notice how 0.1 second long waveforms continuously appear on the screen, in a manner similar to an Oscilloscope. Before, we were in Strip Chart mode where these segments were continuous with respect to each other. We are now in Oscilloscope mode. To learn much more about digitizing, please refer to the Record page discussion in *Chapter 5, instruNet World Reference.*

- Click the Stop button to stop digitizing, and then click on the top display's channel name label at the right edge of the display. The Display dialog will open. Choose General in the Settings popup. Enter the value 20 in the % samp rate field as shown to the right and press OK.



This will cause the top channel to be digitized at 20% of the master sample rate, or 200s/sec. The channel in the lower displays will continue to run at the master sample rate of 1000s/sec.



- Click the Start button to begin digitizing, and then click Stop after a few moments.

  Notice how the wave in the top display contains fewer points, due to its reduced sample rate, as illustrated to the left.

- Click the Setup button at the top of the Record page, and then click the Trigger button to open the Trigger dialog, as shown in Figure 2.2.

Trigger Modes

instruNet World allows triggering from any channel on either a low-to-high or high-to-low transition through a threshold value. The threshold is specified in the Thresh EU field, and the trigger direction is specified in the Slope field (i.e. low-to-high, or high-to-low). The channel to trigger from is specified by its network address in the Trig Net#, Trig Dev#, Trig Mod# and Trig Chan# fields.

Three trigger types are allowed, as specified in the Trigger field: Off, Auto and Norm. If Off is selected, data acquisition begins as soon as the Start button is pressed in the Record Page. If Auto is selected, data acquisition begins after the trigger criteria is met, but if the trigger condition is not met within a second or so, the recording begins anyway. If Norm is selected, instruNet waits until the trigger condition is met, indefinitely of necessary.

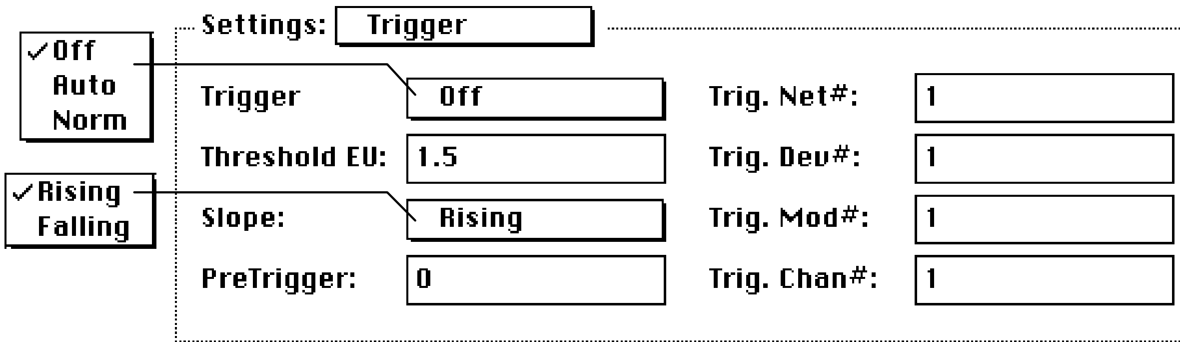
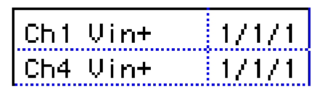


Fig 2.2 The Trigger Dialog



- Enter the address of a channel to trigger from into the Trig Net#, Trig Dev#, Trig Mod# and Trig Chan# fields. If you are not sure of a channel's address, go back to the Network Page and look at the Channel and Addr column for the channel you want to trigger from. The address for the two channels shown in the above figure would be {1,1,1,1} and {1,1,1,4}. For example, if you wanted to trigger from channel Ch4 Vin+, you would enter the following values: 1 into Trig Net #, 1 into Trig Dev#, 1 into Trig Mod#, and 4 into Trig Chan#.

- Select Auto in the Trigger popup, type a reasonable threshold voltage into the Threshold EU field (e.g. 1V) and then select Rising or Falling in the Slope popup. Click OK to exit the Trigger Dialog, click OK to exit the Setup Dialog and then click the Start button to begin recording.

The waveforms should appear on the screen, with the beginning of each Scan synchronized to the trigger event. If the signal applied to the trigger channel does not periodically cross the threshold voltage, Auto trigger will digitize anyway every second or so.

- Press Stop when you are done acquiring.

To learn more about Triggering, please refer to the Record page discussion in *Chapter 5*.

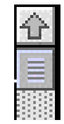
- Click the Setup button at the top of the Record page to open the Record Setup dialog, as shown in Figure 2.1.

Display Options

The Horiz Scale field sets the display horizontal scale in seconds-per-division. If set to Auto, instruNet picks a horizontal scale that is appropriate based on the sample rate and number of data points being acquired. This popup is also provided at the lower right of the Record page when iW+ is installed.

The Plot popup is used to set the drawing mode to plot Dots or Lines (i.e. light one pixel for each data point, or connect these data points with lines), and the Grid popup (accessed by pressing the Record button in Setup dialog) selects whether or not to overlay a grid on each display.

Viewing Many Channels

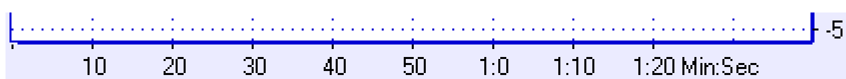


The Disp Height field sets the minimum height of each display in the Record page, in pixels. If the number of waveforms being digitized is greater than the available space on the screen, only a subset are displayed, and the vertical scrollbar selects that set. For example, if the Record page is 1000 pixels high, the Disp Height field is 100, and one is digitizing 512 channels (the maximum amount); only 20 would be shown at one time, and the vertical scrollbar would select that set of 20. In another example, if the window was 640 pixels high, one was digitizing 32 channels, and one wanted to see all channels at once; one would need to set the Disp Height field to 20 (i.e. 20 pixels * 32 channels = 640pixels).

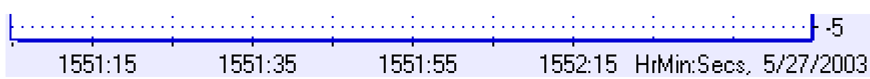
Another technique is to hide individual displays by setting the Display View field to Hide for each channel that you do not want a Display in the Record page. One does this by opening the channel options dialog for each channel you want to hide, select the Display setting, and then set the Display View field to Hide. If you want to view channels numerically, then please consider Panel Meters, which are only available in iW+ (described at www.instrunet.com/plus).

Time Of Day *or* Relative Time

The Horizontal Label field allows one to show either Relative time or Time of Day in the Record page horizontal timescale, as illustrated below. Relative time shows the number of seconds since starting the digitization at 0 seconds. Time of Day shows the time of day, as known by the computer's clock. For example, if you start digitizing at 1:05 am on Feb 1, 2003, it will show "0117, 2/1/03" 12 minutes later. Post-Acquisition, one can change this field to view the data with either timescale format. Displaying Time of Day requires instruNet World PLUS (iW+).



Time Relative to start of digitization, left edge corresponds to 0 secs



Time of Day, e.g. first grid corresponds to May 27, 2003, 3:51pm + 15 seconds

Digitize Into Ram *or* File

The Digitize Into popup has 2 primary settings: To Ram Buffer, which saves digitized data into RAM; and To File, which digitizes data directly to disk. If Digitize Into is set to To File, instruNet automatically prompts the user for a folder name every time a recording session is initiated with the Start button. The waveforms are then saved to this folder while they are recorded. One can then scroll through these long disk-based waveforms (e.g. 20M points per channel) via the horizontal scrollbar. Any waves saved to disk using the To File option can be opened and scrolled through with the Record Page's Open button.

The File Type field determines the file format for the saved data, and is set to one of Binary, Binary Merge, Text, or Text Merge. For a detailed description of each format, please see File Type Application Notes.

- Select Lines in the Plot popup, select Off in the Grid popup, select "Strip Chart" in the Scan Mode popup and, set the Pts per Scan field to 10000 to set the buffer size of an intermediate RAM buffer that holds data before it is sent to disk (10000 points at 1Ks/sec is a comfortable size). Set the Digitize Into popup to "To File", and select 0.5 secs/div in the Horiz Scale popup. Click OK to exit the dialog. Click Start to begin recording. When the File save dialog appears, type a folder name and select a location for the waveforms that are about to be "spooled" to disk.

- After a minute or so, press the Stop button to stop digitizing. Scroll through your waveforms via the horizontal scrollbar. Notice that the computer goes to your hard disk periodically to automatically load in information from disk.

RAM-Based Digitizing

It is recommended that one Digitize Into RAM if your RAM is large enough. RAM based digitizing is easier, since data in RAM can easily be saved in different file formats, is easily loaded back into RAM from disk to be saved to disk in another file format, and supports faster digitizing rates. Due to these advantages, we recommend digitizing directly To Ram unless your RAM is not large enough to hold the data. To digitize into RAM, set the Digitize Into field to "To Ram Buffer", set the No Of Scans field to 1, and then use the Pts Per Scan field to determine how long you digitize. If you digitize multiple scans directly To Ram, data is overwritten in the RAM buffer and lost; therefore, we set the No Of Scans field to 1. After digitizing into RAM one can press the Save button in the Record page to save the data in the RAM buffer to disk in the format specified by the File Type field. To transfer data to another software package, one typically sets File Type to "Text Merge". This causes a file named "Excel Waveform Data.txt" to be saved to disk that is easily opened by a spreadsheet, with each channel in its own column. To save RAM based data in a compact fast format, we recommend File Type "Binary Merge". To calculate the amount of RAM used to hold your data, multiply the number of points, by the number of channels, by 4bytes-per-point. For example, 3 channels of 10K points each would consume 120KBytes of RAM (120KB = 4 * 3 * 10000).

File-Based Digitizing

It is recommended that one Digitize Into File for bigger-than-RAM data, yet one must consider how they will process the huge disk-based file. To Digitize Into File, set the Digitize Into field to "To File", set the File Type field to "Binary Merge", set the Sample Rate field to the desired points-per-second-per-channel, set the Pts Per Scan field to a nominal value (e.g. 5000) to set the intermediate RAM buffer size, and then set the No Of Scans field to the number of RAM buffers of data that are collected. For example, digitizing 1000 scans of 5000pts-per-scan data digitized at 1000pts-per-sec will spool to disk a total of 5M points over a 5Ksec period. When the Start button is pressed in the Record page, it will prompt you for a file name before digitizing, and send the data directly to disk.

The main issue, when digitizing directly To File is, "How are you going to deal with all that data on disk?". instruNet World will not allow you to load it into ram (since file based data is typically too large to fit into ram) and then save it back out in another file format. It will only allow you to scroll though and view the data (it automatically pages in segments from disk, as needed, for display). And to digitize To Disk quickly, you need to use the Binary Merge File Type, which interlaces all channels into one file, in 32bit floating point form. There is physically no other way to spool to disk at fast rates without saving in this manner. Therefore, to process a large disk based stream, one typically needs a software package that interprets 32bit floating point interlaced data. For details on this file format, please the File Format Application Notes.

If you want to save data to a spreadsheet program, post-acquisition, please refer to Appendix III. If you want to digitize directly to a spreadsheet, while acquiring data, please see the *Digitizing Directly to Excel* discussion, later in this chapter.

To learn more about Setup Options, please refer to the Record Page discussion in *Chapter 5*.

- Try various options and settings to gain some familiarity with the wonder world of instruNet World. Some things to try are listed below:

  - Press the <u>Start</u> button to start recording again.
  - Press the <u>Save</u> button to save the digitized waves to disk (if they are RAM based).
  - Press the <u>Open</u> button to load previously recorded waveforms from disk.
  - Press the <u>Setup</u> button to adjust the sample rate and number of points that are digitized when the Start button is pressed.
  - Press the <u>Trigger</u> button within the Setup Dialog to adjust the trigger options.
  - Press the <u>Network</u> tab to select the Network page, and then turn on other channels for digitizing by clicking on their digitize on/off cells.

The instruNet Data Tree

instruNet stores field settings in a hierarchical data tree illustrated in figure 2.4.

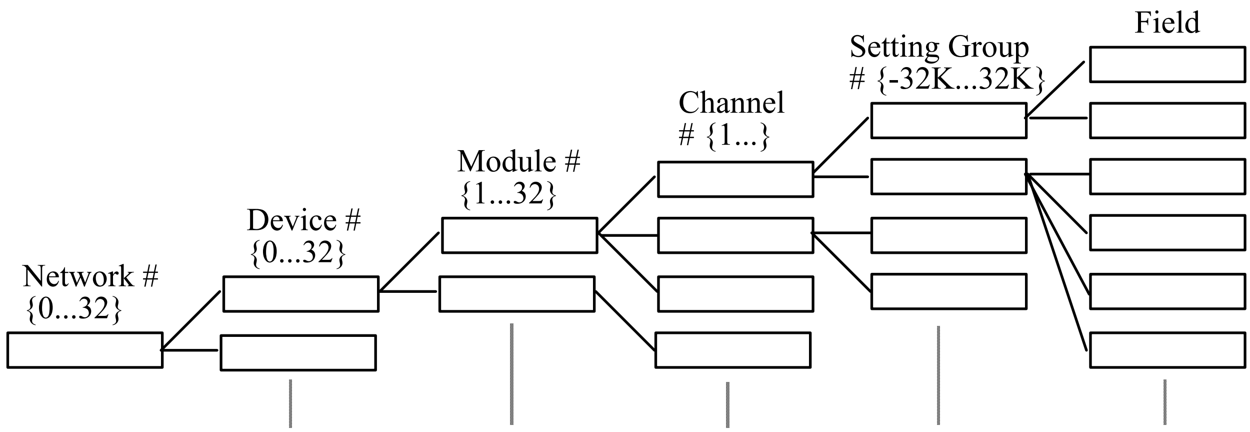


Figure 2.4 Network Hierarchy for instruNet

To access a piece of information, you must supply an address within this data tree. This address consists of 6 parameters, as described below:

Network Number If 0, this refers to the Driver itself (e.g. plot lines or dots in the Record displays); otherwise, this number refers to an instruNet Controller board, where the first board found in the computer is designated Network Number 1, the 2nd board found is Network #2, etc.

Device Number This refers to hardware devices (e.g. Model 100) attached to an instruNet Controller board, where the hardware Device closest to the Controller is Device #1, the next device is Device #2, etc.

Module Number This refers to the module within a hardware device. At this time, all Devices have only 1 module that is referred to as Module #1.

Channel Number This refers to a specific channel in a hardware device. Each channel typically corresponds to a physical wire somewhere, such as a voltage input, voltage output, digital input, or digital output. For example, in the Model 100, the screw terminal marked "Ch1 Vin+" is Channel #1 and is a voltage input.

Setting Number Each channel includes different Settings areas such as: lowpass filter settings, highpass filters settings, Hardware settings, etc. It is here that one selects a settings group (e.g. Lowpass Filter fields have a Settings Number of -9).

Field Number This is the Field Number {1..8} within a settings group. For example, in the Lowpass Filter settings group, the cut-off frequency in Hertz is stored in Field #5.

The instruNet World user navigates within this data tree via the Probe dialog, described in the next discussion. instruNet World does not require the user to know about Setting numbers and Field numbers since all items are defined using popups and edit fields. The programmer, on the other hand, must supply 6 numbers to a subroutine to read and write to any field on the instruNet data tree.

The Network page shows the current Field settings for each channel in a tabular (i.e. spreadsheet) format, and is also a useful tool for navigating around the instruNet data tree. The data tree maintains any changes you make until you Reset the network via the Reset button, reset the computer, or load in new setting from disk via the Restore or Open buttons at the top of the Network page. In many cases, a user will set the fields as needed, stored them to disk, and then reload them when instruNet world is first opened.

Explore Your World

- Exit all instruNet software and run instruNet World.
- Select the Network page by pressing the Network tab at the bottom of the window.
- Press the Reset button at the top of the window to reset all Fields in the Data Tree. Press OK when a dialog asks for confirmation.

Figure 2.5 illustrates how information is organized in the Network page. The channels that are displayed on your computer will vary depending on what hardware is installed; therefore, don't worry if your screen is a little different from the Figures.

Channel #
Network #
Device #
Module #

Channels

Channel	Addr	Value Input	Units	% sRate	Sensor	Wiring
Ch1 Vin+	1/1/1	+2.919919	Volts	100	Voltage	Vin - Gnd
Ch4 Vin+	1/1/1	-0.000338	Volts	100	Voltage	Vin - Gnd
Ch7 Vin+	1/1/1	+0.007571	Volts	100	Voltage	Vin - Gnd

Fields

Figure 2.5 Partial view of the Network Page

The Network Page

Each row in the Network page corresponds to an input or output channel, which is often associated with a physical sensor in the real world. Each channel has a {Network, Module, Device, Channel} address both within the software data tree and the physical outside world. This address is shown in the first 4 columns of the Network page. The first column indicates the Channel name and number. For example, "Ch1 Vin1+" is Channel #1 and the channel name is "Ch1 Vin1+". Columns #2 through #4 indicate the channel's Network #, Device # and Module #, which correspond to a physical address.

The column labeled "Value Input" depicts the current real-time value (input or output) of the channel, in engineering units. All columns to the right of the Value Input column are Fields that specify the type of signal connected to the channel and how it is being read. The horizontal and vertical scroll bars are used to move around and make changes to the tables contents. To change a Field's setting, one can click on its cell and then change its value. For example, to change the name of channel Ch1 Vin1+, one would click on the "Ch1 Vin1+" cell.

- Click on any cell in the Units Label column to open the Probe dialog, as shown in figure 2.6.

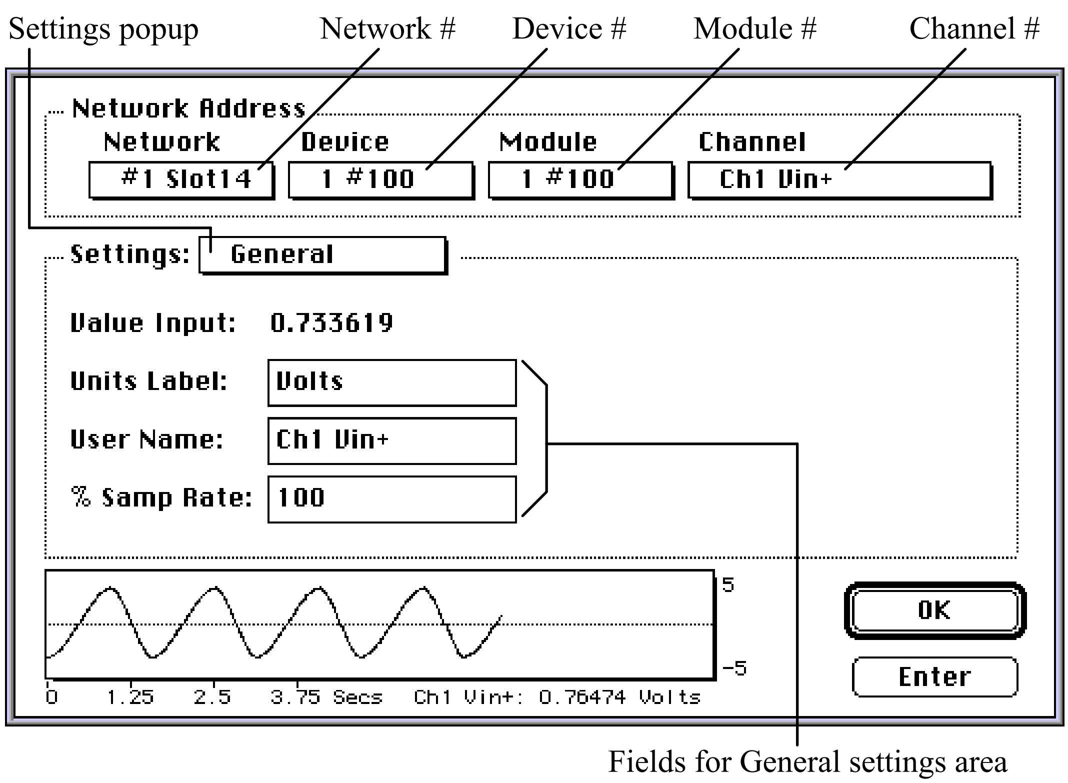


Figure 2.6 The Probe Dialog

The Probe Dialog

Using the Probe dialog, any Field within the instruNet Data Tree can be viewed or modified. The upper-most 4 popup menus specify a channel address (i.e. Network, Device, Module, Channel), the Settings popup specifies a Setting group (e.g. General, Lowpass Filter, Highpass Filter, etc), and the Settings area shows between 1 to 8 Fields depending on the Settings group selected.

For example, in Figure 2.6, we are viewing the General settings for channel Ch1 Vin+, which is physically connected to the 1st Device (a Model 100) attached to the 1st Controller (i.e. Network), which is plugged into computer slot #12. The General settings group contains 4 Fields: Value Input, Units Label, User Name, and % Sample Rate.

The Value Input field shows the real-time, current value, of the channel, Units Label is the displayed label for the channel's value (e.g. "Volts", "Amps", "C"), User Name is the user's name for the channel (e.g. "Temp 1", "Pressure 2", etc), and % Sample Rate is the speed, as a fraction of the Master Sample Rate, that the channel is digitized (e.g. 50% would mean the channel is digitized at one-half the sample rate specified in the master Setup dialog). The small display at the bottom of the Probe dialog shows a plot of the current real-time value of the Channel.

- Click on the Units Label field and change "Volts" to "Amps". Click OK to exit the Probe dialog.

The clicked on cell should be update to "Amps", as shown in figure 2.7.

Channel	Addr	Value Input	Units
Ch1 Vin+	1/1/1	3.2143	Amps
Ch4 Vin+	1/1/1	-0.001557	Volts
Ch7 Vin+	1/1/1	+0.007571	Volts

Amps

Figure 2.7 Edited cell within the Network page

- Scroll through the Fields of the Network page using the horizontal scroll bar at the base of the window.

Notice that the first 5 columns remain fixed, while the cells to the right of Column #5 shift left and right with the horizontal scrollbar. You can scroll through, and view, all Fields for all Channels in this manner. Figure 2.8 shows the Network Page for Ch1 Vin+ after scrolling a little to the right.

Channel	Addr	Range	Ro	Rshunt
Ch1 Vin+	1/1/1	+- 5V	100	5000

Figure 2.8 Network Page scrolled horizontally to view the Ro & Rshunt Fields

- Scroll horizontally to the left edge so that "Value Input" is in Column #6 and then scroll vertically until an input channels is no longer in the top row. For example, in Figure 2.9, a Voltage Output channel is in the top row.

Notice that the title to Column #6 changed from Value Input to Value Output. This is because the titles are optimized for the one channel in the top row.

Channel	Addr	Value Outpu	Units
Ch12 Vout	1/1/1	+0.013493	Volts
Ch15 Vout	1/1/1	+0.013740	Volts

Figure 2.9 Network Page scrolled vertically to view the title for voltage outputs.

- Vertically scroll to the top of the table and then click on the net cell of the first row. (it should contain a 1) This will cause the Probe dialog to open and to display the clicked on cell.

Channel Addresses

The upper region of the Probe Dialog, shown in Figure 2.10, is used to select a Channel's address (i.e. Network Number, Device Number, Module Number, and Channel Number).

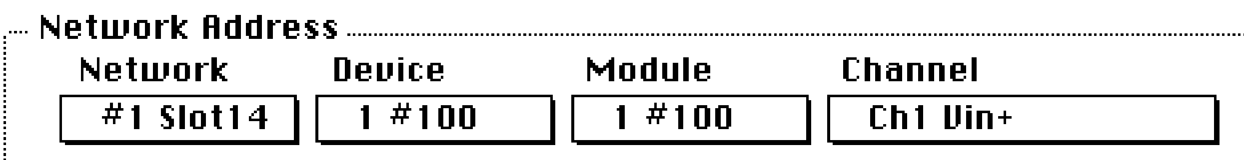
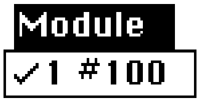


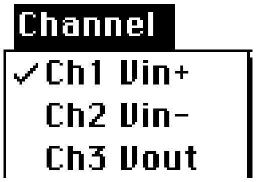
Figure 2.10 Network Address Configuration section of Probe Dialog

When instruNet World resets (e.g. powers on), the expansion slots in the computer are scanned for instruNet controllers. Each controller found is given a Network Number. For example, in Figure 2.10, instruNet World found a controller in Slot 14 of a computer and designated the network connected to it as Network #1. Hence the popup reads #1 Slot 14. This popup enables the user to select any instruNet controller installed on the computer. There is a special "Controller" in the popup that is labeled "Driver" (Network #0). This refers to the instruNet Driver itself, and appears only once in the Network popup no matter how many Controllers appear. The Driver contains fields that determine things like the way displays show data in the Record page.

The Device popup menu lists the network Devices that are attached to the Controller specified in the Network popup menu. When you select a Controller in the Network popup, all devices attached to it appear in the Device popup. In the figure shown to the left, only one network device is attached to the Controller in Slot 14 and it appears as the second item in the popup. It is a Model 100 and is designated Device #1 (1 #100). The Controller itself is a Device (Device #0) and appears as the first item in the Device popup. The Controller contains Fields that are specific to the controller, such as the network sample rate, or the value of a digital output on the Controller's Digital I/O Connector.



The Module popup lists all Modules in the currently selected {Network, Device}. Most Devices only have 1 module, as shown to the left.



The Channel popup lists all analog and digital I/O channels in the currently selected {Network, Device, Module}. The illustration to the left shows 3 channels, 2 of which are voltage inputs, and the 3rd which is a voltage output.

- Explore your instruNet world via the 4 Channel Address popup menus at the top of the Probe dialog, and the Settings popup menu. Press OK when you are done exploring.

Saving & Loading Network Settings

All instruNet Fields (i.e. all the cells in the Network page) are saved to disk and loaded from disk with the press of a button. When a configuration is saved, all information including items such as trigger conditions, sample rates and channel units are stored. Waveform data is not stored at this time, but can be saved by pressing the Save button in the Record page. When a configuration is loaded, all items are restored to their previously saved condition. This means that instruNet configurations for specific experiments only need to be set up once. And once a configuration is loaded, it can be changed and then saved again if needed, possibly in a different file.

- Select the Network page by clicking the Network tab.

The first two buttons at the top of the Network Page, Restore and Store, work as a pair. Clicking the Store button saves the current network settings to a preferences file within your operating system folder. Clicking Restore loads in this file. File open and save dialogs do not appear, since the Fields are always saved to the same file (i.e. a file with the same name). Obviously, you loose your last saved network when you press the Restore button (careful !).

- Press the Store button to save your current Field settings to disk.

- Press the Clear button to erase your Field settings to their default values.

Notice how the "Amps" units label has now returned to its default setting of "Volts".

- Press the Restore button to restore the previously saved settings.

Notice how the "Amps" units label has returned. To save the settings to the file of your choosing, click the Save and Open buttons.

- Press the Save button . Type a file name and select a file location when the File Save dialog appears. Remember where you put this file.

- Now press the Clear button to clear all settings to their default values.

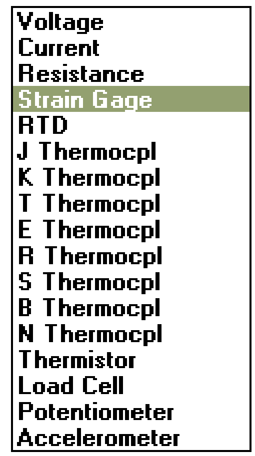
- Press the Open button and select your saved file in the File Open dialog.

Notice how the "Amps" units label now appears. At this time, you have 2 files on your hard disk with your saved network settings.

The Reset button differs from the Clear button in that it resets the hardware in addition to clearing your fields. It has the same affect on an instruNet network as restarting the computer. For example, Reset will reset clock in the controller, whereas Clear will not.

Working with Sensors

Any voltage input channel can attach to any of the following sensors: Voltage source, Current source, Resistance source, Strain Gage, Load Cell, Accelerometer, Potentiometer, RTD, or types J, K, T, E, R, S, B, and N Thermocouples. Sensors can be wired in a variety of configurations including: Differential Voltage Measurement (requires 2 voltage input channels, e.g. Ch1 Vin+ and Chi Vin-), Single-ended Voltage Measurement, Shunt Resistor, Voltage Divider, Bridge, Quarter Bridge and for strain gages: Half-Bridge Bend, Half-Bridge Axial, Full-Bridge Bend, Full-Bridge Axial I and Full-Bridge Axial II.



- Select the Network page by clicking on the Network tab.
- Click on the name of the voltage input channel with the attached signal source (e.g. "Ch1 Vin+").

The Probe dialog will open with the address of the channel you clicked on displayed in the Network Address. Additionally, the real-time value of the channel, in Engineering Units (EU) will appear at the bottom of the display, as shown in Figure 2.11.

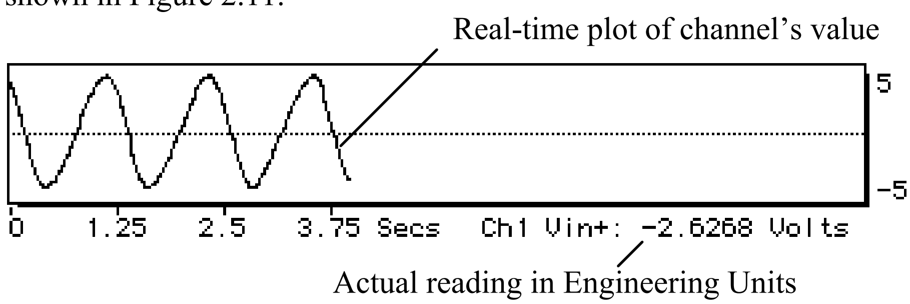


Figure 2.11 The Snapshot Display

- Select Hardware in the Settings popup, as shown in Figure 2.12.

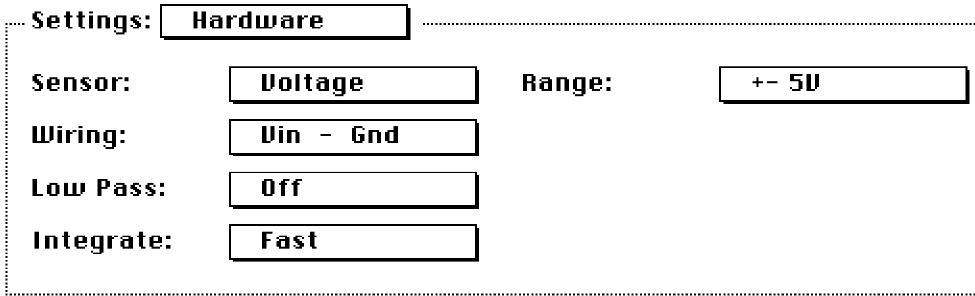
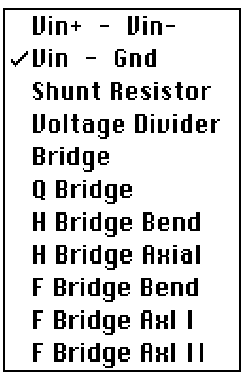


Figure 2.12 Hardware Settings Area

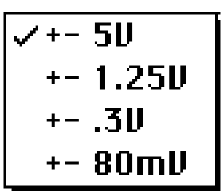
- Click on the <u>Sensor</u> popup to view the various Sensors that can directly be attached to this channel, as shown to the right.

This popup tells instruNet which sensor is connected to your physical hardware terminals (instruNet has no way of seeing what is out there). For more detailed information on connecting sensors to instruNet, please refer to *Chapter 3 Connecting to Sensors*.



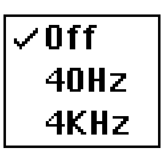
- Click on the <u>Wiring</u> popup and view the choices, as illustrated to the left.

The <u>Vin+ - Vin-</u> option is used for differential Voltage measurements to measure the voltage <u>between</u> the Vin+ and Vin- terminals. The "common" signal on both terminals is ignored, and therefore this technique can be used to reduce noise. The <u>Vin+ - Gnd</u> option specifies Singled-Ended voltage measurement, which measures the voltage between the voltage input terminal and the Ground terminal. The latter 6 options (Q Bridge, Half Bridge Bend, Half Bridge Axial, Full Bridge Bend, Full Bridge Axial I and Full Bridge Axial II are used to specify a wiring options when working with a <u>Strain Gage</u> sensor. These wiring options are described in more detail in Chapter 3.



- Click on the <u>Range</u> popup and view the options, as illustrated to the left. Select the largest range (e.g. +- 5V).

This Field specifies the voltage input range. Accuracy is increased as the range is reduced. For example, a +-80mV range might be accurate to +-45uV, whereas a +-5V range might only be accurate to +-1.5mV. If you input a voltage in excess of a bound, the bound is read. For example. If you apply 3V a voltage input with a +-1.25V range, then +-1.25 will be read by the computer.



- Click in the Low Pass popup and view the options, as illustrated to the left.

The options that you see will depend on the connected hardware device. This Field is used to select an analog filter at the front end of the voltage input amplifier. Please consult *Chapter 6, Hardware Reference* to learn more about the analog filter options for each hardware Device.

The Integrate field specifies how long, in Seconds, instruNet averages an input signal before 1 number is returned to the user. This is often used to reduce high frequency noise that has been added to a signal. The integration feature is implemented by sampling the signal many times with the A/D converter, as fast as it can, and then averaging the A/D values with software. The maximum allowable integration time depends on the number of digitized channels and the sample rate. For example, 2 channels could be sampled at 1000s/sec per channel and integrated each for .5ms.

- Select <u>Constants</u> in the <u>Settings</u> popup, as illustrated in Figure 2.13.

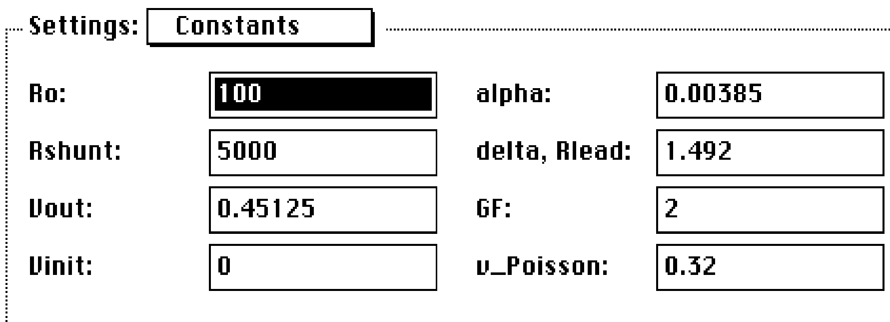
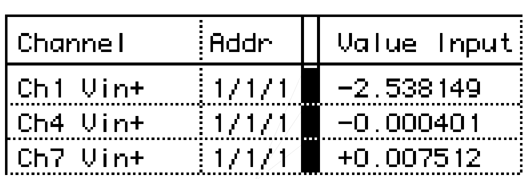


Figure 2.13 Constants Settings Area

These Fields are used to specify constants that are used to calculate engineering units when working with Resistance, Current, RTD, Load Cell, Accelerometer, Potentiometer, and Strain Gage sensors. For example, <u>Rshunt</u> specifies the value of the shunt resistor, in ohms, when doing a Resistance measurement. Please refer to Chapter 3 for details on how to use these.

- Click OK in the Probe Dialog to return to the Network page.

- Enable the first three voltage input channels for digitizing by clicking once on Column #5 of each channel, as illustrated in Figure 2.14.



Click here to enable digitizing

Figure 2.14 First 3 channels of Model 100 are enabled for digitizing.

- Select the Record page by clicking on the Record tab at the bottom of the window.

- Click the Start button to begin recording.

The Record Page automatically creates a separate display for each recorded channel, as shown in Figure 2.15. The actual signal that appears will depend on the connected signal sources.

- Click the Stop button to Stop recording.

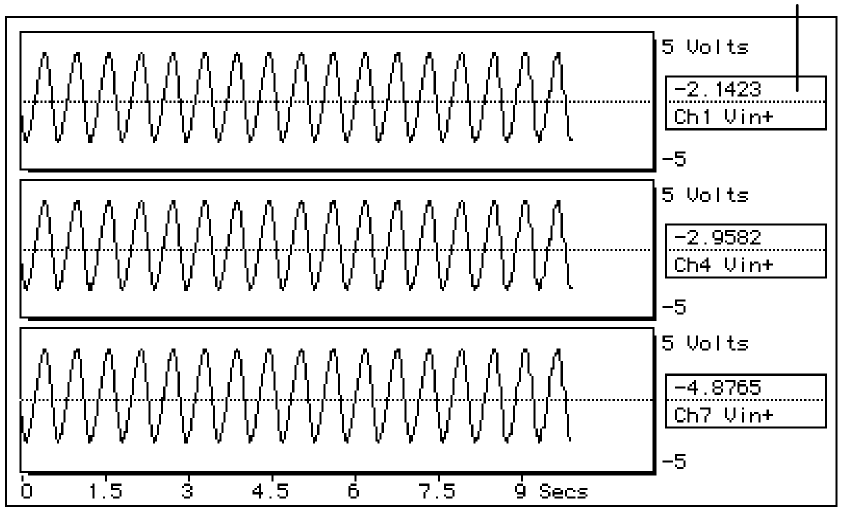


Figure 2.15 Three Channels in Record Page

Each display has a <u>Channel Name</u> box which appears to the right of the display, as shown to the left and in figure 2.15. This box displays the channel name and the real-time channel value in Engineering Units.

- Click on the <u>Channel</u> <u>Name</u> box of the top-most display to open the Probe dialog at the Display settings area, as show in Figure 2.16.

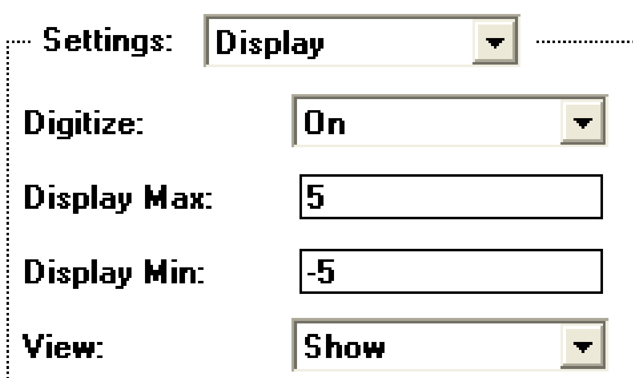


Figure 2.16 Display Settings

- Change the <u>Display Max</u> Field to 2, change the <u>Display Min</u> Field to -2, and click the lower-right Enter button.

These 2 Fields are used to set the top and bottom plot values of the vertical axis in both the Record page display and the Probe dialog snapshot display. These changes take affect when the Enter button is pressed, and can be viewed at the bottom of the Probe Dialog, as shown in Figure 2.17. In many cases, one must edit these values, depending on the Engineering Unit range of the digitized signal. For example, one might set 0 and 100 for a temperature that ranges from 0 to 100C.

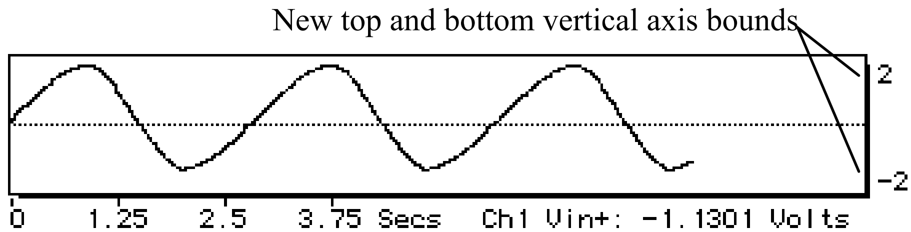


Figure 2.17 Probe display with E.U.'s set to +/- 2V

- Press OK to exit the Probe dialog, and then press Start to begin recording.

Notice how the vertical axis scale change effects the appearance of the recorded signal, as shown in Figure 2.18.

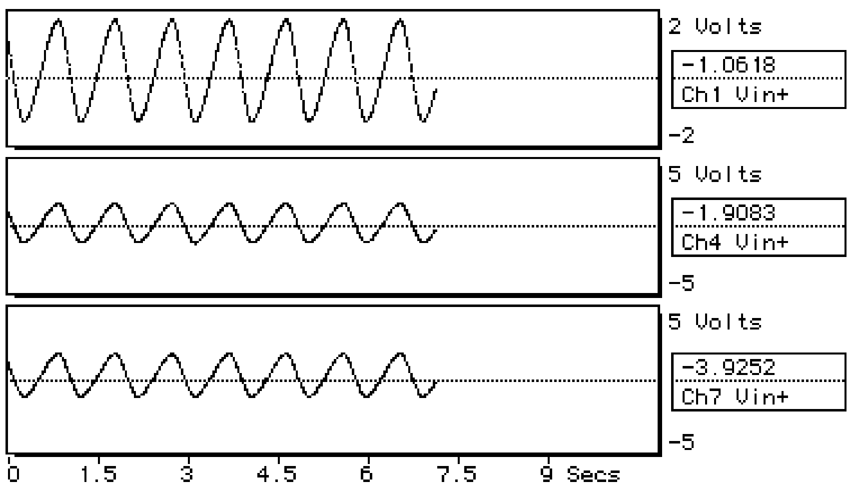


Figure 2.18 Record Page with top display Min/Max E.U. set to +/- 2V

Working With Calibration, Different Scales, & Mapping

instruNet supports Calibration and Converting to different scales (e.g. show psi at sensor instead of Volts at screw terminals) with a 2 point mapping scheme. All channels have a Mapping settings area that defines the relationship between "internal units" and "external units". Internal units are the native units used by instruNet, such as Volts. External units are what the user sees in the Record page, the Network page, and the numbers returned by iNet(). External units are linearly mapped to Internal units, and are therefore defined with 4 numbers, that define 1 line ({x_1,y_1}, {x_2,y_2}), on an Internal Units Vs. External Units 2-dimensional coordinate axis plane.

Sounds a little complicate? Imagine a linear temperature sensor that puts out .1Volts when dipped in ice water (0°C) and 1Volts when dipped in boiling water (100°C); and you want instruNet to display °C numbers in the Record page and the Network page. To do this, one would set the Mapping fields to:

$internal_1$: .1 (Volts)	$internal_2$: 1.0 (Volts)
$external_1$: 0 (°C)	$external_2$: 100 (°C)

And set the Units Label field in the General Settings area to "C". The Units Label does not effect instruNet numerically, yet is interpreted as random text that is simply placed next numbers (i.e. it is a "label"). The Mapping fields; however, effect numbers, yet not labels.

The Mapping numbers can also be used to implement calibration. Suppose a thermocouple (i.e. temperature sensor) is attached to instruNet and is already returning °C numbers (since that is the native units for the thermocouple), yet you find there is a 2°C offset error in your sensor, and want instruNet to "correct" for this error. To do this, the user would set up a mapping from internal units to external units that reflected the offset error. e.g.

$internal_1$: 100 (°C)	$internal_2$: 0.0 (°C)
$external_1$: 102 (°C)	$external_2$: 2.0 (°C)

Notice that Mapping can be used to correct for an offset error by adding a constant and can correct for a gain error by multiplying by a constant.

The Mapping setting area describes a line using two different methods: two points "$\{x_1,y_1\}$, $\{x_2,y_2\}$", and "y = x*scale + offset". One can use either method. For example, instead of setting the above 4 values, one could have set the Offset field to 2.0, to show a 2°C Offset. When the Scale or Offset fields are changed, the x_1, y_1, x_2, y_2 fields update automatically to reflect the new line when the OK or Enter button is pressed.

Working with Digital Filters

All voltage input channels support digital lowpass, highpass, bandpass and bandstop filters. The cut-off frequencies, minimum dB stopband attenuation (i.e. filter order), maximum dB passband attenuation, and filter type (e.g. elliptic, Chebyshev B, Chebyshev S, and Butterworth) can all be programmed separately for each channel via the Lowpass, Highpass, Bandpass and Bandstop Settings areas. In fact, these 4 models can be run at the same time (i.e. in serial) to simultaneously do lowpass, bandpass, bandstop and highpass filtering on the same channel. For example, one might only want to see frequencies between 20 and 1000Hz, except for the 55 to 65 band. This would involve a highpass filter at 20Hz, a bandstop filter between 55 and 65Hz, and a lowpass filter at 1000Hz.

- Select the Network page by clicking on the Network tab and then press the Reset button to reset the network and all Fields.

- Click the channel name cell of the channel that is attached to your signal source (e.g. Ch1 Vin+), and then select Lowpass in the Settings popup, as shown in Figure 2.19.

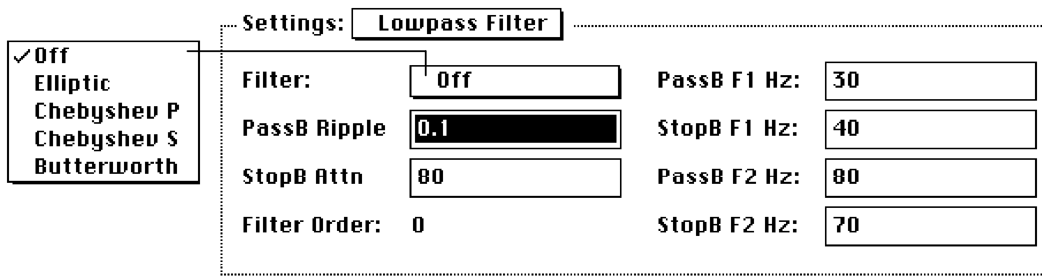


Figure 2.19 Lowpass Filter settings area

The lowpass filter is illustrated in Figure 2.20. Notice that 4 numbers are needed to describe the lowpass filter: minimum stop band attenuation (dB), maximum pass band ripple (dB), pass band cut-off frequency (Hz) and stop band cut-off frequency (Hz). And notice how these 4 numbers corresponds to the StopB Attn, PassB Ripple, PassB F1 Hz, and StopB F1 Hz Fields in the Lowpass Filter settings area.

- Turn the filter On by selecting Elliptic in the Filter popup (or any of the other options other than Off).

- Enter a passband cut-off frequency value into the PassB F1 Hz field, a stopband cut-off frequency into the StopB F1 Hz field, a minimum stop band attenuation into the StopB Attn field, and a maximum pass band ripple into the PassB Ripple field. Acceptable values would be {100, 150, 80, 1}. Press the Enter key when done setting the values. If the filter is impossible to design due to constraints of the specified values, an alert will appear with a message coaching the user into selecting different parameters. In fact, instruNet will not allow you to exit this dialog until the parameters are acceptable, or the filter has been turned off by selecting Off in the Filter popup.

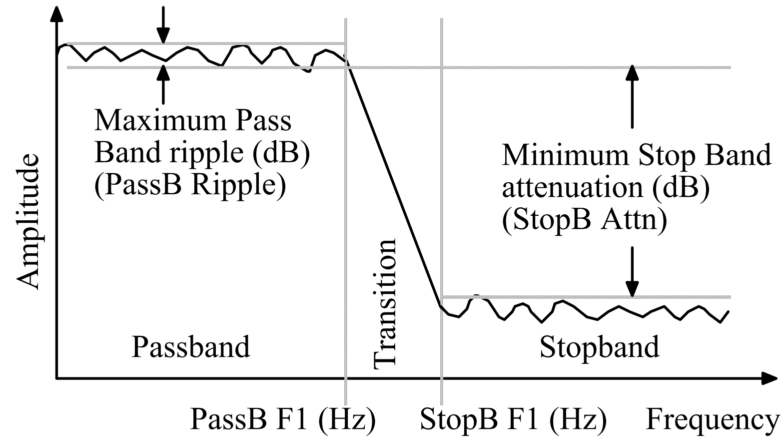


Figure 2.20 Lowpass Filter Model

The filter order ranges from 1 to 32 in bandpass and bandstop filters, and from 1 to 16 in highpass and lowpass filters, depending on the supplied parameters. As the filter becomes more demanding, the filter order increases, and the time to run the filter also increases. In a typical case, it takes .1 to 2us per order per point to run the filter. For example, on a Macintosh 840av (1993 technology) it takes 2us per order per pt or 20us per point to implement a 10th order filter. This would limit the maximum sample rate to 50Ksamples/second (i.e. 1 / 20e-6 = 50e3).

- View the results of your filter by observing the effect it has on the waveform in the Snapshot display. If you have a function generator connected, watch what happens when you slowly change its frequency from 10Hz to 500Hz, for example.

Figure 2.21 shows a 35 Hz signal applied to Ch1 Vin+, before and after the implementation of a 30 Hz lowpass filter.

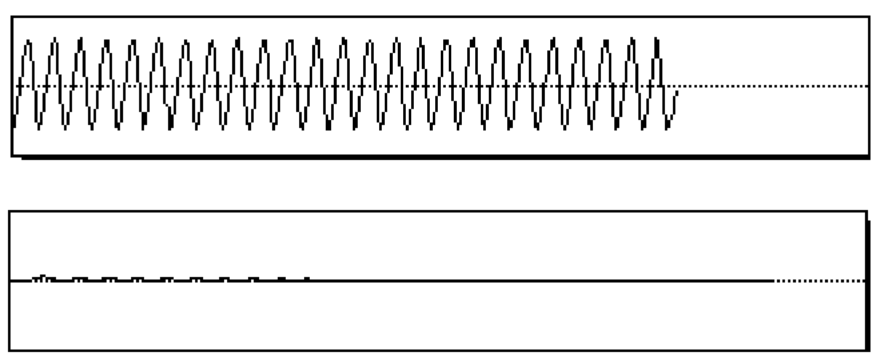


Figure 2.21 35 Hz Sine Wave before and after 30 Hz Lowpass Filter

Figures 2.19 through 2.21 show the filter models for the Highpass, Bandpass and Bandstop filters.

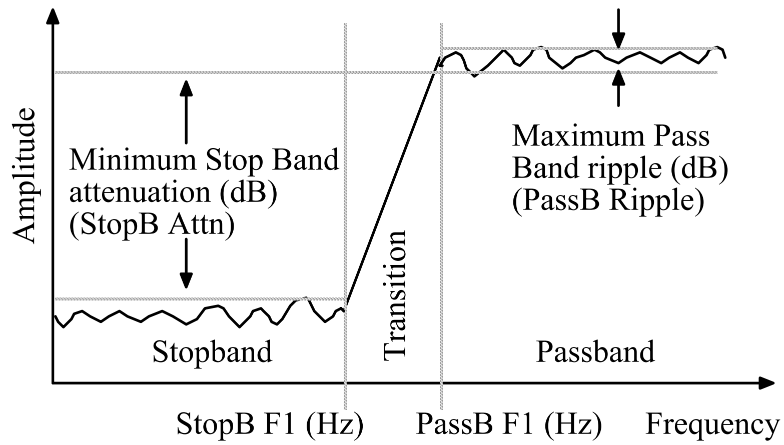


Figure 2.22 Highpass Filter Model

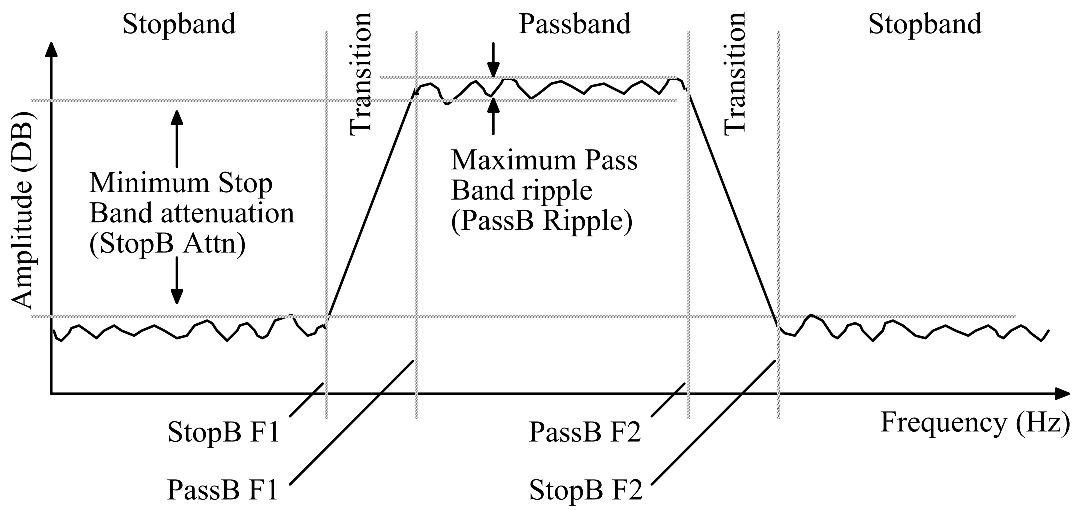


Figure 2.23 Bandpass Filter Model

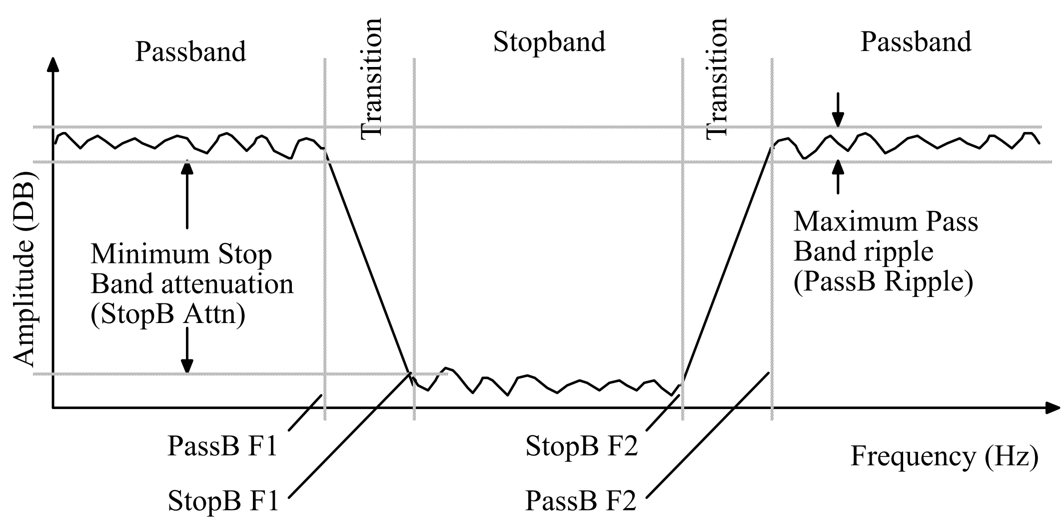


Figure 2.24 Bandstop Filter Model

Table 2.1 shows which fields in the Settings area must be set for each filter model.

Edit field or popup	**Lowpass**	**Highpass**	**Bandpass**	**Bandstop**
Filter popup	used	used	used	used
PassB Ripple	used	used	used	used
StopB Attn.	used	used	used	used
PassB F1 Hz:	used	used	used	used
StopB F1 Hz:	used	used	used	used
StopB F2 Hz:	not used	not used	used	used
PassB F2 Hz:	not used	not used	used	used

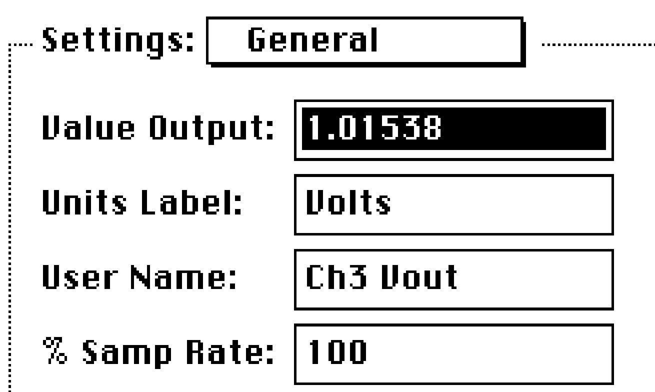
Table 2.1 Fields required for each Filter model

- Play with the different filter models and settings while viewing their effects in the snapshot display. For example, to turn on a highpass filter, select Highpass Filter in the Settings popup, select Elliptic in the Filter popup, and then set the PassB F1 Hz, StopB F1 Hz, StopB Attn, and PassB Ripple fields as desires using Figure 2.19 as a guide. Remember to press the Enter button to create the filter.

Working With Voltage Output Channels

Some hardware devices, such as the Model 100, provide Voltage Output channels. In summary, these channels output the Voltage specified in the Value Output field within the General settings area. Accuracy's are discussed in *Ch6, Hardware Reference*. Voltage Output channels are named ChX Vout in the Network Page.

- In the Network page, click on the 1st cell of the first voltage output channel. For the Model 100, this would be the "Ch3 Vout" channel. When the Probe dialog opens, set the Value Output field to 1, and press the Enter button, as illustrated to the right.



The analog output channel is immediately updated to the new value when Enter is clicked.

- Click OK to return to the Network page and view the Value Input column for the chosen Voltage output channel. It should display a value in the vicinity of 1V.

With Voltage outputs, instruNet reads back the output voltage and displays this value in the Value Out column. This is useful information when loading of the output signal will change its value. An example of this is a bridge excitation circuit where it is important to know the value of the excitation voltage within a small margin (e.g. to ±.01%) yet the actual voltage only needs to be within several percent of the target voltage.

Working With Digital I/O Channels

Some hardware Devices provide Digital I/O channels. For example, the Model 100 Device offers 8 bi-directional digital I/O bits where 8 different terminals physically marked "DIO1..DIO8" can be set up independently as either a digital input or digital output. With the Model 100 digital outputs, 0 to .8V is a logic 0, and 2V to 5V is a logic 1; however, with the Model 100 digital inputs, 0 to 1V is a logic 0, and 3.2V to 5V is a logic 1. Also with these signals, voltages above 12V or below -12V will result in physical damage (careful !). These 8 hardware bits are all handled by 1 instruNet channel named "Ch25 Dio", where the 8bits are read from or written to with a 1 byte word (i.e. 8 bits in a number between 0 and 255).

To set digital output bits, such as those on the Model 100, it is necessary to do the following kinds of things:

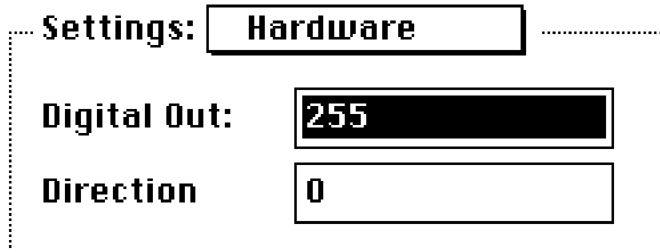
- Click on the cell that contains the name of a Digital I/O Channel (e.g. "Ch25 Dio") and select Hardware in the Settings popup menu.

The Direction field specifies the bit direction, where a value of 1 specifies output, and 0 specifies input. This field packs the 8bits into one 0 to 255 number, where d0 corresponds to DIO1, d1 corresponds to DIO2, etc. For example, $4_{10} = 00000100_2$ would specify DIO3 as an output, and the rest of the hardware bits are inputs. A Direction value of $0_{10} = 00000000_2$ specifies all hardware bits as inputs, and a value of $255_{10} = 11111111_2$ specifies all bits as outputs.

The Digital Out field is used to set the state of the bits that have been marked as outputs by the Direction field. It does this with one 0 to 255 number where each bit in the byte sets its corresponding hardware bit. For example, if 8 bits were set up as outputs, then a Digital Out value of $17_{10} = 00010001_2$ would set DIO1 and DIO5 to a logic 1, and the rest to a logic 0.

The Value EU field in the General settings area is used to read the states of the 8 bits via a 0 to 255 number where each bit in the received byte corresponds to a hardware bit. For example, reading $3_{10} = 00000011_2$ would mean that hardware bits DIO1 and DIO2 are high, and the rest are low.

- Set the Digital Out field to 1_{10} = 00000001_2, set the Direction field to 3_{10} = 00000011_2 and click the Enter button to update the bits.



- Select General in the Setting popup menu and note the value displayed in the Value EU field.

Value EU should show 253_{10} = 11111101_2. This is because the high 6 bits have been set up as digital inputs in the Direction field, and if nothing is connected to them, they will float high to a logic 1. DIO1 is set up as an output in the Direction field, and has been set to a 1 in the Digital Out field, and is therefore read as a 1. DIO2 has also been set up as an output in the Direction field, yet has been set to a 0 in the Digital Out field, and is therefore read as a 0.

Working With i200 Controller Digital Timer I/O Channels

The Model 200 (not 230) instruNet Controller boards have 10 digital input/output channels, each of which can be operated in the following modes:

- Digital Input: Read logic 1 (>2V) or logic 0 (<.8V) at input pin.

- Digital Output: Set connector pin to logic 1 (>2V) or logic 0 (<.8V).

- Clock Output: Output a continuous clock, or a fixed number of pulses {1...2e9}. Set high time and period of clock. If period > 10ms, then we generate clock with a 1ms timebase that drives two 0 to 32K timers; otherwise, we use a .25us timebase. Periods must be >100us when fixing the # of pulses.

- Pulse Counter: Read a 64bit counter that counts incoming >100us pulses since reset.

- Period Measurement: Measure the duration of 1 to 2e9 periods by counting a .25us or 1ms timebase with a 64bit counter via the following options:

  a) "Cycles" - aggregate number of periods to measure (between 1 and 255, or multiple of 10 if between 256 and 2e9). e.g. 100 cycles of 10Hz yields 10sec. Minimum pulse duration is 100us when measuring < 10 periods; and >10us otherwise.
  b) "Measure" sum of entire cycle or just sum of high durations.
  c) Count high "Resolution" 0.25us or lower resolution 1ms timebase.
  d) "Modes":

- Wait & Continue: Wait till measurement completes, start another measurement after read result.
- Wait & Reset: Wait till measurement completes, start another measurement when next read.
- No Wait & Continue: Read 0 if not done, , start another measurement after read result.
- No Wait & Reset: Read 0 if not done, start another measurement when next read.

- <u>Frequency Measurement</u>: Same as Period Measurement, except it returns the frequency of the input signal after measuring the period for "Clk Period" seconds. It checks to see if it passed the "Clk Period" second duration every "Cycles" (1...255) input pulses. For example, if you set "Cycles" to 100, and are measuring a ~10KHz signal, and set the "Clk Period" to 0.1sec, w/ a 0.25us resolution, then it will count the 4MHz timebase w/ a 64bit counter (overflows after 146,000 years). It will check to see if it passed the .1sec mark every 100 pulses (e.g. 10ms for 10KHz input). So it might stop after .11secs and see a timebase count of ~440,000 for ~1100 cycles. It will then divide these 2 numbers and return to you a VERY accurate frequency measurement (e.g. 10,000.00). Notice that traditional frequency measurement devices count a timebase for an specific duration, and error by the fraction of the input cycle that has transpired since its last trigger edge. The minimum pulse duration is 100us when measuring < 10 periods; and >10us otherwise.

- <u>Quadrature Measurement</u>: Returns the 4x position of a quadrature sensor with a +/-32K counter. Connect wire A to Channel N, connect wire B to Channel N+1, and program Channel N for Quadrature, and away you go. For details, see Application Note #110.

Each counter/timer channel runs independently of the others, and of the other channels on the network. For example, "Ch1 Timer" is the first channel in a controller board. Its network address is Device 0, Module 1, Channel 1; and its physical location is two pins on a 34 pin header connector located on the controller board. The two connector pins are labeled "Ch1 Din" and "Ch1 Dout", one for digital input and one for digital output. With the controller digital <u>outputs</u>, 0 to .8V is a logic 0, and 2V to 5V is a logic 1; however, with the controller digital <u>inputs</u>, 0 to .8V is a logic 0, and 3.5V to 5V is a logic 1. Also with these signals, voltages above 6V or below -6V will result in physical damage (careful !). An instruNet Field is used to specify the function of Channel 1 as digital input, digital output, clock output or period measurement. If digital input or period measurement are chosen, the "Ch1 Din" pin is used; otherwise, the "Ch1 Dout" pin is used.

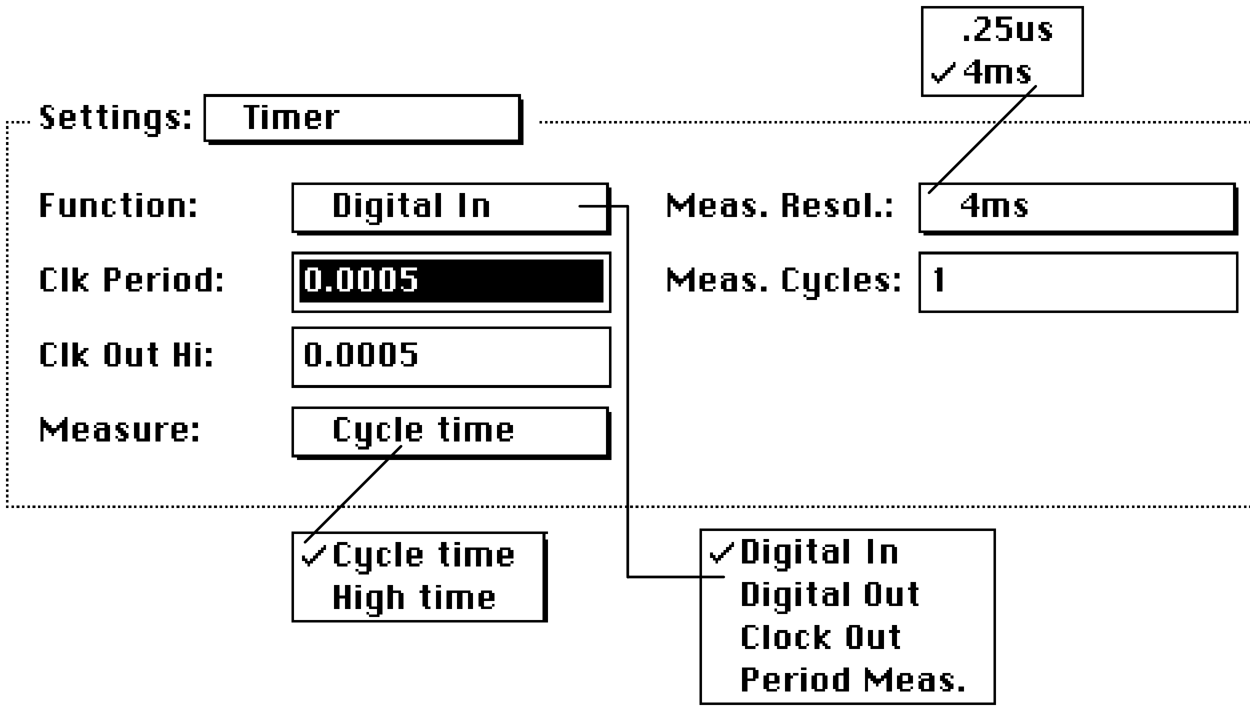


Figure 2.25 The Timer Settings Area

Physically, all controller channels are accessed at a 34 pin header connector (.025" square pins with .1" spacing) located on the Controller board. This connector is independent of the DB25 Connector that is used to cable to the network Devices. One typically cables to this connector via a 34-wire ribbon cable that terminates outside the computer at a screw terminal panel. Please refer to *Chapter 1 Hardware Installation* for details on cabling, and *Chapter 6 Hardware Reference* for details on each pin within this connector.

- Select the Network page by clicking the Network tab, and then click in Column #1 of the first controller's Ch 1 Timer channel (you might need to scroll down a little). When the Probe dialog appears, select Timer in the Settings popup, as illustrated in Figure 2.25.

The Function popup selects the channel's Function, as described below:

Digital Output The "Ch*X* Dout" connector pin is set to a logic 0 (0-.8V) if the Value I/O field in the General settings area is set to a 0; otherwise, this pin is set to a logic 1 (2V-5V).

Digital Input The Value I/O field in the General settings area is read as 0 if the "Ch*X* Din" connector pin is held by an external source below .8V (otherwise, it floats to 5V). However, if this pin is above 3.5V, it is read as a 1.

Clock Output A square wave clock signal is driven out of the "Ch*X* Dout" connector pin where the cycle time is specified by the Clk Period field, in units of seconds, and the high time is specified by the Clk Out Hi field, in units of seconds.

Period Measurement The duration of the signal applied to the "Ch*X* Din" connector pin is measured and returned in the Value I/O field of the General settings area, in Seconds units. If the Measure field is set to Cycle time, the time is measured between consecutive falling edges; otherwise the time is measured between a rising edge and the next falling edge. The Meas Cycles field varies from 1 to 255 and sets the number of cycles (or high times) that must occur over the measured duration. For example, measuring 10 cycles of a 1KHz square wave would return 10ms; whereas measuring 5 high times of a 20KHz square wave would return 125us. If the Meas Resol popup is set to .25us, the measurement is accurate to +/- .25us and the falling-edge-to-falling-edge time (or rising-edge-to-falling-edge) must range from 3us to 16ms if Meas Cycles is 1, and 3us to 4.1sec if Meas Cycles > 1. Otherwise, if the Meas Resol popup is set to 1ms, the measurement is accurate to +/- 1ms and the falling-edge-to-falling-edge time (or rising-edge-to-falling-edge) must range from 3ms to 32 secs if Meas Cycles is 1, and 3ms to 16Ksec if Meas Cycles > 1.

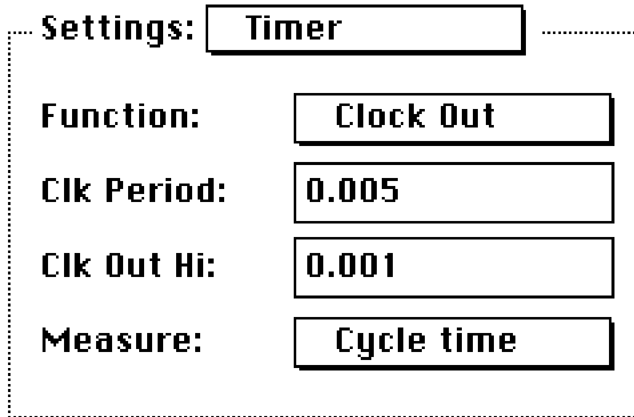
Table 2.2 shows which Fields are used for each of the different functions. instruNet ignores settings not needed for a particular function. For example if Digital Input or Digital Output is selected in the Function popup, instruNet ignores all other Fields in the Timer area.

Field	Digital In	Digital Out	Clock Out	Period Meas.
Clk Period	not used	not used	used	not used
Clk Out High	not used	not used	used	not used
Measure	not used	not used	not used	used
Meas. Resol.	not used	not used	not used	used
Meas. Cycles	not used	not used	not used	used

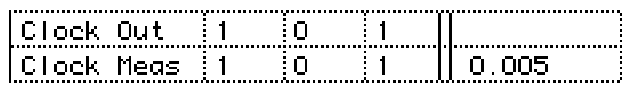
Table 2.2 Fields used with different Controller Digital I/O Functions.

We will now set up one channel as a clock output and measure its duration with another channel. This will require a 34 wire screw-terminal block cabled to the digital connector on a Model 200 instruNet Controller board. If you do not have a screw terminal block wired to your Controller, you can still do this experiment, yet the measured period will not be correct.

- Connect a physical wire between Pin #6 (Ch1 Dout) and Pin #7 (Ch2 Din) of the Controller Digital I/O connector.



- Set the Function popup to Clock Out, the Clk Period field to .005, and the Clk Out Hi field to .001.
- Select General in the Settings popup and set the User Name field to "Clock Out".
- Click the Enter button in the bottom right of the Probe Dialog to start the clock output.
- Select Ch2 Timer in the upper-right Channel popup to select the 2nd channel in you instruNet Controller.
- Set the User Name field to "Clock Meas".
- Select Timer in the Settings popup.
- Set the Function popup to Period Meas, set the Resolution popup to .25µs, set the Measure popup to Cycle Time, and set the Meas Cycles field to 1; as shown in figure 2.25.
- Press OK to return to the Network page.



Notice how the names of your first two Controller channels have changed to "Clock Out" and "Clock Meas", and how Clock Meas's Value field is displaying .005 seconds of measured period. This is the period of the clock signal being output by Ch1 Timer and being measured by Ch2 Timer. If you remove the wire connecting the Output of Ch1 Timer and the input of Ch2 Timer, the Value field will change.

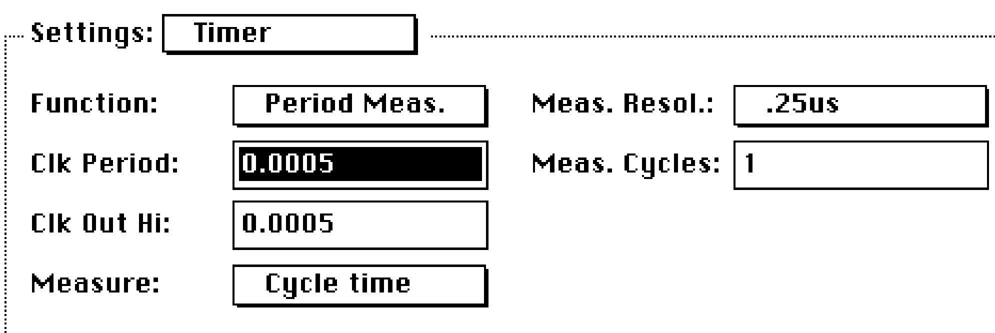


Figure 2.25 Timer Settings for Period Measurement

High speed >1Ks/sec/ch digitizing of analog signals by the instruNet Controller ties up the i2x0 DSP processor, and therefore inhibits it from doing the following during that time: pulse counter, output a specific number of clock pulses, period measurement, and frequency measurement.

For more details on the counter timer features, please see Application Notes 96, 110, 181, 182,

Working With The Controller Time Since Reset Channel

The instruNet Controllers offer a channel that provides the time, accurate to .25us, since the Controller was last reset. A reset occurs when instruNet is first used after power up, the computer resets, and when the Reset button is pressed in the Network page. This clock channel is called "Ch12 Time" and returns a number in units of seconds that is derived from a 62bit counter that counts 4MHz.

- Select the Network page by clicking the Network tab, and scroll down until you see the Controller's Ch12 Time channel. Notice how the Value cell slowly increments at a 1 second rate.

Working With Multiple Controllers

instruNet supports multiple controllers (i.e. networks) in one computer. In this case, the meaning of the "network number" becomes more important, since it selects a particular network in the data tree from which to operate. To digitize simultaneously from multiple controllers, one must select channels for digitizing in the Network page (any channel combination across all controllers), and then press the Start button in the Record page. There is one complexity however, which is that each controller has its own Trigger settings. This is because each controller operates independently (i.e. each has its own processor that manages its own data acquisition task). To access a Trigger dialog for a specific network, press the Trigger button in the Setup Dialog, and then select a specific network in the Network popup menu.

Setting the Trigger with multiple controllers is a little tricky since one controller cannot physically see the trigger signal attached to other controllers, and each controller has no way of sending messages to its colleagues in a short period of time. To make the digitizing from all networks trigger off the same signal, one must attach the trigger signal to one channel from each network, and set up the Trigger dialogs for each network to trigger off that one channel in each of their respective networks.

Triggering Off A Digital Input

To trigger off the instruNet 100 Ch25 Digital Input, specify this channel as the trigger source, attach your trigger signal to Ch25 DIO8, and set the trigger threshold to 200. When DIO8 is high (i.e. 2-5V), Ch25 is read as {128..255}; and when DIO8 is held low (i.e. 0-.8V) Ch25 is read as {0-127}.

Next Step

This concludes the tutorial. From here you might consider exploring the following areas: *Chapter 3, Sensor Reference*, which summarizes how to connect sensors; *Chapter 4, Programmers Tutorial*, which is a tutorial designed to get programmers up and running quickly.

Chapter 3, Connecting to Sensors

This chapter describes how to connect sensors to instruNet hardware and how to configure instruNet software for each type of sensor. instruNet allows different sensors to be connected directly to each input channel including RTD's, thermocouples, load cells, accelerometers, potentiometers, strain gages, resistance sources, current sources; and single-ended and differential voltage sources. In some cases, the user must add an external resistor to facilitate the reading of a sensor. instruNet supports a direct connection, returning native engineering units, to the following sensors:

Sensor	Units	Wiring	Constants	Figure #
Voltage	Volts	single-ended	None *(measure voltage between Vin & Gnd)*	3-1
Voltage	Volts	differential	none *(measure voltage between Vin+ & Vin-)*	3-2
Voltage	V/V	bridge	Vinit, Vout *(measure ratio of Vout/Vin)*	3-2b
Current	Amps	shunt resistor	Rshunt	3-3
Resistance	Ohms	voltage divider	Rshunt, Vout	3-4
Resistance	Ohms	bridge	Ro, Rshunt, Vout	3-5
Strain Gage	Strain	voltage divider	Ro,Shunt, Vout, Vinit, GF	3-6
Strain Gage	Strain	Q bridge	Ro, Vout, Vinit, GF, Rlead	3-7
Strain Gage	Strain	H bridge bend	Ro, Vout, Vinit, GF, Rlead	3-8
Strain Gage	Strain	H bridge axial	Ro, Vout, Vinit, GF, Rlead, v_Poisson	3-9
Strain Gage	Strain	F bridge bend	Vout, Vinit, GF	3-10
Strain Gage	Strain	F bridge axial I	Vout, Vinit, GF, v_Poisson	3-11
Strain Gage	Strain	F bridge axial II	Vout, Vinit, GF, v_Poisson	3-12
RTD	Celsius	voltage divider	alpha, delta, Ro, Shunt, Vout	3-13
RTD	Celsius	bridge	alpha, delta, Vout, Vinit, Ro	3-14
J, K, T, E, R, S, B, & N Thermocouple	Celsius	differential	none	3-15
Thermistor	Celsius	voltage divider	Ro, Rshunt, Vexe, Vinit, alpha, delta, GF, v_Poisson	3-16
Load Cell	Kg, LBs	bridge	Ro, Vexe, Vinit, GF	3-17
Accelerometer	G	single-endeded or diff.	Vinit, GF	3-18
Potentiometer	0 to 1	bridge	Ro, Vexe, Vinit	3-19
Quadrature Sensor	Counts	Please see Application Note #110		

Table 3.1, Sensors that can directly be connected to instruNet

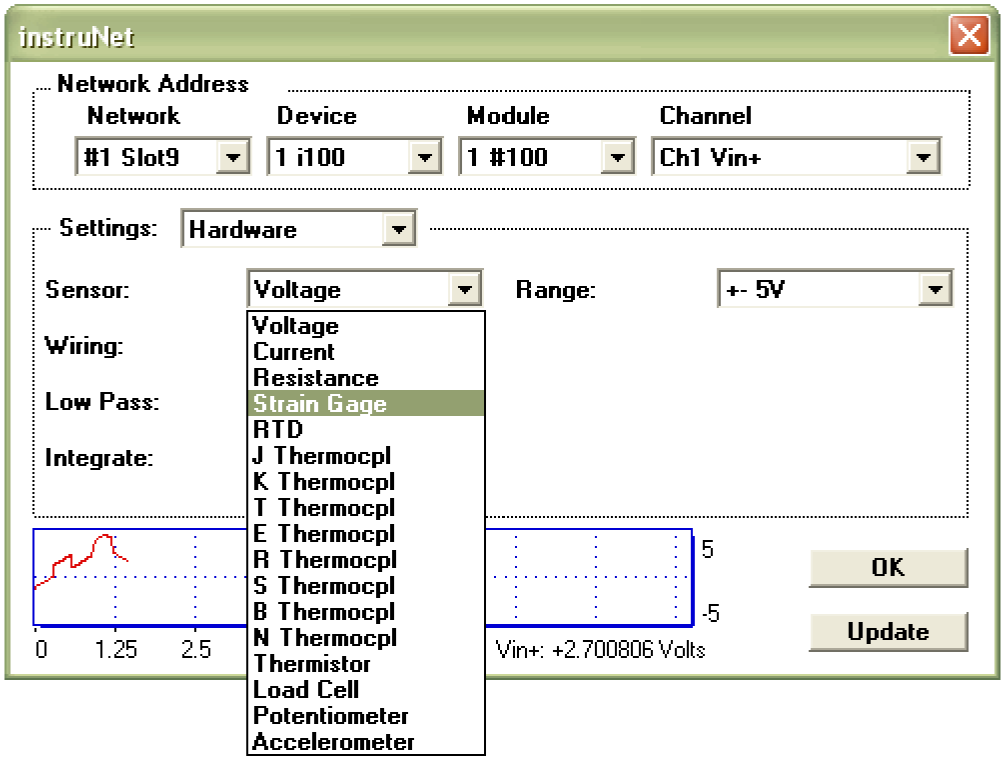
Connecting a Sensor Directly to *instruNet*

The instruNet voltage input terminals (often labeled "Vin(+)" or "Vin(-)") can directly connect to a variety of sensors, with instruNet software returning values in native engineering units (e.g. degrees Celsius, strain, Volts, Amps, ohms). This requires wiring the sensor to the instruNet hardware in the appropriate manner, and then configuring the instruNet software for your particular sensor, as described in the following steps:

1) Physically Wire your Sensor to instruNet hardware
The first step is to physically wire your sensor to the instruNet hardware, per the Hardware Reference guidelines at the end of this chapter. Each sensor listed in the previous table is discussed in detail in this reference. For example, to connect a J Thermocouple, one would attach the positive lead of the thermocouple to the instruNet Vin+ terminal, and the negative lead to the Vin- terminal. The input terminals are protected against over voltage and electrostatic discharge, therefore it is not always necessary to power off network devices (e.g. Model 100) while wiring sensors, although turning power off is recommended as a good safety practice. If your sensor does not fit into one of the standard categories in the previous table, then you need to choose the closest category, and then do the calculations necessary to resolve your desired engineering units. If you do power OFF your network, please use the following sequence:

1. Power OFF powered devices connected to the Network
2. Power OFF Computer
3. Configure network cables, sensors, and devices
4. Power ON Computer
5. Power ON powered devices connected to the Network

2) Tell the instruNet software which Sensor is connected
Set the <u>Sensor</u> popup in the <u>Hardware</u> settings area of the Probe dialog to the correct Sensor (e.g. Volts, J Thermocouple, etc). The sensors to choose from are listed in the first column of the Sensor Hook-up Table, at the beginning of this chapter.



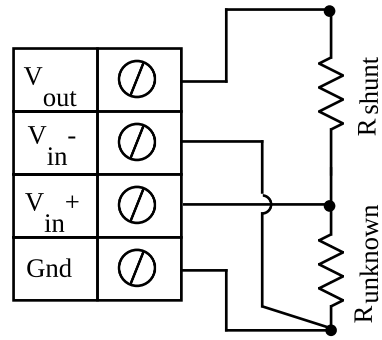
To set the Sensor popup, one would:

1. Open the instruNet World window by launching the Windows "`instruNet World Win32.exe`" or Macintosh "`instruNet World Mac`" application program (e.g. double-click on its icon).
2. Click on the Network tab at the bottom of the window to select the network page.
3. Click on the name of your channel in the left-most column to open the Probe dialog.
4. Select Hardware in the Settings popup menu.
5. Select the desired sensor in the Sensor popup menu, as illustrated in the previous figure.

3) Tell the instruNet software how the Sensor is wired

Set the Wiring popup in the Hardware settings area of the Probe dialog to the correct Wiring (e.g. Single-Ended, Differential, Bridge, Voltage Divider, etc). Please consult the Sensor Hook-up Table, at the beginning of this chapter, for guidance. Recall that the chosen wiring popup must match your actual physical hardware wiring.

4) Set the appropriate constants as required



Set the fields in the Constants settings area as needed. This involves selecting Constants in the Setting popup, and then setting specific Constants fields as noted in the previous table. For details on specific sensors, please consult the Sensor Reference at the end of this chapter. For example, to measure a resistance instruNet needs to know the value of an external shunt resistor "Rshunt", in ohms, and the value of the excitation voltage "Vout", in volts, as illustrated to the right. This would involve setting the Rshunt and Vout Constants fields to correspond to your actual wiring. instruNet software would then automatically return an amplitude in "ohms" units after measuring the voltage at the Vout terminal, measuring the voltage between the Vin+ and Vin- terminals, and calculating the realtime $R_{unknown}$.

5) Select the appropriate input voltage range
Set the Range popup in the Hardware Settings area as needed. This selects a voltage range that is used by instruNet. If the voltage exceeds a bound, then the bound is returned by the software. For example, if you input 2V and the range is +/- 1V then instruNet will return 1Volt. The resolution and accuracy of the measured signal increases when the range is reduced. For example, with the Model 100, the voltage accuracy is 15uV in the +/-10mV range, and 1.5mV in the +/- 5V range with 1ms of integration. Some sensors require a specific voltage range, or only allow one range. For example, the range with thermocouples is always in the neighborhood of +/- 100mV.

6) Set the Integrate popup
Set the Integrate popup in the Hardware Settings area as needed. This selects the duration that the signal is averaged before instruNet returns one number. For example, if you choose 16.666ms (one 60Hz line cycle), it will return the average signal value over a period of 16.666ms. This is helpful at reducing noise for signals acquired at slow sample rates (e.g. to integrate 16ms worth of data the sample rate must be greater than 16ms per point, or slower that 60samples/second). Each hardware device offer different Integrate options.

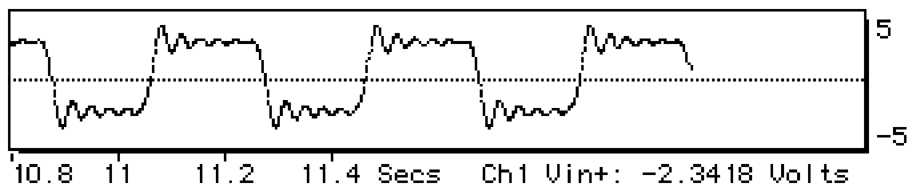
7) Select the appropriate Analog Filter
Set the analog Low Pass filter popup in the Hardware Settings area as needed (e.g. off, 40Hz, 4KHz). Each hardware device offers different analog low pass filter options as described in *Ch 8, Hardware Reference* (not all devices support hardware analog filtering). Low pass filters cause high frequencies to be rejected, while low frequencies are passed. Visually, the signal becomes "smoother". To see the effects of various filters, view the digitized signal at the bottom of the probe dialog, after making different selections (you might not be able to do this until you are further along in the set up process).

8) Set your Digital filters as needed
If you need to do a digital low pass, high pass, band pass, or band stop filter on your digitized waveform, please set the fields in the filter Settings areas (e.g. Low

Pass, High Pass) as desired. Digital filters are run on the waveform post acquisition, and only effect digitized waveforms (not single values read by instruNet). Please refer to Ch2 Tutorial, Setting Digital Filters for an expanded discussion of this powerful feature.

9) Consult the Sensor Reference for specifics
Carefully implement the Sensor Reference instructions for your particular sensor, later in this chapter, and read the corresponding footnotes at the end of this chapter. Different sensors require different range, filter, and constants settings; and care should be taken to insure accurate results.

10) Check your Work
To check your work, view the incoming signal, in realtime, at the bottom of the Probe dialog.



This display shows the sensor's value, in realtime, in native engineering units (e.g. Volts, Amps, degrees Celsius, strain) based on your software settings and external hardware wiring. The numerical value displayed at the bottom right is the actual real time value being read by instruNet. The plot shows the digitized version of your sensor value vs. time. The horizontal scale of this display is determined by the Pts Per Scan, No. of Scans, Scan Mode, Horiz Scale, and Sample Rate fields within the global Setup dialog (i.e. press the Record tab at the bottom of the instruNet World window, and then press the Setup button at the top). For details on how these work, please consult "Ch2, Working with the Voltage Inputs". To adjust the vertical scale of the probe display, select Display in the Setting popup, and then set the Display Max and Display Min fields to the desired engineering units values that correspond to the top and bottom of the display (e.g. set 1 and -1 to view a signal that varies from -1V to 1V). Press the Enter button to cause the new settings to take effect, and then view the updated display.

11) If your Sensor is not working, Fix it !

Listed below are several debugging hints for channels that are returning "bad" results:

a. Check that you have the correct channel
To check that you have the correct channel, and that it is being digitized by the software, view the probe dialog display as you disconnect one wire from your sensor. The value printed in the dialog lower right corner should change, to indicate that your displayed value is in some way connected to that wire. Also, if you touch the input terminal with your finger, you should notice some slight variation since your body acts as an antenna and causes radio stations, and such, to drive the input terminal (due to its high input impedance).

b. Check that instruNet is correctly measuring the voltage
At any time, you can set the Sensor popup to Volts, and set the Wiring popup to Vin+ - Vin- to cause instruNet to measure the voltage between the Vin+ and Vin- terminals. The measured voltage appears in the Probe dialog lower right. You can then check this against a volt meter that is placed in parallel between the Vin+ and Vin- terminals. Remember to set the Range popup if your signal is "clipping" on a bound. To measure the voltage between your Vin screw terminal (Vin+ or Vin-) and ground, you would do the same thing, yet select Vin - Gnd in the Wiring popup.

c. Check the engineering units calculations
If your voltage looks good, yet the returned engineering units value looks bad, then pull out a calculator (or better yet, a spreadsheet) and run your constants and known values into the equations listed in the Sensor Reference to check the processing of these numbers. Perhaps one of the fields in the Constants settings area is not set correctly.

d. Make sure you do not exceed the maximum input voltage
Make sure you are not exceeding the maximum input voltage, with respect to the Gnd terminal, as specified in "Ch8 Hardware Reference". To check this, measure the voltage between the Vin+ (or Vin-) terminal and the Gnd terminal with a Volt Meter. For example, in the Model 100, this voltage must not be less than -5V or greater than +5V. Exceeding a maximum typically does not cause damage unless it is very large (e.g. greater than +/- 20V).

e. Check your ground connections
If the ground between the instruNet device and your signal source is unstable, then connecting a wire between the instruNet Gnd terminal, and your signal source ground might help (e.g. attach a wire, or 1Kohm resistor, between the GND and Vin(-) terminals). Alternatively, if the instruNet Gnd is tied to the ground of your signal source, and these are at different voltages with respect to Earth ground, then current will flow between them. This current will cause voltage drops and subsequently may induce weird effects -- breaking this ground connection might help. The best way to determine what helps is to try different things (e.g. hold a wire between two grounds) and observe what happens in the display at the bottom of the Probe dialog.

g. Recheck your work
Recheck your hardware wiring and software settings.

12) Save your work.
To save the current configuration of instruNet (i.e. the settings displayed in the Network page and accessed via the Probe dialog), press the Network tab at the bottom of the window to select the Network page, and then press the Save button at the top of the window to save the settings to disk. A file save dialog appears, and it is here that you must specify a file name and file location (remember where you put it).

To check your saved settings: exit instruNet world, re-enter instruNet (e.g. launch Windows "`instruNet World Win32.exe`" or Macintosh "`instruNet World Mac`"), press the Network tab at the bottom of the window to select the Network page, press the Open button at the top of the window, select your saved settings file (this will load your save settings), click on the channel that you just set up, and then view the realtime display at the bottom of the Probe dialog.

Alternatively, one could press the Store button to save the settings directly to a preferences file in the operating system folder, and then press the Restore button, at a later date, to restore them. The advantage of Save/Restore is the user does not need to specify a file name or file location; whereas the disadvantage is the saved file is overwritten the next time someone presses Store.

Sensor Reference

instruNet connects directly to a variety of sensors and returns engineering units, as summarized in the following pages. Each sensor must be wired exactly as shown in the following figures, and the software must be configured exactly as described in the adjacent instructions.

Single-ended Voltage Measurement

Single-Ended Voltage measurement involves reading a voltage between the Vin+ (or Vin-) instruNet input terminal and the Gnd input terminal, as illustrated in the figure to the right. The Gnd terminal is typically tied to earth ground through the user's cable, the instruNet network cable, or an external power supply cable. Most amplifiers that supply a single-ended output signal have their grounds tied to earth ground via the power supply cable. instruNet channels, configured for "Voltage" measurement, return a value in units of Volts.

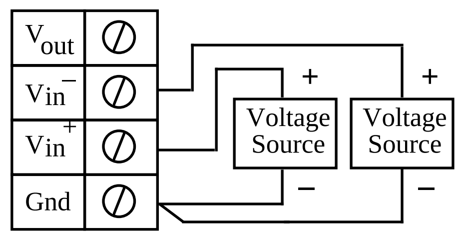


Fig 3.1 - Singled-Ended Voltage Measurement Mode with Two Signal Sources.

The Vin+ and the Vin- screw terminals function identically when used to read single-ended voltages (e.g. Ch1 corresponds to Vin+, and Ch2 corresponds to Vin- on the Model 100).

To do a Single-Ended Voltage measurement you must:

1. Set the <u>Sensor</u> field in the <u>Hardware</u> settings area to <u>Voltage</u>.
2. Set the <u>Wiring</u> field in the <u>Hardware</u> settings area to <u>Vin - GND</u>.
3. Wire your voltage source per figure 3.1, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.
4. For information on measuring voltages in excess of +-5V, please see Application Note #82 Measuring Large Voltages.

Differential Voltage Measurement

Differential Voltage measurement involves reading a voltage between a pair of instruNet Vin+ and Vin- input terminals, as illustrated in the figure to the right. The Gnd terminal is not used when connecting a differential voltage source to instruNet; however, you must be careful to insure that the voltage applied to the Vin+ or Vin- terminal does not exceed the maximums specified in "Ch8 Hardware Reference" (e.g. all Vin terminals on the Model 100 must be kept between -5V and +5V, with respect to the Gnd terminal, in order to assure accurate readings). Differential mode is preferable for applications involving significant amounts of low frequency (< 5kHz) common mode noise that might result from long signal cables. instruNet channels, configured for "Voltage" measurement, return a value in units of Volts.

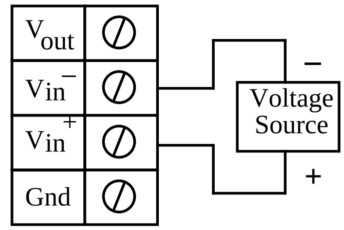


Fig 3.2 - Differential Voltage Measurement with One Signal Source.

To do a Differential Voltage measurement you must:

1. Set the <u>Sensor</u> field in the <u>Hardware</u> settings area to <u>Voltage</u>.
2. Set the <u>Wiring</u> field in the <u>Hardware</u> settings area to <u>Vin+ - Vin-</u>.
3. Wire your voltage source per figure 3.2, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.
4. If your measured signal is noisy, try connecting a wire (or 1Kohm resistor) between the Vin- and GND terminals (to reduce common mode noise), and refer to footnote #5 at the end of this chapter for details on how to low pass filter the measured signal.
5. For information on measuring voltages in excess of +-5V, please see Application Note #82 Measuring Large Voltages.

Bridge Ratio Voltage Measurement

Bridge Ratio Voltage measurement involves measuring the ratio of the voltage measured across a device (e.g. a bridge) to the excitation voltage applied to the device, as illustrated in the figure to the right. This involves applying a voltage across the device and measuring the voltage across the two intermediate nodes via a pair of instruNet Vin+ and Vin- input terminals. The excitation voltage for the bridge is supplied by either the instruNet Vout terminal or by an external voltage source. instruNet calculates the ratio, returning "V/V" engineering units, using the equations:

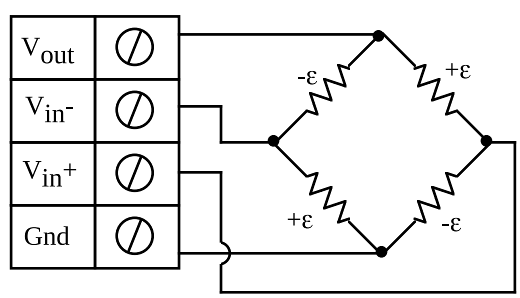


Fig 3.2b- Bridge Ratio Voltage Measurement

$$V_{ratio}\ (V/V) = ((V_{in}+ - Vin-) - Vinit) / V_{out}$$

Vinit and Vout are fixed values that are specified by the user in the Constants Settings area, whereas (Vin+ - Vin-) are measured in realtime by instruNet. For more details, please see Application Notes #14 Strain Gages, #75 Load Cells, and #139 Ratio Measurements.

To do Voltage ratio measurement you must:

1. Set the Sensor field in the Hardware settings area to Voltage.
2. Set the Wiring field in the Hardware settings area to Bridge .
3. Set the Ro field in the Constants settings area to the value of one Ro bridge resistor (i.e. or approx resistance across the device), in ohms units.[1,3]
4. Set the Vout field in the Constants settings area to specify an excitation voltage to be applied to the bridge. If you are applying an external excitation voltage, enter -Ro value in the Ro edit field (e.g. -100 instead of 100ohms) to tell the software that the excitation is external, and then enter the external excitation voltage in the Vout.[2] In high current cases (e.g. >2mA), it is often helpful to alternate the polarity of the excitation voltages to evenly burden the +/-12V supplies.[11]
5. Set the Vinit field in the Constants settings area to the voltage measured when the bridge is not stimulated, in Volts units (to null the bridge).[8]
6. Wire your voltage source per figure 3.2b, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10]
7. To eliminate susceptibility to RFI noise, it may be necessary to place a capacitor between the Vin+ and Vin- screw terminals, and between the Vin- and Gnd screw terminals, as noted in Application #14 Strain Gages. Typical values are .01uF.

Current Measurement

Current measurement involves reading the voltage across an external user supplied shunt resistor, to which a current source is connected, as illustrated to the right. The voltage is measured between a pair of instruNet Vin+ and Vin- input terminals. instruNet then calculates the current through the shunt resistor using the following equation, and returns a value in "Amps" units:

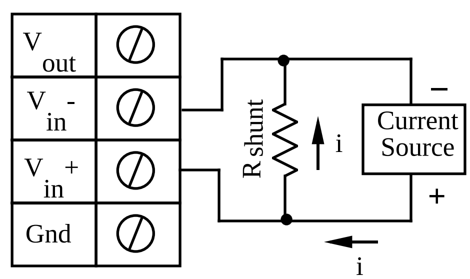


Fig 3.3 - Current Measurement.

$$\text{current (Amps)} = (V_{in+} - V_{in-}) / R_{shunt}$$

Shunt resistor values are typically chosen to cause a large voltage (several volts maximum) to be measured by instruNet, without saturating the current source (i.e. exceeding its maximum output voltage), and without heating up the resistor significantly to cause its resistance to change. If the Rshunt value is low, then the voltage across it is low, and this decreases the signal to noise ratio of the measured signal. Also, Rshunt must be selected such that the voltage across it does not exceed the instruNet maximum input voltage (e.g. +/-5V with the Model 100). Due to these limitations, instruNet might not let you set some of the fields too high or too low.

To do a Current measurement you must:

1. Set the Sensor field in the Hardware settings area to Current.
2. Set the Wiring field in the Hardware settings area to Shunt Resistor .
3. Set the Rshunt field in the Constants settings area to the value of your external user supplied Rshunt resistor, in ohms units.[1,3]
4. Set the Voltage Range field in the Hardware settings area to something similar to the maximum expected current * Rshunt.
5. Wire your current source per figure 3.3, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.
6. If your measured signal is noisy refer to footnote #5 at the end of this chapter for details on how to low pass filter the measured signal; also, if your current source is fully isolated from GND, connecting a 1K ohm (or wire) between the Vin- and GND terminals might reduce common mode noise.
8. For more details, please see Application Note #82 Measuring Large Currents.

Resistance Measurement: Voltage Divider Circuit

Resistance measurement using a voltage divider involves connecting a resistor of unknown value in series with an external user supplied shunt resistor of known value, applying a voltage across the divider circuit, and measuring the voltage across Runknown, as illustrated to the right. The voltage across Runknown is measured between a pair of instruNet Vin+ and Vin- input terminals while the excitation voltage is supplied by the instruNet Vout terminal. instruNet then calculates the value of Runknown using the following equation, and returns a value in "ohms" units:

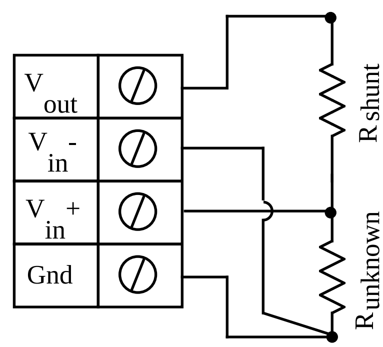


Fig 3.4 - Resistance Measurement - Voltage Divider Method

$$R_{unknown}\,(\text{Ohms}) = R_{shunt} * (V_{in+} - V_{in-}) / ((V_{in+} - V_{in-}) - V_{out})$$

To do a Resistance measurement using a Voltage Divider, you must:

1. Set the Sensor field in the Hardware settings area to Resistance.
2. Set the Wiring field in the Hardware settings area to Voltage Divider .
3. Set the Rshunt field in the Constants settings area to the value of your external user supplied Rshunt resistor, in ohms units.[1,3,6]

4. Set the Vout field in the Constants settings area to specify the excitation voltage that is to be applied to the divider. In high current cases (e.g. >2mA), it is often helpful to alternate the polarity of the excitation voltages to evenly burden the +/-12V supplies.[11]
5. Set the Voltage Range field in the Hardware settings area to something similar to Vout * (Runknown_Maximum / (Runknown_Maximum + Rshunt)).
6. Wire your voltage source per figure 3.4, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[10]

Resistance Measurement: Bridge Circuit

Resistance measurement using a bridge circuit involves connecting a resistor of unknown value as one leg of a full-bridge circuit, applying a voltage across the bridge, and measuring the voltage across the two intermediate nodes. The intermediate node voltage is measured between a pair of instruNet Vin+ and Vin- input terminals while the bridge excitation voltage is supplied by either the instruNet Vout terminal or an external voltage source. In figure 3.5, Runknown is a resistor whose value is being measured and Ro is a similar valued resistor of known value[4]. This technique is only accurate if Runknown stays in the range of Ro, +/- 50%. If you need to measure a resistance with more range, please use the Resistance Measurement using a Voltage Divider, described earlier. instruNet calculates the value of Runknown using the following equation, and returns a value in "ohms" units:

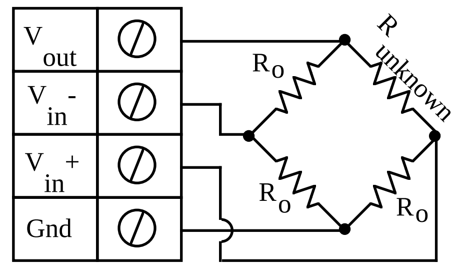


Fig 3.5 - Resistance Measurement - Bridge Circuit

$$\text{Runknown (ohms)} = R_o * (V_{out} - 2.0 * (V_{in+} - V_{in-})) / (V_{out} + 2.0*(V_{in+} - V_{in-}))$$

To do a Resistance measurement using a Bridge circuit you must:

1. Set the Sensor field in the Hardware settings area to Resistance.
2. Set the Wiring field in the Hardware settings area to Bridge .
3. Set the Ro field in the Constants settings area to the value of one Ro bridge completion resistor, in ohms units.[1,3,4]
4. Set the Vout field in the Constants settings area to specify an excitation voltage to be applied to the bridge. If you are applying an external excitation voltage, enter -Ro value in the Ro edit field (e.g. -100 instead of 100ohms) to tell the software that the excitation is external, and then enter the external excitation voltage in the Vout. In high current cases (e.g. >2mA), it is often helpful to alternate the polarity of the excitation voltages to evenly burden the +/-12V supplies.[11]
5. Wire your voltage source per figure 3.5, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10]

Strain Gage Measurement: Voltage Divider Circuit

Strain measurement using a voltage divider circuit involves connecting a strain gage in series with a shunt resistor of known value, applying a voltage across the pair and measuring the voltage across the strain gage, as illustrated to the right. The voltage across the strain gage is measured between a pair of instruNet Vin+ and Vin- input terminals while the excitation voltage for the divider is supplied by the instruNet Vout terminal. instruNet calculates the value of the strain using the following equations, and returns "strain" engineering units.

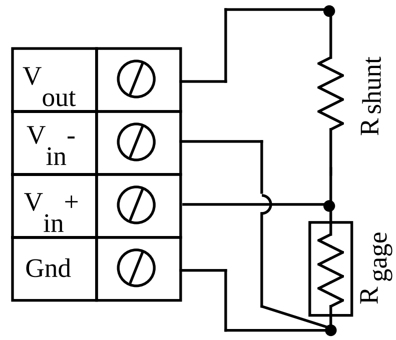


Fig 3.6- Strain Gage Measurement - Voltage Divider Circuit

R_{gage} (Ohms) = R_{shunt} * (Vin+ - Vin-) / (V_{out} - (Vin+ - Vin-))

Strain = ((1 / (GF * Ro)) * (R_{gage} - Ro))

Ro is the resistance of the unstrained strain gage and Rgage is the resistance of the gage when strained. Voltage divider circuits are less accurate than bridge circuits when measuring small resistance changes (which are typical in strain gage measurements), but are easier and less expensive to build. For strain gage measurements, a bridge circuit is highly recommended over a voltage divider.

To do a strain gage measurement using a Voltage Divider, you must:

1. Set the Sensor field in the Hardware settings area to Strain Gage.
2. Set the Wiring field in the Hardware settings area to Voltage Divider .
3. Set the Voltage Range field in the Hardware settings area to something similar to Vout * (Ro / (Rshunt + Ro)).
4. Set the Rshunt field in the Constants settings area to the value of your external user supplied Rshunt resistor, in ohms units.[1,3,6]
5. Set the Ro field in the Constants settings area to the value of your gage when unstrained, in ohms units.
6. Set the GF field in the Constants settings area to the gage's gage factor.[9]
7. Set the Vout field in the Constants settings area to specify the excitation voltage that is to be applied to the divider.[6] In high current cases (e.g. >2mA), it is often helpful to alternate the polarity of the excitation voltages to evenly burden the +/-12V supplies.[11]
8. Wire your strain gage per figure 3.6, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10]

Strain Gage Measurement - Quarter Bridge

Strain measurement using a 1/4 bridge circuit involves wiring a strain gage as one leg of a full-bridge circuit, applying a voltage across the bridge, and measuring the voltage across the two intermediate bridge nodes via a pair of instruNet Vin+ and Vin- input terminals. The excitation voltage for the bridge is supplied by either the instruNet Vout terminal or by an external voltage source. In figure 3.7, R_{gage} is a strain gage, R_o is either a fixed resistor of known value or a fixed unstrained strain gage of value R_o, and R_L is the lead wire resistance. instruNet calculates the value of the strain, returning "strain" engineering units, using the equations:

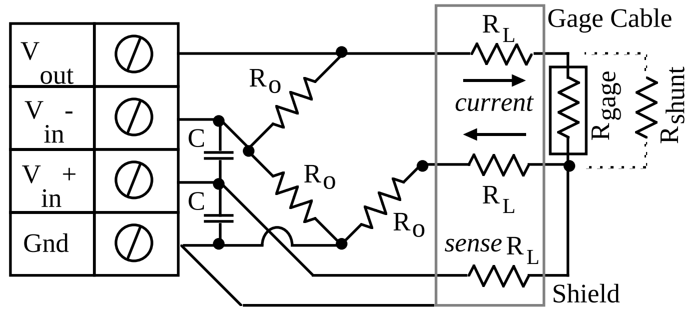


Fig 3.7 - Strain Measurement - Quarter Bridge Circuit

$$V_{ratio}\ (V/V) = ((V_{in}+ - Vin-) - Vinit) / V_{out}$$

$$Strain = (-4V_{ratio} / [GF * (1 + 2V_{ratio})]) * (1 + R_L / R_o)$$

Ro, R_L, GF, Vinit and Vout are fixed values that are specified by the user in the Constants Settings area, whereas (V_{in}+ - Vin-) are measured in realtime by instruNet. R_o and Unstrained-R_{gage} must be the same value (e.g. 350ohms) in order for the bridge to operate properly.[4] For more details, please see Application Notes #14 Strain Gages, #75 Load Cells, and #139 Ratio Measurements.

To do Strain Gage measurement using a 1/4 Bridge circuit you must:

1. Set the Sensor field in the Hardware settings area to Strain Gage.
2. Set the Range field in the Hardware settings area to +/- 10mV.[7]
3. Set the Ro field in the Constants settings area to the value of one Ro bridge completion resistor, in ohms units .[3,4]
4. Set the GF field in the Constants settings area to the gage's gage factor.[9]
5. Set the Vout field in the Constants settings area to specify the voltage that is to be applied to the bridge (1V is typical). If you are applying an external excitation voltage, enter -Ro value in the Ro edit field (e.g. -100 instead of 100 ohms) to tell the software that the excitation is external, and then enter the value of the external excitation voltage into the Vout field (e.g. 4V).[2] In high current cases (e.g. >2mA), it is often helpful to alternate the polarity of the excitation voltages to evenly burden the +/-12V supplies.[11]
6. Set the delta, Rlead field in the Constants settings area to the resistance, in ohms, of the wires leading to the bridge (0 ohms is typically ok).
7. Set the Vinit field in the Constants settings area to the voltage measured when the bridge is unstrained, in Volts units.[8]
8. Set the Wiring field in the Hardware settings area to Q Bridge.

9. Capacitors across the voltage input terminals are highly recommended for reducing errors caused by RFI. With 350ohm gages, 0.1uF caps create a low pass filter at 4KHz [4K = 1 / (6.28 * 350 * 0.1e-6)], and are ideal at minimizing RFI effects.
10. Wire your voltage source per figure 3.7, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10]

Strain Gage Measurement - Half Bridge (Bending)

Measuring bending strain using a 1/2 bridge configuration involves wiring two strain gages as shown in figure 3.8, applying a voltage across the bridge, and measuring the voltage across the two intermediate bridge nodes via a pair of instruNet Vin+ and Vin- input terminals. The excitation voltage for the bridge is supplied by either the instruNet Vout terminal or by an external voltage source. In figure 3.8, R_{gage} is a strain gage, R_o is either a fixed resistor of known value or a fixed unstrained strain gage of value R_o, and R_L is the lead wire resistance. instruNet calculates the value of the strain, returning "strain" engineering units, using the equations:

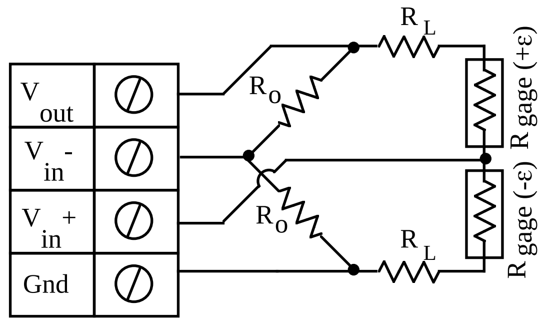


Fig 3.8 - Strain Measurement - Half Bridge Circuit

V_{ratio} (V/V) = ((V_{in}+ - Vin-) - Vinit) / V_{out}

Strain = (-2V_{ratio} / GF) * (1 + R_L /R_o)

Ro, R_L, GF, Vinit and Vout are fixed values that are specified by the user in the Constants Settings area, whereas (V_{in}+ - Vin-) are measured in realtime by instruNet. R_o and Unstrained-R_{gage} must be the same value (e.g. 350ohms) in order for the bridge to operate properly[4].

To do a bending Strain Gage measurement using a 1/2 Bridge circuit you must:

1. Do steps #1 through #7 listed in the previous "Strain Gage Measurement - Quarter Bridge" discussion.
2. Set the Wiring field in the Hardware settings area to H Bridge Bend .
3. Wire your voltage source per figure 3.8, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10]

Strain Gage Measurement - Half Bridge (Axial)

Measuring axial strain using a 1/2 bridge configuration involves wiring two strain gages as shown in figure 3.9, applying a voltage across the bridge, and measuring the voltage across the two intermediate bridge nodes via a pair of instruNet Vin+ and Vin- input terminals. The excitation voltage for the bridge is supplied by either the instruNet Vout terminal or by an external voltage source. In figure 3.9, R_{gage} is a strain gage, R_o is either a fixed resistor of known value or a fixed unstrained strain gage of value R_o, and R_L is the lead wire resistance. instruNet calculates the value of the strain, returning "strain" engineering units, using the equations:

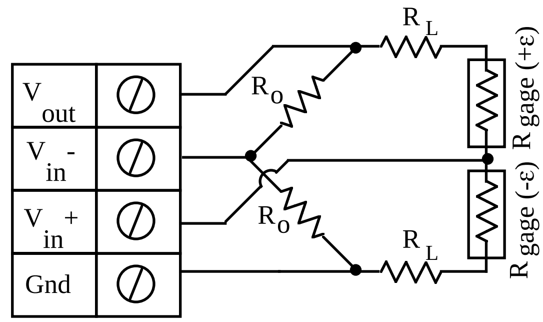


Fig 3.9 - Strain Measurement - Half Bridge Circuit (Axial)

$$V_{ratio}\ (V/V) = ((V_{in}+ - Vin-) - Vinit) / V_{out}$$

$$Strain = (-4V_{ratio} / [GF * [(1 + \upsilon) - 2V_{ratio}\ (\upsilon - 1)]]) * (1 + R_L / R_o)$$

υ, Ro, R_L, GF, Vinit and Vout are fixed values that are specified by the user in the Constants Settings area, whereas (V_{in}+ - Vin-) are measured in realtime by instruNet. R_o and Unstrained-R_{gage} must be the same value (e.g. 350ohms) in order for the bridge to operate properly[4].

To do axial Strain Gage measurement using a 1/2 Bridge circuit you must:

1. Do steps #1 through #7 listed in the previous "Strain Gage Measurement - Quarter Bridge" discussion.
2. Set the Wiring field in the Hardware settings area to H Bridge Axial .
3. Set the v_Poisson field in the Constants settings area to the υ poissson value of the material that you are twisting (e.g. aluminum is .32).
4. Wire your voltage source per figure 3.9, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10]

Strain Gage Measurement - Full Bridge (Bending)

Measuring bending strain using a full bridge configuration involves wiring four strain gages as shown in figure 3.10, applying a voltage across the bridge, and measuring the voltage across the two intermediate bridge nodes via a pair of instruNet Vin+ and Vin- input terminals. The excitation voltage for the bridge is supplied by either the instruNet Vout terminal or by an external voltage source. instruNet calculates the value of the strain, returning "strain" engineering units,
using the equations:

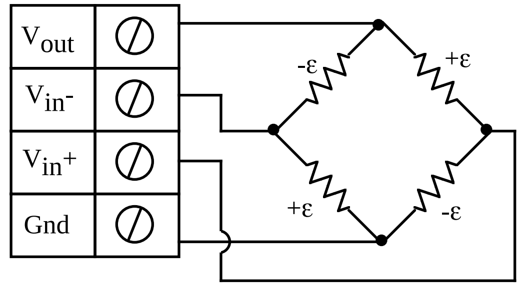


Fig 3.10 - Strain Measurement - Full Bridge Circuit (Bending)

V_{ratio} (V/V) = ((V_{in}+ - Vin-) - Vinit) / V_{out}

Strain = -V_{ratio} / GF

GF, Vinit and Vout are fixed values that are specified by the user in the <u>Constants</u> Settings area, whereas (Vin+ - Vin-) are measured in realtime by instruNet.

To do bending Strain Gage measurement using a Full Bridge circuit you must:

1. Do steps #1 through #7 listed in the previous "Strain Gage Measurement - Quarter Bridge" discussion.
2. Set the <u>Wiring</u> field in the <u>Hardware</u> settings area to <u>F Bridge Bend</u>.
3. Wire your voltage source per figure 3.10, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10]

Strain Gage Measurement - Full Bridge (Axial I)

Measuring axial strain using a full bridge configuration involves wiring four strain gages as shown in either figure 3.11 or figure 3.12, applying a voltage across the bridge, and measuring the voltage across the two intermediate bridge nodes via a pair of instruNet Vin+ and Vin- input terminals. The excitation voltage for the bridge is supplied by either the instruNet Vout terminal or by an external voltage source. instruNet calculates the value of the strain, returning "strain"
engineering units, using the equations:

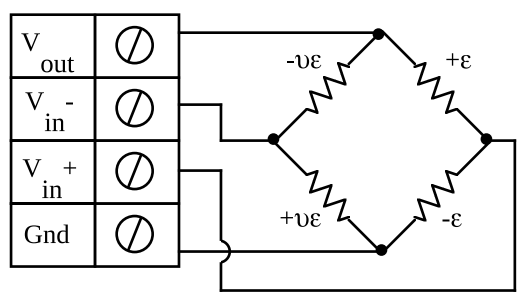


Fig 3.11 - Strain Measurement - Full Bridge Circuit (Axial I)

V_{ratio} (V/V) = ((V_{in}+ - Vin-) - Vinit) / V_{out}

Strain = -2V_{ratio} / GF (υ + 1)

υ, GF, Vinit and Vout are fixed values that are specified by the user in the Constants Settings area, whereas (Vin+ - Vin-) are measured in realtime by instruNet.

To do Axial Strain Gage measurement using a Full Bridge circuit you must:

1. Do steps #1 through #7 listed in the previous "Strain Gage Measurement - Quarter Bridge" discussion.
2. Set the Wiring field in the Hardware settings area to F Bridge Axl I.
3. Set the v_Poisson field in the Constants settings area to the υ poison value of the material that your are twisting (e.g. aluminum is .32).
4. Wire your voltage source per figure 3.11, and refer to the steps at the beginning of this chapter for more information on how to set up your

Strain Gage Measurement - Full Bridge (Axial II)

Measuring axial strain using a full bridge configuration involves wiring four strain gages as shown in figure 3.12, applying a voltage across the bridge, and measuring the voltage across the two intermediate bridge nodes via a pair of instruNet Vin+ and Vin- input terminals. The excitation voltage for the bridge is supplied by either the instruNet Vout terminal or by an external voltage source. instruNet calculates the value of the strain, returning "strain" engineering units, using the equations:

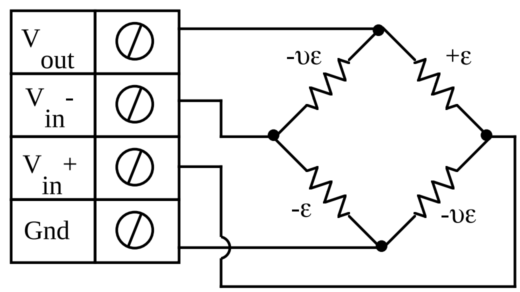


Fig 3.12 - Strain Measurement - Full Bridge Circuit (Axial II)

V_{ratio} (V/V) = ((V_{in}+ - Vin-) - Vinit) / V_{out}

Strain = -2V_{ratio} / [GF [(υ + 1) – V_{ratio} (υ – 1)]]

υ, GF, Vinit and Vout are fixed values that are specified by the user in the Constants Settings area, whereas (Vin+ - Vin-) are measured in realtime by instruNet.

To do Axial Strain Gage measurement using a Bridge circuit you must:

1. Do steps #1 through #6 listed in the previous "Strain Gage Measurement - Quarter Bridge" discussion.
2. Set the Wiring field in the Hardware settings area to F Bridge II .

3. Set the v_Poisson field in the Constants settings area to the υ poisson value of the material that your are twisting (e.g. aluminum is .32).
4. Wire your voltage source per figure 3.12, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10]

Temperature Measurement (RTD) Voltage Divider[1]

Temperature measurement using a voltage divider circuit involves connecting an RTD in series with a shunt resistor of known value, applying a voltage across the pair and measuring the voltage across the RTD, as illustrated to the right. The voltage across the RTD is measured between a pair of instruNet Vin+ and Vin- input terminals while the excitation voltage for the divider is supplied by the instruNet Vout terminal. instruNet calculates the value of the strain using the following equations, and returns "degrees C" engineering units.

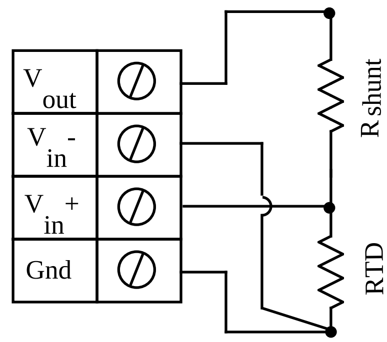


Fig 3.13 - Temperature Measurement (RTD) Voltage Divider Circuit

R_{RTD} (Ohms) = R_{shunt} * (Vin+ - Vin-) / (V_{out} - (Vin+ - Vin-))
a = Ro * alpha (1.0 + (delta / 100.0))
b = Ro * -1.0 * alpha * delta / (100.0 * 100.0)
c = Ro - R_{RTD}
Temperature (Celsius) = c / (-0.5 * (b + sqrt((b*b) - (4.0 * a * c))))

Alpha is the temperature coefficient of the RTD at 0C (typically .00385 for American RTD's, and .00392 for European RTD's) and delta is the Callendar-Van Dusen delta constant (typically 1.492). These constants are often supplied by the manufacturer of the RTD. The instruNet temperature linearizer only supports temperatures above 0°C.

Many RTD manufacturers recommend a 1mA RTD current source with RTD voltage dividers since this often dissipates several milliwatts, and therefore does not cause noticeable "self" heating. An example would be a 100 ohm RTD (which will vary from 25 to 400 ohms as the temperature varies; i.e. 25% to 400%), a 4.5V excitation voltage (i.e. Vout) and a 5000 ohm shunt resistor. The average current and power dissipation of the RTD at 0°C would be:

Current (Amps) = Volts / Resistance = 4.5V / [5000 + 100] = .88mA

Power (watts) = Current * Current * Resistance = .0088 * .0088 * 5100 = 3.8mW

The voltage across the RTD would vary from 22mV to 352mV as the resistance across the RTD changed from 25 to 400 ohms (corresponding to a temperature change of -260 to +850 Celsius); therefore, an input Voltage Range of ±600mV would be ideal with a 100 ohm RTD, 4.5V Vout voltage, and 5000 ohm resistor.

To do temperature measurement using an RTD in a voltage divider circuit you must:

1. Set the Sensor field in the Hardware settings area to RTD.
2. Set the Wiring field in the Hardware settings area to Voltage Divider .
3. Set the Voltage Range field in the Hardware settings area to something similar to Vout * (RTD_Max / (Rshunt + RTD_Max)), where RTD_Max is the RTD resistance at 0°C times 4.
4. Set the Rshunt field in the Constants settings area to R_{shunt}.[1,3,6]
5. Set the Ro field in the Constants settings area to the resistance of the RTD at 0°C, in ohms units.
6. Set the Vout field in the Constants settings area to specify the excitation voltage that is to be applied to the divider.   In high current cases (e.g. >2mA), it is often helpful to alternate the polarity of the excitation voltages to evenly burden the +/-12V supplies.[11]
7. Set the alpha field in the Constants settings area to the alpha value of your RTD.
8. Set the delta,Rlead field in the Constants settings area to the delta value of your RTD.
9. Wire your voltage source per figure 3.13, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor. To reduce noise, 0.001 seconds of integration is often helpful (i.e. set the Integrate field in the Hardware setting area to 0.001). [5, 10]

Temperature Measurement (RTD) Bridge Circuit

Temperature measurement using an RTD in a bridge circuit involves wiring an RTD as one leg of a full-bridge circuit, applying a voltage across the bridge, and measuring the voltage across the two intermediate bridge nodes via a pair of instruNet Vin+ and Vin- input terminals. The excitation voltage for the bridge is supplied by either the instruNet Vout terminal or by an external voltage source. In figure 3.14 R_{RTD} is an RTD and R_o is a resistor of known, similar, value. instruNet calculates the value of the temperature, returning degrees Celsius engineering units, using the equations:

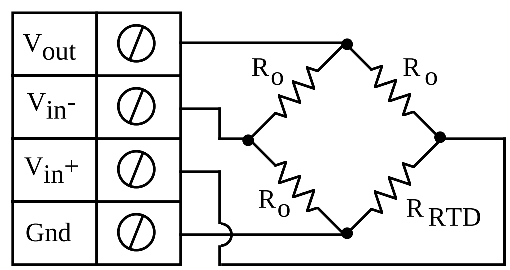


Fig 3.14 - Temperature Measurement (RTD) Bridge Circuit

R_{RTD} (Ohms) = (Ro * (Vout - 2.0 * (Vin+ - Vin-)) / (Vout + 2.0 * (Vin+ - Vin-))
a = Ro * alpha (1.0 + (delta / 100.0))
b = Ro * -1.0 * alpha * delta / (100.0 * 100.0)
c = Ro - R_{RTD}
Temperature (Celsius) = c / (-0.5 * (b + sqrt((b*b) - (4.0 * a * c))))

Alpha is the temperature coefficient of the RTD at 0C (typically .00385 for American RTD's, and .00392 for European RTD's) and delta is the Callendar-Van Dusen delta constant (typically 1.492). These constants are supplied by the manufacturer of the RTD. The instruNet temperature linearizer only supports temperatures above 0°C.

The RTD bridge circuit is very accurate when the RTD resistance is close to the Ro bridge resistance. (i.e. the RTD resistance varies between .5 * Ro and 2 * Ro). If you need more RTD temperature range, please use the RTD Voltage Divider circuit.

To do temperature measurement using an RTD in a bridge circuit you must:

1. Set the Sensor field in the Hardware settings area to RTD.
2. Set the Wiring field in the Hardware settings area to Bridge.
3. Set the Range field in the Hardware settings area to something small such as +/-100mV.
4. Set the Ro field in the Constants settings area to the resistance of the RTD at 0°C, in ohms (typically, this is also be the bridge completion resistor resistance).[3,4]
5. Set the alpha field in the Constants settings area to the alpha value of your RTD.
6. Set the delta,Rlead field in the Constants settings area to the delta value of your RTD.
7. Set the Vout field in the Constants settings area to specify the voltage that is to be applied to the bridge (1V is typical). If you are applying an external excitation voltage, enter -Ro value in the Ro edit field (e.g. -100 instead of 100 ohms) to tell the software that the excitation is external, and then enter the value of the external excitation voltage into the Vout field (e.g. 4V).[2] In high current cases (e.g. >2mA), it is often helpful to alternate the polarity of the excitation voltages to evenly burden the +/-12V supplies.[11]
8. Wire your voltage source per figure 3.14, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor. To reduce noise, 0.001 seconds of integration is often helpful (i.e. set the Integrate field in the Hardware setting area to 0.001).[5,10]

Temperature Measurement Thermocouple

Temperature measurement using a thermocouple involves connecting the two thermocouple leads to a pair of instruNet Vin+ and Vin- input terminals. If this voltage floats with respect to the instruNet Gnd terminal, it might be necessary to also attach the instruNet Vin- terminal to the instruNet Gnd terminal with a short wire. Sometimes this wire makes things better, and sometimes this wire makes things worse. instruNet calculates the value of the temperature in degrees Celsius using a polynomial linearizing equation.

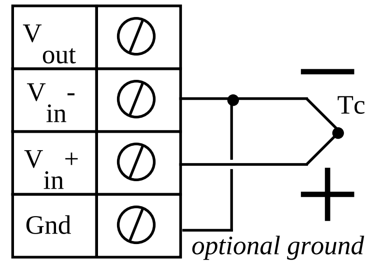


Fig 3.15 - Temperature Measurement (Thermocouple)

To do temperature measurement using a thermocouple you must:

1. Set the Sensor field in the Hardware settings area to the appropriate Thermocouple type (J, K, T, E, R, S).
2. Set the Wiring field in the Hardware settings area to Vin+ - Vin-.
3. Set the Range field in the Hardware settings to either +/- 80mV or +/- 10mV (approximately) depending on the temperature range being measured. If your temperature range is <170C for a type J thermocouple, <230C for a type K thermocouple, <180C for at type T thermocouple, <900C for a type R thermocouple, <900C for a type S thermocouple or between -250C to 140C for type E thermocouple, then set the range to approximately +/- 10mV. Otherwise set the Range for approximately +/- 80mV. Measurement resolution is approximately 1.2µV in the +/- 10mV range and approximately 10µV in the +/- 80mV range.
4. Wire your voltage source per figure 3.15, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10] If your thermocouple leads are backwards, then the measured temperature will be shown as varying in the opposite direction from ambient (e.g. if the instruNet terminals are at 25°C, the thermocouple is at 35°C, and the leads are backwards, then instruNet will report 15°C.).
5. For debugging help, please see Application Note #71 Thermocouple Troubleshooting. Attaching Vin- to Gnd with a small wire solves many RFI problems. Integrating for .001 to .001666 seconds solves many noise problems.

Thermistor Temperature Measurement

Thermistors are two wire devices whose resistance varies with temperature in a known fashion, often accurate to +/- 0.2°C. The instruNet thermistor measurement feature supports ysi-Omega 100ohm to 1Mohm thermistors between the temperatures of -80°C to 250°C. When Thermistor is selected in the Sensor popup, instruNet assumes a thermistor is connected in the prescribed fashion, and subsequently returns the thermistor temperature in degrees C units after applying a Steinhart & Hart resistance-to-temperature conversion. Measuring a thermistor temperature involves a voltage divider circuit with a shunt resistor of known value, applying a voltage across the pair and measuring the voltage across the thermistor, as illustrated to the right. The voltage across the thermistor is measured between a pair of instruNet Vin+ and Vin- input terminals while the excitation voltage for the divider is supplied by the instruNet Vout terminal. instruNet calculates the thermistor resistance using the following equation.

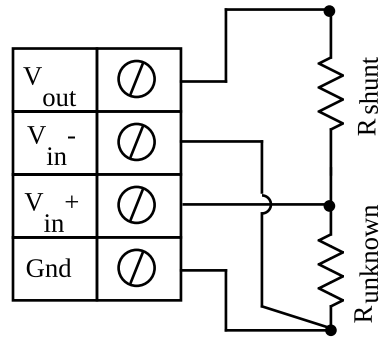


Fig 3.16 - Temperature Measurement (Thermistor) Voltage Divider Circuit

R_{Therm} (Ohms) = R_{shunt} * (Vin+ - Vin-) / (V_{out} - (Vin+ - Vin-))

$\text{Temperature}_{Therm}$ (°C) = -273.15 + 1.0 / (a + b (Ln(R_{Therm})+c (Ln(R_{Therm}))^3)

a, b, and c are a function of 3 points in the resistance-to-temperature table, and are calculated by instruNet after the user completes a short dialog box interview. To minimize "self heating", it is recommended that thermistors operate at less than 100uW (50uW is better). An example would be a 2252 ohm thermistor (which will vary from 394.5 to 7355 ohms in the 0-70°C temperature range). With 0.37V excitation voltage (i.e. Vout) and a 1000 ohm shunt resistor, the current and power dissipation at 70°C would be 100uW:

Current (Amps) = Volts / Resistance = 0.37V / [1000 + 394.5] = .265mA
Power (watts) = Current ^{2} * Resistance = .000265^2 * 1394.5 = 100uW

The voltage across the thermistor would vary from 104mV to 320mV as the resistance across the thermistor varied from 394 to 7355 ohms; therefore, an input Voltage Range of ±.6V would work nicely in this case. Shunt resistors with an initial accuracy of .025%, and a temperature drift of 20ppm/°C, such as the Caddock part #TN130-resistance-0.025%-20, are recommended[1,3,6].

To do temperature measurement using a Thermistor, the user must:

1. Set the Sensor field in the Hardware settings area to Thermistor. This will cause a series of dialog boxes to appear, asking the user several questions about the thermistor type (i.e. resistance at 25°C), shunt resistor value, and excitation voltage. It also recommends an excitation voltage and shunt resistor value based on the thermistor in use, and the temperature range of interest -- in many

cases, using the recommended values are the best options. Based on the responses to these questions, instruNet loads the following fields in the Constants setting group, with the following information:

Ro	thermistor resistance at 25°C, in units of ohms
Rshunt	actual shunt resistance, in units of ohms
Vout	excitation voltage output the Vout screw terminal
alpha	the 'a' Steinhart & Hart coefficient
delta,Rlead	the 'b' Steinhart & Hart coefficient
GF	the 'c' Steinhart & Hart coefficient
V_poisson	the maximum expected temperature, in degrees C units
Vinit	the minimum expected temperature, in degrees C units

instruNet also sets the Voltage Range field in the Hardware settings area to a value appropriate to the specified temperature range. The smaller the temperature range, the better the accuracy; therefore one should not make the specified range unnecessarily large.

If you want to run through the dialog box interview again, select Voltage in the Sensor field, and then select Thermistor, to invoke the interview again.

If you want to manually set any of the fields in the Constants settings group, do the interview, and then set them to your liking, after selecting Constants in the Settings popup.

2. Wire your thermistor per figure 3.16, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor. To reduce noise, 0.001 seconds of integration is often helpful (i.e. set the Integrate field in the Hardware setting area to 0.001) [5, 10].

Load Cell Measurement

A load cell is an external device similar to a strain gage that measures force in Kg or LBs units. The i100xx connects directly to load cells by providing an excitation voltage and returns Kg or LBs units to the end user. When one selects "Load Cell" in the "Sensor" field, an interview leads the user through the setting up of the device. One enters parameters such as device resistance (ohms), excitation voltage specified on package label (V), maximum force (Kg or LB) at specified excitation voltage, and mV/V sensitivity (e.g. one enters "2" if the package label specifies 2mV/V).

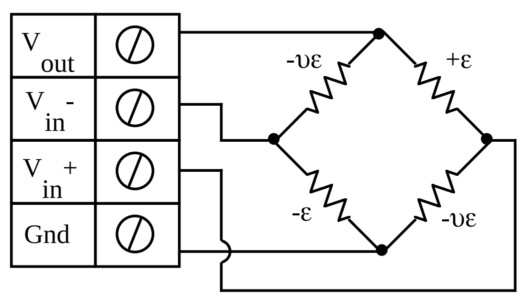


Fig 3.17 - Load Cell Measurement

Load cells are set up in a bridge circuit where a voltage is applied across the bridge, and the voltage across the two intermediate bridge nodes is measured via a pair of instruNet Vin+ and Vin- input terminals. The excitation voltage for the bridge is supplied by either the instruNet Vout terminal or by an external voltage source. instruNet calculates the force using the equations:

V_{ratio} (V/V) = ((V_{in}+ - Vin-) - Vinit) / V_{out}

Force (Kg) = V_{ratio} / Sensitivity

Sensitivity = V_{meas} per V_{exc} per Kg*, as specified on package label*

Ro (bridge resistance in ohms), GF ($V_{measure}$ per $V_{excitation}$ per Kg, as specified on the physical sensor label), Vinit (Voltage measured with 0 force, used to calibrate 0) and Vout are fixed values that are specified by the user in the Constants Settings area (indirectly if set via the interview), whereas (V_{in}+ - Vin-) are measured in realtime by instruNet. For more details, please see Application Notes #14 Strain Gages, #75 Load Cells, and #139 Ratio Measurements.

To do a Load Cell measurement, you must:

1. Set the Sensor field in the Hardware settings area to Load Cell.
2. Set the Range field in the Hardware settings area to +/- 10mV.[7]
3. Set the Ro field in the Constants settings area to the value of one Ro bridge completion resistor, in ohms units.[3,4]
4. Set the GF field in the Constants settings area to the sensors sensitivity in $V_{measure}$ per $V_{excitation}$ per Kg units.
5. Set the Vout field in the Constants settings area to specify the voltage that is to be applied to the bridge (1V is typical). If you are applying an external excitation voltage, enter -Ro value in the Ro edit field (e.g. -100 instead of 100 ohms) to tell the software that the excitation is external, and then enter the value of the external excitation voltage into the Vout field (e.g. 4V).[2] In high current cases (e.g. >2mA), it is often helpful to alternate the polarity of the excitation voltages to evenly burden the +/-12V supplies.[11]
6. Set the Vinit field in the Constants settings area to the voltage measured when 0 force is applied, in Volts units.[8]
7. Set the Wiring field in the Hardware settings area to Bridge.
8. Capacitors across the voltage input terminals are highly recommended for reducing errors caused by RFI. With 350ohm sensors, 0.1uF caps create a low pass filter at 4KHz [4K = 1 / (6.28 * 350 * .1e-6)], and are ideal at minimizing RFI effects.
9. Wire your voltage source per figure 3.17, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10]

Accelerometer Measurement

An Accelerometer is an external device that measures acceleration in G units. They contain an internal constant mA excitation current and provide a voltage output to the instruNet, which does not provide excitation.

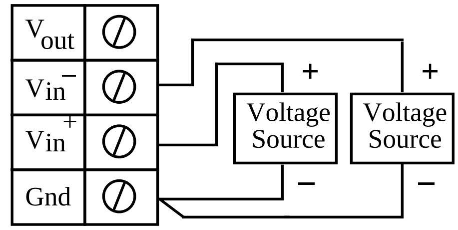


Fig 3.18 - Accelerometer Measurement, Single-Ended Wiring

The i100xx connects directly to accelerometers and returns G units to the end user. When one selects "Accelerometer" in the "Sensor" field, an interview leads the user through the setting up of the device. One enters parameters such as maximum and minimum acceleration (G units) and mV per G sensitivity as specified on the physical package label (e.g. one enters "2" if the package label specifies 2 mV/G).

Accelerometers are set up in either single-ended or differential wiring mode. Figure 3.18 shows a single-ended configuration, yet one can implement differential by wiring as shown in Figure 3.2 and setting the "Wiring" field to differential "Vin+ - Vin-". instruNet calculates acceleration using the equations:

$V_{measure}$ (V) = (V_{in+} - V_{in-}) *or* V_{in+} *or* V_{in-}

Acceleration (G) = $V_{measure}$ / Sensitivity

Sensitivity = $V_{measure}$ per G*, as specified on package label*

GF ($V_{measure}$ per G) is a fixed value that is specified by the user in the Constants Settings area (indirectly if set via the interview), whereas $V_{measure}$ is measured in realtime by instruNet. To do an Accelerometer measurement, you must:

1. Set the Sensor field in the Hardware settings area to Accelerometer.
2. Set the Range field in the Hardware settings area to accommodate the maximum accelerometer voltage (e.g. +/- 5V).
3. Set the GF field in the Constants settings area to the sensors sensitivity in $V_{measure}$ per G units.
4. Set the Wiring field in the Hardware settings area to either differential "Vin+ - Vin-" (figure 3.2) or single-ended "Vin - Gnd" (figure 3.18), and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10]

Potentiometer Measurement

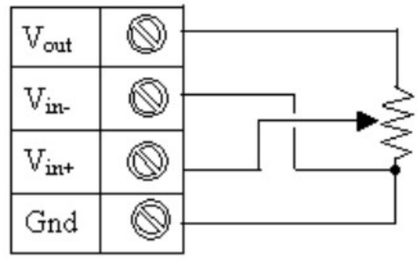


Fig 3.19 - Potentiometer Measurement

A potentiometer (pot) is an external device that measures linear distance or rotation angle. The i100xx connects directly to pots by providing excitation at 4V and returns data in variable 0.0 to 1.0 units (e.g. 0.1 would represent 27 degrees on a 270 degree rotation pot). One can scale this 0 to 1 data to other units using the Channel Mapping dialog. We recommend 5K to 25K ohm pots since lower values may overload the Vout driver and higher values may incur excess noise. Figure 3.19 shows the Vin- terminal being wired to the sensor, yet a jumper wire between Vin- and Gnd will work as well.

When one selects "Potentiometer" in the "Sensor" field, an interview leads the user through the setting up of the device. One enters parameters such as pot maximum resistance (ohms). The excitation voltage for the pot is supplied by either the instruNet Vout terminal or by an external voltage source. instruNet calculates the pot position using the equation:

$$Pot_{position}\ (0 \text{ to } 1) = ((V_{in}+ - Vin-) - Vinit) / V_{out}$$

Vinit (Voltage measured in CCW position, used for calibration) and Vout are fixed values that are specified by the user in the Constants Settings area (indirectly if set via the interview), whereas (V_{in}+ - Vin-) are measured in realtime by instruNet.

To do a Potentiometer measurement, you must:

1. Set the Sensor field in the Hardware settings area to Potentiometer.
2. Set the Range field in the Hardware settings area to +/- 5V.
3. Set the Ro field in the Constants settings area to the value of the potentiometer maximum resistance, in ohms units. 5K to 25K are recommended.
4. Set the Vout field in the Constants settings area to specify the voltage that is to be applied to the potentiometer (4V is typical). If you are applying an external excitation voltage, enter -Ro value in the Ro edit field (e.g. -10000 instead of 10000 ohms) to tell the software that the excitation is external, and then enter the value of the external excitation voltage into the Vout field (e.g. 4V).[2] In high current cases (e.g. >2mA), it is often helpful to alternate the polarity of the excitation voltages to evenly burden the +/-12V supplies.[11]
5. Set the Vinit field in the Constants settings area to the voltage measured when the pot is in the CCW position, in Volts units.[8]
6. Set the Wiring field in the Hardware settings area to Bridge.
7. Wire your potentiometer per figure 3.19, and refer to the steps at the beginning of this chapter for more information on how to set up your sensor.[5,10]

Sensor Reference Footnotes

1 Heating Considerations with Resistors

When current is pumped through a resistor, it heats up. When a resistor heats up, its resistance changes, and this can cause errors in your measurement. The current though a resistor is calculated via:

```
      Current (Amps) = Volts Across Resistor / Resistance in ohms
```

The power dissipated by a resistor is:

```
      Power Dissipated (Watts)  = Volts * Volts / Resistance
                                = Current  * Current  * Resistance
```

The amount a resistor heats up is:

```
      Change in Temperature (Celsius) = Thermal Resistance (C/Watt)
                                    * Power Dissipated (Watts)
```

The amount a resistor changes for a change in temperature is:

```
      Change in Resistance (ohms) = Change in Temperature (Celsius)
                                * Temperature Coefficient (ppm/C)
                                * Resistor Value (ohms)
```

For example, a 100ohm resistor with a 100 ppm/C temperature coefficient and 30C/Watt thermal resistance that is passing 50 milliAmps would enjoy the following situation:

```
      5Volts across resistor = 100ohms * 50mA
      0.25Watts power dissipation = 5V * 5V / 100ohms
      7.5°C temperature change = 30C/Watt * .25Watts
      0.075 ohms change due to temperature change = 7.5°C temp
            change * .0001ohms/ohm/C thermal resistance * 100 ohms
```

2 Excitation Voltages for Bridge Circuits

If you type an unreasonably high value into the Vout field of the Constants area and press Enter, instruNet will set the output voltage to the highest possible value without allowing the internal output amplifier to saturate (e.g. ≤ 4mA for #iNet-100/100B and ≤15mA for #iNet-100HC). Setting the highest possible Vout, causes the highest possible voltage to be read by the Vin terminals, which increases the signal to noise ratio and therefore increases accuracy. The downside to having a high excitation voltage is that it increases the power dissipated by the resistors, which increases their thermal heating, which increases the drift from their resistance's at ambient temperature (e.g. typical resistors offer 100ppm resistance drift per degree C change in temperature). Resistors with low temperature coefficients (e.g. 25ppm/C) are helpful if this is a problem. Also, if noise is a problem, it is sometimes helpful to install a capacitor (e.g. 1µF) between the Vout terminal and GND, at the bridge (possibly far from instruNet), to hold the excitation voltage steady.

3 Shunt Resistors and Bridge Completion Resistors

Bridge completion resistors and shunt resistors should be as accurate as possible (.1% is often ok, .01% is very good), and have a low temperature coefficient

(25ppm/C is often ok, 5ppm/C is very good). If you use a less accurate resistor, we recommend that you measure it with a DVM, and then type this more accurate value into the Rshunt field in the Constants setting area. To determine the effect of a resistor inaccuracy, calculate your engineering units for a typical case, and then increase the resistor value by its expected error, and note the change in the resulting engineering units output. For example, if a 100ohm shunt resistor is used to measure a 1mA current source and it changes 1%, then the measured reading would change 1%, or .01mA.

Several manufacturers of resistors are listed below:

a. Caddock Electronics Inc, USA telephone 1-541-496-0700, Ask for "Greg, The Resistor Man" or a "Resistor Applications Engineer", fax 1-541-496-0408. Caddock offers very precise resistors in the $2 to $20 range. The "MK132 - *Value* - *Tolerance*" series parts are 3/4W, 30-5Mohms, .1% to 5% initial accuracy, 50ppm/C; and the "TF020R - *Value* - *Tolerance* - *Temperature Coefficient*" series parts are .33W, 1K-2Mohms, .01% to 1%, 5 to 15ppm/C. They accept VISA and MasterCard, have no minimum order, and often have items in stock.

b. Digi-Key Corporation, USA telephone 1-800-344-4539, fax 1-218-681-3380. Part # "*value* X-ND" are $.50, 1/4Watt, 100ppm/C temperature drift, 1% initial accuracy Resistors that are not too accurate yet very easy to order and are low cost.

c. "RN 55 E *value* 0.1%" resistors are a common series that offers 1/8Watt, 0.1% initial accuracy, and 25ppm/C temperature drift for approximately $1. Also, the "RNC 60H *value* 0.1%" series is similar, yet with 50ppm/C temperature drift at 1/4W. These are manufactured by folks like IRC Inc, and Dale Inc; and distributed by companies like Newark Electronics (USA telephone 800-463-9275, fax 1-312-907-5217) and Allied Electronics (USA telephone 1-800-433-5700, fax 1-817-595-6444). They accept common credit cards, and if one series is not in stock, ask them to suggest something that is similar.

[4] Bridge Completion Resistors in Strain Gage Bridge Circuits
In a bridge, all 4 resistors must be the same value, within 10% or so (1% is better, .1% is excellent), in order for the bridge to operate properly. In some bridge circuits, all 4 resistors are supplied by the sensor manufacturer; whereas in others (e.g. 1/4 or 1/2 bridge circuits), the user must supply "completion" resistors of the same value as the gage to complete the bridge circuit. This can be done by installing precision resistors (e.g. 0.1%), or by installing fixed unstrained strain gages of the same ohmic value.[3]

[5] Filter Settings for Low Level Measurements
Strain gage, thermocouple, and RTD voltages are typically very low and therefore often require low pass filtering to reduce noise. Low pass filters cause high frequencies to be rejected, while low frequencies are passed. Visually, the signal becomes "smoother". instruNet provides several low pass filter options:

a) The Low Pass popup menu in the Hardware settings area can select a variety of analog low pass filter options (e.g. the Model 100 provides the following analog one pole low pass options: off, 40Hz, and 4KHz).
b) The Integrate field in the Hardware Settings area selects how long the signal is averaged before instruNet returns one number. This "averaging", in effect, is a low pass filter. Careful, this averaging fully consumes the instruNet controller, and therefore reduces the maximum possible sample rate. A 0.001 sec integration time is often very helpful at reducing noise and increasing accuracy.
c) The Low Pass settings area provides a means by which one can digitally filter a signal, post acquisition, with tremendous accuracy.
d) The user can manually place a capacitor across the Vin+ and Vin- input terminals with any bridge or voltage divider circuit to provide a 1pole low pass filter where the cut-off Frequency in Hertz is equal to 1 / (2 * π * R * C); where R is the source resistance in ohms, C the parallel capacitance in Farads, and π is 3.14159.

6 Selecting a Voltage Divider Shunt Resistor

Shunt resistor values are typically chosen to cause a large voltage (several volts maximum) to be measured by instruNet, without heating up the resistor significantly to cause its resistance to change or causing the excitation voltage source to over shoot its maximum output current (e.g. 4mA on the #iNet-100/100B and 15mA on #iNet-100HC). If the Rshunt value is low, then the voltage across it is low, and this decreases the signal to noise ratio of the measured signal. Also, Rshunt must be selected such that the voltage across Runknown does not exceed the instruNet maximum input voltage (e.g. +/-5V with the Model 100). Due to these limitations, instruNet might not let you set some of the fields too high or too low.

7 Voltage Range Settings For Strain Gages

Since strain gage voltages are often very small, a small input range (e.g. +/- 10mV) works best for most measurements. Increasing the voltage range increases the range of strain that can be read, while sacrificing accuracy with small voltages (e.g. instruNet can read 5mV more accurately with a +/-10mV range, than with a +/-100mV range). Please refer to your equation for details on how strain relates to voltage measured.

8 Balancing your Bridge with the Vinit Correction Voltage

Vinit is the voltage measured across the intermediate nodes of the bridge when the strain gage(s) are unstrained in a bride circuit. This is measured by putting instruNet into Voltage mode, Differential Wiring, with a low voltage Range (e.g. +/-10mV), and then measuring the resulting bridge voltage (e.g. via the value shown at the bottom of the Probe dialog). You must then enter this voltage value into the Vinit field in the Constants settings area, reset your Sensor field to Strain Gage, and reset your Wiring field to where you had it. Subsequently, all reported "strain" values will reflect resistance changes from the "unstrained" scenario. Vinit is used as an offset correction factor to "balance" the bridge. If you do not want to go through the trouble of "balancing" your bridge, simple set Vinit to 0.

9 The Strain Gage "GF" Factor

All strain gages are manufactured with a specific Gage Factor (GF), which relates a change in resistance, to strain. The GF is often printed on the strain gage package, and must be correctly entered into the instruNet GF field within the Constants settings area. This is used to calculate the "strain" value returned by instruNet.

10 Accuracy of Measurements

Accuracy measurements are affected by the noise pickup on the leads, the accuracy of the sensor itself (i.e. thermocouple's are typically accurate to +/- 1C to 3C) and the instruNet measuring system. A noisy environment and long sensor leads are often the worst threat to accuracy. Integrating (via the Integrate field) a signal over a period of time will give a more accurate measurement by filtering out noise at the expense of a lower maximum sample rate.

An example of how to calculate accuracy is as follows:

Suppose you are doing a current measurement where the current is calculated as the voltage drop across a shunt resistor divided by the resistance in ohms of the resistor.

Current (Amps) = Volts across shunt resistor / shunt resistance in ohms

Suppose the measured voltage is accurate to 1mV and the 1K ohm shunt resistor is accurate to 1%. Subsequently, the accuracy of the measured current would be

Max Current Error = 1mV / (.01 * 1K) = 100microAmps

11 Alternating Positive and Negative Excitation Voltages

To reduce the burden on one side of a power supply (e.g. +12V or -12V), excitation voltages often alternate positive and negative. For example, when powering 350ohm strain gages, the excitation voltages are typically set to {+5V, -5V, +5V, -5V...}. The alternating polarity evenly burdens the +/-12V supplies. Please note that in low current cases (e.g. <2mA), this is not necessary.

Chapter 4, A Tutorial For Programmers

This chapter explains how to control instruNet from a programming environment such as C or Visual Basic. It assumes you have done the tutorial in Chapter 2, in its entirety. If you haven't, please do so now. This chapter also assumes you have a working knowledge of your programming language and programming tools.

Getting Started Quickly

The instruNet Setup.exe installer file version >= 1.4 for Windows includes instruNet programming interfaces for the following languages:

- `Microsoft Visual Studio C/C++ 5`
- `Microsoft Visual Studio .Net C/C++ 7`
- `Microsoft Visual BASIC 4`
- `Microsoft Visual BASIC 6`
- `Microsoft Visual Studio .Net BASIC 7`

Documentation resides at:

- Select under Windows START Menu: All Programs / instruNet / Programming Interfaces /.
- See instruNet Programming Application Notes
- See Chapter 4 of the instruNet Manual (i.e. this chapter).

Interface files reside at the following Windows directory after running the instruNet Setup.exe installer program.

- Program Files \ instruNet \ Programming Interfaces \

Programming Overview

An instruNet data acquisition system is controlled with one main subroutine, called iNet(), that is callable from C or Visual Basic. The actual iNet() code resides in the instruNet Driver file that you installed in Ch 1 (i.e. Macintosh Code Resource or a Windows 32bit DLL).

iNet() Function Call

The iNet() function includes 7 parameters that specify a field in the network hierarchy that is to be read or written to, as shown below:

```
iNetINT16 iNet(
   iNetUINT8 netNum,         NETWORK number = {0...numNetworks}, 0 is
                             Driver, 1 is 1st Controller installed
                             into the computer, 2 is the 2nd
                             Controller, etc.

   iNetUINT8 deviceNum,      DEVICE number {0...numDevices}, 0 =
                             "Controller", 1 = 1st device on net...

   iNetUINT8 moduleNum,      MODULE number within a hardware DEVICE
                             {1...32}. Each DEVICE can contain up to
                             32 separate modules.  Many devices have
                             only 1 module.

   iNetUINT8 chanNum,        Hardware CHANNEL number {1...32}. Each
                             device contains a number of channels
                             (i.e. signals that are accessed//via a
                             screw terminal or connector), each of
                             which has its own channel #.

   iNetINT16 fieldGroupNumOrType,
                             If > 0, this is a field Group Number
                             {1...numFieldGroups}.  if < 0, this is a
                             field Group Type.

   iNetINT16 fieldNum,       Field number within the fieldGroup of
                             the CHANNEL.

   iNetUINT8 intention       ion_intention = {intention_getValue,
                             intention_setValue,
                             intention_getNameStr,
                             intention_getMaxValue}

   iNetUINT8 argType         argument type {int16, int32, str15,
                             etc}(void *ptrToArg) this is where data
                             is kept (we read or write to this
                             location).
```

These parameters specify a field, such as the cut-off frequency of a low pass filter, the sample rate, or the actual real-time value of a channel. This function can both read from and write to any field on the network. For a description of each field, please refer to Ch 6. One uses iNet() to both set up the networks, and then do I/O with the various channels. Also, iNet() can be used to tell the instruNet Controllers to digitize, and then download the digitized data into computer memory. iNet() is extremely powerful -- it can even open the instruNet World window and turn over control to the user.

Simple Format Functions

There are also a collection of Simple Format functions that read and write instruNet fields using a structure that specifies a channel address {netNum, deviceNum, moduleNum, chanNum}. This structure ("User Defined Variable Type" if in Visual BASIC) is defined as follows:

```
typedef struct iNetChannelAddr {

  iNetUINT8 netNum;     NETWORK number = {0...numNetworks}, 0 is
                        Driver, 1 is the first controller installed
                        into the computer.
  iNetUINT8 deviceNum;  DEVICE number = {0...numDevices}, 0 =
                        "Controller", 1 = 1st device on network.
  iNetUINT8 moduleNum;  MODULE number within hardware DEVICE {1-32}.
                        Many devices have only 1 module.
  iNetUINT8 chanNum;    Hardware CHANNEL number {1...32}. Each device
                        contains a number of channels, each of has
                        which its own channel number.
} iNetChannelAddr;
```

To read or write an instruNet field with a Simple Format function, one must first load an `iNetChannelAddr` structure with the netNum, deviceNum, moduleNum, chanNum. This is often done with the `LoadChannelAddress()` routine:

```
  LoadChannelAddress()  Specify an instruNet channel address.
```

After the structure is loaded (it holds the netNum, moduleNum, deviceNum, chanNum information), one can then read or write an instruNet field with any of the following routines:

```
  SetField_int32()      Set instruNet field with 32bit signed
                        integer number.
  SetField_uint32()     Set instruNet field with 32bit unsigned
                        integer number.
  SetField_flt32()      Set instruNet field with 32bit floating
                        point number.
  SetField_cStr()       Set instruNet field with C string
                        (terminated with 0x00).
  GetField_int32()      Read instruNet field into a 32bit signed
                        integer variable.
  GetField_uint32()     Read instruNet field into a 32bit unsigned
                        integer variable.
  GetField_flt32()      Read instruNet field into a 32bit floating
                        point variable.
  GetField_cStr()       Read instruNet field into a C string
                        variable (terminated with 0x00).
```

For example, the following C code would read the real-time value of Channel 1 at Device 1, Module 1, Network 1:

```
iNetChannelAddr vin;
iNetFLT32 V;
iNetError e;

LoadChannelAddress(&vin, 1, 1, 1, 1);

V = GetField_flt32(&vin,sgt_General,fldNum_General_valueEu,&e);
```

Digitizing

There are several routines, summarized below, that are used to simultaneously digitized channels to ram or to disk, as described in C file `INET_INT.C` and Visual BASIC file `INET_Common_Code.bas.`

```
EnableChannelForDigitizing()
                     Mark this channel for digitizing (i.e. when
                     digitizing is started).
DisableAllChannelsForDigitizing()
                     Disable all channels for digitizing.
Set_iNet_TIMING_Parameters()
                     Set digitize timing parameters (e.g.
                     sampleRate, ptsPerScan, etc).
Set_iNet_TRIGGER_Parameters()
                     Set digitize trigger parameters (e.g.
                     trigger on/off, trigger source).
Service_All_iNet_Digitize_Buffers()
                     Services internal buffers while digitizing.
Access_Digitized_Data_In_Ram_Buffer()
                     Provides access to digitized data (even
                     during acquisition).
```

Simple digitizing is illustrated nicely in Example#2. In summary, to digitize, one must:

1. Call `DisableAllChannelsForDigitizing()` to disable all channels for digitizing.

2. Specify which channels are to be digitized by calling `LoadChannelAddress()` and `EnableChannelForDigitizing()` for each digitize channel.

3. Specify timing parameters (e.g. sample rate) with `Set_iNet_TIMING_Parameters()`.

4. Specify trigger parameters with `Set_iNet_TRIGGER_Parameters()`.

5. Tell the instruNet driver to start digitizing with `PRESS_iNet_BUTTON( iNetCM_BtnPress_Record_Start )`.

6. Periodically (e.g. 4 times per second) call `Service_All_iNet_Digitize_Buffers()` to provide the instruNet Driver time to service internal buffers (this is mandatory).

7. Call `Access_Digitized_Data_In_Ram_Buffer()` to access digitize data for each channel.

Support Functions

Additional functions defined and described in C file `INET_INT.C`, and in Visual BASIC file `INET_Interface.bas`, are listed as follows:

UTILITY FUNCTIONS

```
iNet_Peek_int16()   Reads 16bit integer number from a specific
                    logical address.
iNet_Peek_int32()   Reads 32bit integer number from a specific
                    logical address.
iNet_Peek_flt32()   Reads 32bit floating point number from a
                    specific logical address.
iNet_Poke_int16()   Writes 16bit integer number to a specific
                    logical address.
```

`iNet_Poke_int32()`	*Writes 32bit integer number to a specific logical address.*
`iNet_Poke_flt32()`	*Writes 32bit floating point number to a specific logical address.*
`iNet_Get_VarPtr()`	*Returns a pointer to the passed variable.*
`iNet_memcopy()`	*Copies memory block from one place to another.*

ADVANCED FIELD READ/WRITE ROUTINES

`iNet_int32()`	*Reads or writes to a field via a 32bit signed integer variable.*
`iNet_uint32()`	*Reads or writes to a field via a 32bit unsigned integer variable.*
`iNet_flt32()`	*Reads or writes to a field via a 32bit floating point variable.*
`iNet_cString()`	*Reads or writes to a field via a C string variable.*
`iNet_pString()`	*Reads or writes to a field via a Pascal string variable.*
`iNet_DLL()`	*Reads or writes to a field via any type of variable.*

DRIVER ROUTINES

`Load_instruNet_Driver()`	*Loads instruNet driver into memory.*
`CloseDriverAndReleaseDriverRam()`	*Closes instruNet driver and releases memory.*
`Show_ALERT_if_hit_iNet_Error()`	*Shows an alert if instruNet hit an error.*
`Show_Simple_Alert()`	*Displays a message in an alert dialog box.*
`Get_iNet_Error()`	*Gets error code to last instruNet function call.*
`Get_Last_iNet_Call()`	*Returns pointer to struct with params to last iNet call.*

Interface Files

The instruNet disk includes interface files for C and Visual BASIC that enable your program to call the above listed functions. No compiler object files are used; therefore, you are not at the mercy of a specific version of a specific compiler on a specific computer. Instead, you simply add the following glue source code to your program:

VB Files	C Files	Interface Description
`INET_Interface.vb`	`INET_INT.C`	*Contains low level source code that interfaces to the instruNet driver.*
`INET_Declarations.vb`	`INET_INT.H`	*Contains many low level enums and #defines that support instruNet.*
`INET_Macros.vb`	`INET_MCS.H`	*Contains macros that help set/read instruNet fields.*

Every instruNet C program must include the following files: `INET_INT.C`, `INET_INT.H`, and `INET_MCS.H`.

Every instruNet Visual BASIC program must include the following files: `INET_Interface.vb`, `INET_Common_Code.vb`, `INET_Declarations.vb`, and `INET_Macros.vb`.

Programming Examples

The instruNet disk includes example programs in Visual BASIC and C. The C Example #1 and VB Example #1 do the same thing, yet just in different languages. In fact, they do many of the same things that are done in Ch 2 Tutorial. For example, in Ch 2, one is asked to turn on a low pass filter by selecting a popup menu. Example#1 will do the same thing with the iNet() function call. With instruNet, each task can be done automatically under program control, or manually in the instruNet World window. This will become more clear as you work with the example programs. The instruNet disk includes the following example programs:

C Files	VB Files	Example Description
`INET_EX1.C`	`EndUser Example1.sln` `EndUser Example1.vb`	*This is a lengthy and comprehensive example that shows how to call almost every instruNet function in a text window-based environment.*
`INET_EX2.C`		*This is a simple example that shows how to read and write instruNet fields and how to digitize.*

instruNet Data Types

instruNet defines its own data types that allow you to keep your code platform, machine, and operating system independent. instruNet data types are defined in the file `INET_INT.H`. An example of an instruNet data type definition is:

```
typedef unsigned short iNetUINT16;  /* 16bit unsigned integer */
```

Always use instruNet data types when calling the instruNet driver and macros.

In this manual, the following are used to refer to variable types:

iNetINT8	signed 8bit integer, -128 to +127.
iNetUINT8	unsigned 8bit integer, 0 to +255.
iNetINT16	signed 16bit integer, -32768 to +32767.
iNetUINT16	unsigned 16bit integer, 0 to +65536.
iNetINT32	signed 32bit integer, -2,147,483,648 to +2,147,483,647.
iNetUINT32	unsigned 32bit integer, 0 to +4,294,967,295.
iNetFLT32	32bit floating point number

instruNet Macros

instruNet defines a number of macros for the more common driver calls. The macros are defined in file `INET_MCS.H`. The macros are platform, machine and operating system independent. An example of a macro call is:

```
OPEN_instruNet_WINDOW( instruNetCM_OpenWindow_Network );
```

which when called will open the Network Page of instruNet. The Network Page could also have been opened by directly calling the instruNet C function:

```
iNet(netNum_DRIVER, deviceNum_CONTROLLER, moduleNum_1stModule,
   driver_ChanNum_OpenWindow, sgt_General,
   fldNum_General_valueEu, intention_setValue, instruNetDT_INT16,
   (void *) ((gINT16TempArg = specificPage) ? &gINT16TempArg :
   &gINT16TempArg));
```

Example Code

Example#1 does the following things using the instruNet interface functions. C source code is shown in `Courier font`.

1. <u>Load instruNet Driver</u>

The C code below calls `Load_instruNet_Driver()` to load the instruNet Driver, and to get the number of installed controller cards. If the driver does not load properly, an error is returned and `Show_ALERT_if_hit_iNet_Error()` displays a dialog box with an error code. Please consult Appendix II for more information on error codes.

```
iNetError e;
iNetINT16 driverIsInstalledOK, numNetworks;
e = Load_instruNet_Driver(TRUE /* reset_instruNet */,
            &driverIsInstalledOK, &numNetworks);
if (e != iNetErr_None)  {
    Show_ALERT_if_hit_iNet_Error(e);  return 0;  }
```

2. <u>Tell the instruNet Driver to open the Network Page Window</u>
The C code below tells the driver to open the Network Page Window. From the Network Page, fields can be viewed and set following the procedures in *Chapter 2 instruNet Tutorial.* If a non zero error code is returned the program will jump to an exit routine.

```
e = OPEN_instruNet_WINDOW( instruNetCM_OpenWindow_Network );
if (e) { goto Exit; }
```

3. <u>Read several voltage inputs and set a voltage output</u>
The next few lines of code read several voltage inputs and write to a voltage output. They assume a Model 100 is attached to the 1st network controller card.

4. <u>Print the values for the first 5 channels on instruNet</u>
The C code below calls the function `Print_iNet_Channels()` which reads one value from each of the first five input channels of an instruNet Network, and prints them to the console window. The function call is defined in file `INET_EX1.C`.

```
e = Print_iNet_Channels(5);
if (e) { goto Exit; }
```

6. <u>Print the first 3 Fields</u>
The C code below calls the function `Print_iNet_Fields()` which scans the network and prints the values of the first 5 fields that it finds. The function call is defined in file `INET_EX1.C`

```
e = Print_iNet_Fields(3);
if (e) { goto Exit; }
```

7. <u>Read and write instruNet Fields</u>
The C code below calls the function `ReadWriteFields_iNet()` prompts the user for a field address, displays the current value of the field and allows the user to change the field value. The function call is defined in the program `INET_EX1.C`.

```
e = ReadWriteFields_iNet();
if (e) { goto Exit; }
```

8. <u>Digitize some analog input channels</u>

The C code below calls the function `SimultaneouslyDigitizeAndAnalyze()` which digitizes several channels. When writing code that acquires data seamlessly it is important to continuously call `Service_All_iNet_Digitize_Buffers()` while the data acquisition is running. This function call is defined in the file Example#1 file.

```
e = SimultaneouslyDigitizeAndAnalyze();
if (e) { goto Exit; }
```

9. Modify the Example source code to meet your own needs.

Tutorial Summary

In summary, the Programming Tutorial will involve the following steps:

1. Make sure you have done Ch2 Tutorial in its entirety.

2. Install your programming environment on your hard disk, load an example program shipped with your Programming Language, compile, and then run to verify that your Programming Environment is set up properly.

3. Create a new Folder/Directly on your hard disk. Make a copy of the instruNet example program and place it into this folder, along with include files.

4. Compile and link the code. If you have any compile or link errors, please use the source code to debug the system -- all source code is provided.

5. Run the example programs to verify that the compiler, example files, driver, and computer are all working well together.

6. Read the example program source code and comments to get a feel for instruNet programming.

Getting Started

To do the above steps with the programming language of your choice, please jump to the appropriate "Working with ..." in the following pages.

WORKING WITH ANY C COMPILER (Macintosh or Windows)

instruNet provides C source code that is designed to run on virtually any C compiler for the Macintosh and Windows.

To run the example instruNet C code, we recommend that you create a project that supports the printf text console (not needed with Windows), C, the Operating System routines, and ANSI routines. At a bear minimum, you would need to include the following files into a project in order to run an instruNet Example:

`INET_EXN.C`	Contains Example#*N* code (use one example .c file at a time)
`INET_INT.C`	Contains interface to instruNet Driver (includes files: `INET_MCS.H, INET_INT.H`)
---	
ansi library	Contains ANSI subroutines
os library	Contains interface to Operating System
console library	Supports printf() text window (only use with Macintosh)

To compile the instruNet example code and run it, you must:

1. Install the instruNet software onto your computer following the directions in Chapter 1 *Installation.*

2. Make sure you have the correct version of the instruNet Driver installed on your computer.

3. Create a new folder, called "instruNet Example", and place 2 folders inside it called "instruNet C Source" and "End User Source". Copy (i.e. duplicate) the following files and place them into the "instruNet C Source" folder:

`INET_INT.C, INET_MCS.H, INET_INT.H, INET_EXn.C`

4. Create a new project that supports the ANSI library and the console window.

5. Add `INET_INT.C` and `INET_EXn.C` to the project.

6. Compile and run.

`INET_INT.C` and `INET_EXn.C` are designed to run without a prefix file pulled in before them (i.e. prefix option in the .... dialog). If you do have a prefix that is automatically pulled in, you might get some compiler errors and need to debug it a little. In general, `INET_INT.C` and `INET_EXn.C` can run with either 4 or 2 byte int's (it assumes short's are 2bytes), 8 to 12 byte doubles, different struct packing schemes, etc. These 2 files are designed to be portable across different processors, different operating systems, and different compilers.

The `INET_INT.C` file is all you need to add to your project to run instruNet. It uses the `INET_MCS.H` and `INET_INT.H` include files. The `INET_EX1.C` file contains many useful routines, yet you might not want the console to run, and therefore you might want to modify `INET_EX1.C` to your own taste.

Some lines of code in the example program open the instruNet World window, and pass control to instruNet World (until the user closes the instruNet World window to return back to the example program). Don't let this freak you out.

7. Read the example program source code and comments to get a feel for instruNet programming. Step through the source code with the debugger to see it runs on a line-by-line basis. Read the documentation in the `INET_MCS.H` and `INET_INT.H` include files.

Getting Started with Mac metrowerks CodeWarrior C/C++

The instruNet interface for metrowerks CodeWarrior C/C++ is compatible with metrowerks CodeWarrior PowerPC. Recommended system requirements are a PowerPC/G3/G4 Macintosh with 32 Meg of RAM. System 8.0 or newer are also required.

Getting Started

instruNet can run on any version of metrowerks C greater than 7.0. To get started with CodeWarrior C/C++, please do the following steps:

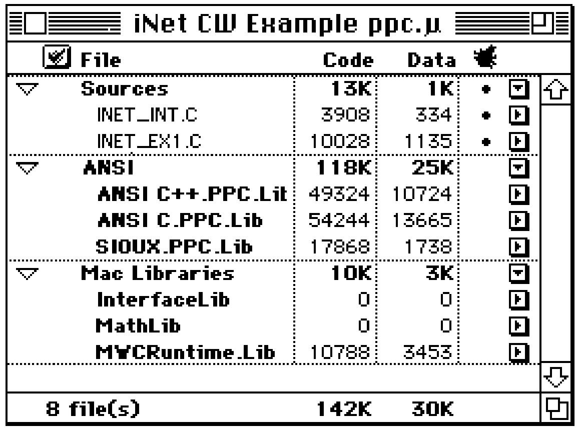
1. Make sure you have done *Ch1 Installation* and *Ch2 Tutorial* in its entirety.

2. Install metrowerks CodeWarrior onto your computer, launch the compiler, and then run an example CodeWarrior project to verify that your C/C++ compiler is set up properly. You can try running the "MW Hello World" project or any of the other metrowerks supplied projects. Close the example project.

3. Create a new folder, called "instruNet CodeWarrior C Example", and place 2 folders inside it called "instruNet C Source" and "End User Source". Copy (i.e. duplicate) the following files and place them into the "instruNet C Source" folder:

`INET_INT.C, INET_MCS.H, INET_INT.H, INET_EX`*`n`*`.C`

4. Select New Project... from the File menu. In the Project Stationary popup, select `ANSI PPC C/C++.µ` if compiling for PPC. Specify a project name (e.g. "instruNet CodeWarrior C Example"), and then place the project into the instruNet example folder created in step 3. After creating the project: choose Preferences under Edit, select "PPC Project" and set the Preferred and Minimum Heap (i.e. application file) size to 1500KB. Also, select Enable Debugger in the Project menu to turn on source level debugging (unless you want to use less ram at run time and do not want to step though the source code and read the documentation).



PPC CodeWarrior C Project

5. Add the files `INET_INT.C` and `INET_EXn.C` to the project.

6. Select Run from the Project menu. The `main()`in file `INET_EX1.C` will first run instruNet World. You can view and change fields and record data following the direction in *Chapter 2 Tutorial*. When you quit instruNet World the program will put you in the console window where you can read and write individual fields. To exit the console window select Quit from the file menu. Some lines of code open the instruNet World window, and pass control to instruNet World (until the user closes the instruNet World window to return back to the example program). Don't let this freak you out.

7. Read the example program source code and comments to get a feel for instruNet programming. Step through the source code with the debugger to see it run on a line-by-line basis. Read the documentation in the `INET_MCS.H` and `INET_INT.H` include files.

8. Modify the Example source code to meet your own needs.

Getting Started with Microsoft Visual BASIC ≥4.0 for Windows

Compatibility

The instruNet interface for Visual BASIC is compatible with VB Version ≥ 4.x, which requires a ≥ 80386 PC / PC Compatible Computer running Windows >= 95 and ≥8 MB Ram. A BASIC for the Macintosh is not supported by instruNet.

Every instruNet Visual Basic program must include the .vb files in the "VB Common Source" directory.

Getting Started

To get started with Visual BASIC ≥ 4.x, please do the following steps:

1. Make sure you have done Ch2 Tutorial in its entirety.

2. Install Visual BASIC onto your hard disk, load an example program, compile, and then run to verify that your BASIC is set up properly.

3. Duplicate the "instruNet \ Programming Interfaces \ Visual Basic" directory -- we will run/modify the files in the duplicate copy. This directory contains the following sub-directories:

`VB Example1`	example program #1
`VB Scope`	1 to 16 channel oscilloscope/chart recorder
`VB Instrument`	2ch oscilloscope, XY recorder, Spectrum Analyzer
`VB Direct To Excel`	digitize directly to Excel Version ≥8
`VB Common Source`	common source code files

4. Run one of the example programs. If you have any errors, please use the source code to debug the system -- all source code is provided. Some lines of code open the instruNet World window, and pass control to instruNet World (until the user closes the instruNet World window to return back to the example program). Don't let this freak you out.

5. Run an example program to verify that the compiler, example files, driver, and computer are all working well together.

6. Read the example program source code and comments to get a feel for instruNet programming.

7. Modify the Example source code to meet your own needs.

Getting Started with Microsoft C/C++ ≥ 5.0 for Windows

Compatibility The instruNet interface for Microsoft C/C++ is compatible with C Version ≥5.0 (including Visual Studio .Net 7), which requires a ≥ 80386 PC / PC Compatible Computer running Windows ≥ 95 and ≥8 MB Ram.

Getting Started To get started with Microsoft C/C++, please do the following steps:

1. Make sure you have done Ch2 Tutorial in its entirety.

2. Install Microsoft C/C++ onto your hard disk, load an example program, compile, and then run to verify that your C/C++ is set up properly.

3. Duplicate the "instruNet \ Programming Interfaces \ Visual C" directory -- we will run/modify the files in the duplicate copy.. It contains the following source files: `Inet_int.c`, `Inet_int.h`, `Inet_Ex1.c`, `Inet_Ex2.c, and iNet_mcs.h`.

4. Open "`Inet32_ExampleN.sln`" (or ".dsp" with C version 5) with the Microsoft Developer Studio, select Set Default Configuration under Build, choose Win32 Debug, press OK, select Debug > Step Into under Build and watch it compile. You should not get any compiler errors or warnings. If it builds properly, it will begin to run under the debugger. Press the Step Over under Debug (i.e. F10) button to step through the example program.

5. Step through the example program to verify that the compiler, example files, driver, and computer are all working well together. Some lines of code open the instruNet World window, and pass control to instruNet World (until the user closes the instruNet World window to return back to the example program). Don't let this freak you out.

6. Read the example program source code and comments to get a feel for instruNet programming.

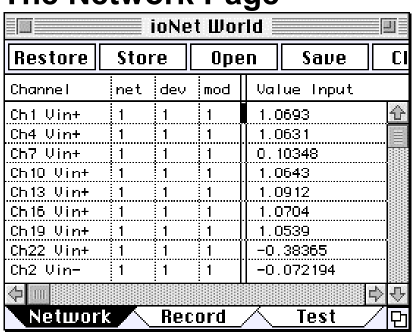
7. Modify the Example source code to meet your own needs.

8. If you want to operate instruNet from another C program, copy files `Inet_int.c`, `Inet_int.h`, and `iNet_mcs.h` to your target project source directory, add `Inet_int.c` to your project, call any of the `iNet_mcs.h` macros from your own source code, call any of the `Inet_int.c` subroutines from your own source code, copy any of the useful subroutines in `Inet_ExN.c` into your own source code, and away you go.

Chapter 5, instruNet World Program Reference

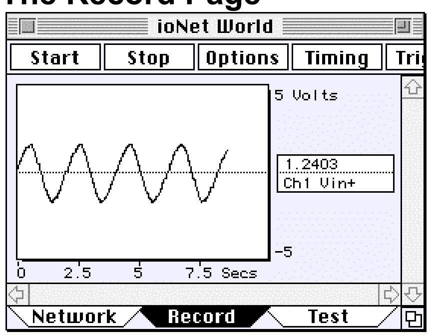
This chapter provides a detailed reference of the features within the instruNet World Application Program, which is what we believe the 21st Century Strip Chart Recorder/Oscilloscope will look like. This chapter is intended to be used as a reference, whereas *Chapter 2 Tutorial* is recommended to teach the basics. To install instruNet World, please follow the directions in the Software Installation section of *Chapter 1 Installation* .

The Network Page



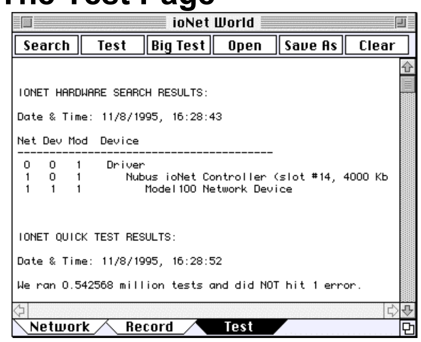
The Network Page is used to view and set parameters within an instruNet network. This page provides a spreadsheet-like format to scroll vertically through channels (i.e. sensors), and horizontally through the settings for each channel. At any time, one can click on a cell to open the Probe dialog, which is used to view and modify individual settings.

The Record Page



The Record Page is used to Start, Stop, View in real-time, Save to disk, and Load from Disk waveforms. Only channels that have been turned "on", via the Probe dialog, are recorded. The sample rate and number of points digitized per channel are specified by pressing the Setup button. The Trigger is specified by pressing the Trigger button.

The Test Page



The Test Page is used to determine what instruNet hardware is attached to your computer, and to test all instruNet hardware and software. After each test, a report is printed to a miniature text editor within the Test Page. The user can then type notes into this window, and save it to disk as a text file.

Figure 5.1 instruNet World Pages

Overview

instruNet World is an application program that can manage, monitor and operate an instruNet hardware network. This network can be used to digitize long continuous waveforms, spool them to disk, view incoming waveforms in real-time and then allow post acquisition viewing. instruNet World is included, free of charge, with all instruNet Controller boards and does not require programming experience -- it is as easy to use a simple word processor (relax !). This chapter describes the three primary instruNet pages, as illustrated in Figure 5.1. Each page is selected by pressing a Tab at the bottom of the window.

Menubar Vs. Buttons

This manual focuses on the buttons at the top of the instruNet World pages, yet these functions can *also* be accessed in the menubar. For documentation on the menubar and on instruNet World PLUS "iW+" (a version of instruNet World with more features), please select "instruNet World+ Manual" in the "Help" menu within instruNet World, or see Internet URL www.instrunet.com/plus.

The Network Page

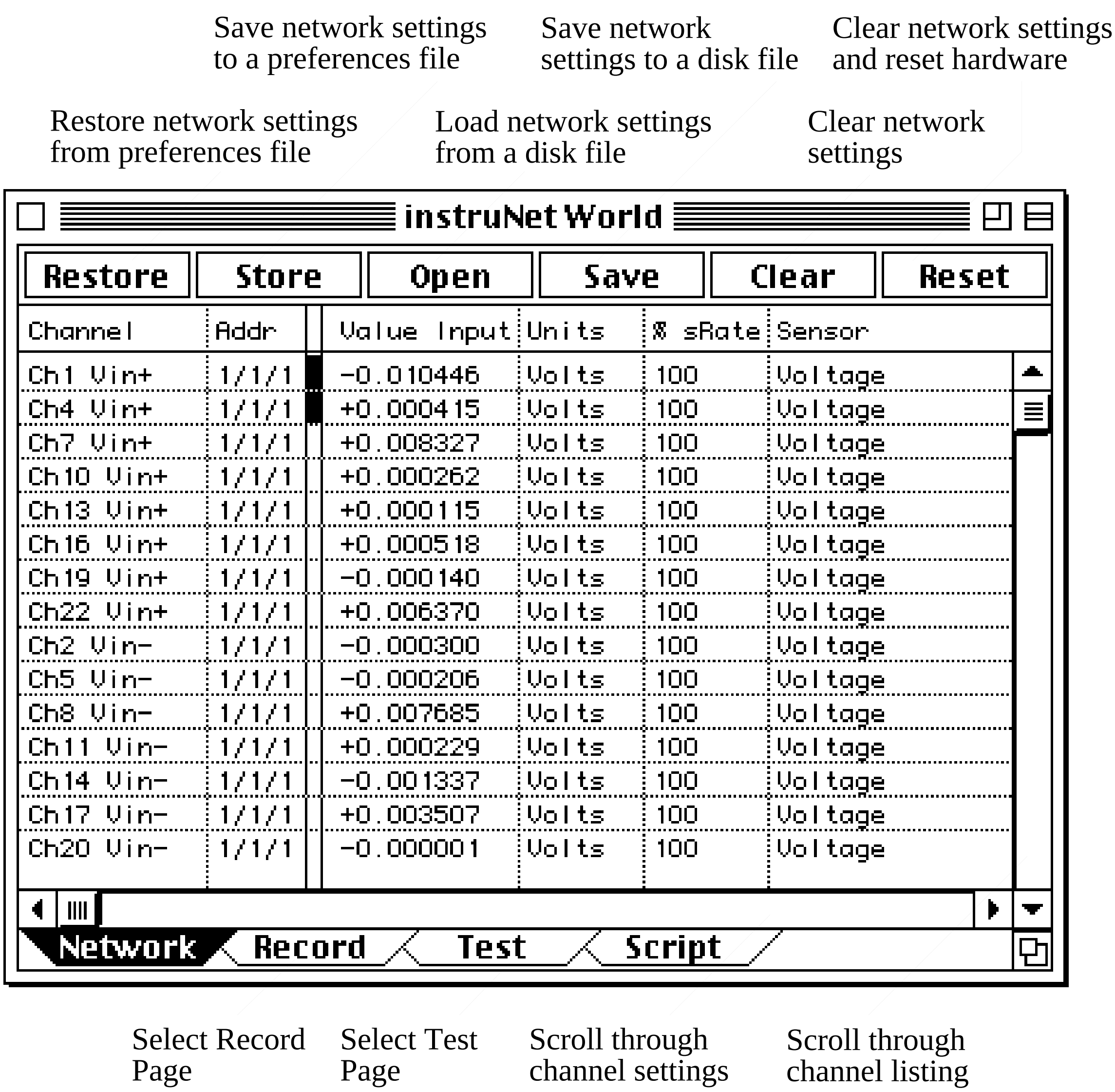
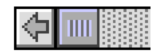


Figure 5.2 The Network Page

Overview

The Network Page is used to view and modify Fields that control all Analog, Digital, and Timer I/O channels within an instruNet network. These Fields do things like define the cut-off frequency of a digital low pass filter, set the range of a voltage input, or define the type of sensor that is connected to the input.

Surfing The Net



After starting the instruNet World application the Network Page is selected by clicking the Network tab at the bottom of the instruNet World window. Your Network Page will resemble Figure 5.2 although your rows and columns may differ depending on what is attached to your network.

Each row corresponds to a Channel (e.g. typically one sensor is attached to each voltage input channel), and the columns are used to display the settings of Fields that pertain to each Channel. The horizontal scroll bar is used to scroll through the Fields, and the vertical scroll bar is used to scroll the Channels.

Modifying Fields

To change a Field, simply click on its cell. The Probe dialog will pop open and it is here that one can view and modify all Fields within the network. Typical networks will house thousand of Fields yet their default values, in most cases, will suffice (relax !). For a detailed description of the various settings, please refer to *Chapter 8 Settings Reference* of this manual.

Channels

Figure 5.3 shows an expanded view of the Network Page. The Channel names appear in the left-most column, and the real-time values of each channel are under "Value Input". The "network address" (i.e. network number, device number, and module number) of each channel is shown in column 2. The first channel in the first module of the first device attached to the first Controller will have a physical network address of {1,1,1,1}. The table header is optimized for the upper-most visible channel, and therefore may change slightly when scrolling vertically.

Turning A Channel On

To "enable" a channel for digitizing (so you can see its signal in the Record Page), one must click on the small vertical rectangle after the network address. It will darken to indicate that the channel will be digitized when the Start button is pressed in the Record Page. Clicking it again will turn it "off". Figure 5.3 shows Channel "Ch1 Vin+" as being "turned On".

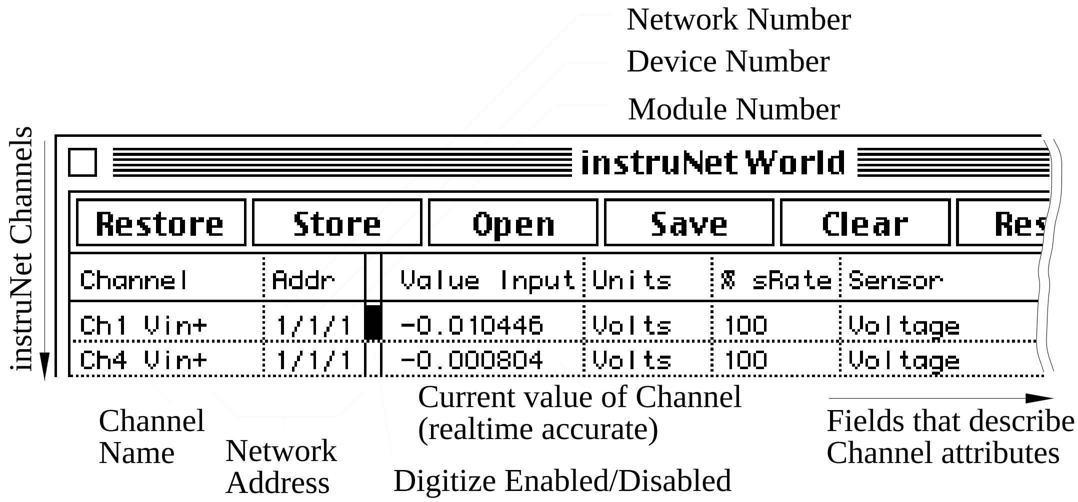


Figure 5.3 Expanded view of the Network Page

Pull Down Menus

Pull down menus vary depending on your operating system. For example, the Macintosh version of instruNet World provides the minimum menus of Quit under File; and Cut, Copy, & Paste under Edit. Control keys (e.g. Control C, or Command C for Copy-To-Clipboard) are also supported in the Test Page's text window.

Saving Your Configuration

The buttons at the top of the Network Page are used to
save Field settings to disk, load them from disk, Clear them to their default values, and to Reset hardware. In a typical session, the user will modify several Fields (e.g. turn on 2 channels and set their filter and sensor parameters), save these settings to disk via the Save button, exit instruNet World, run instruNet World a week later, and reload these settings via the Open button. When the state of instruNet World is saved to disk, you are saving the Fields that have changed from their default values. When you load the state back in, instruNet will first set all Fields to their default values, load the file, and then update the Fields that were changed. -- therefore, these "state" files are typically small (e.g. several KB). In fact, you can open these text files with a Word Processor or Spreadsheet program to glean a sense as to how instruNet internally stores Fields settings. instruNet keeps all state in Fields, therefore, it is not necessary to save anything else (except waveform data) in order to get you back to where you were. Saving waveform data is done independently via the Save and Open buttons at the top of the Record Page. The Network Page manages Fields, whereas the Record Page manages waveforms.

Reconnecting With A Changed Network

If your instruNet Network changes (e.g. you physically pull out 1 device, and then install 2 more), the stored Field settings may no longer apply to your new network. instruNet does its best to "reconnect" and will alert you to any discrepancies, yet it is always a good idea check your settings after physically changing your network. Typically, adding hardware will not cause discrepancies, since you are adding Fields that will be kept at their default values when you load in the settings to "other" fields. However, if you physically reduce your instruNet network (or replace one device with another), a settings file may try to set a field that no longer exist, and in this case, instruNet might show an alert that states it is having a little trouble.

Buttons The Network Page has the following buttons at the top of the window:

Restore Restores the state of all Fields that existed when the Store button was last pressed. Restore loads a .prf settings file that is kept in an operating system preferences directory. You must press Store to save this file before you press Restore. Restore and Store are very similar to Open & Save, except Open & Save show a File dialog that enables the user to pick the file name & directory; whereas Restore & Store always go to the same file & directory and do not bother the user with file issues.

Store Saves an instrument setup .prf file to disk that contains the state of all Fields. This preferences file is stored within the operating system directory, and is reloaded by pressing the Restore button. File Save and File Open dialogs are not used since the file name and directory are always the same.

Save Saves a .prf instrument setup file to disk that contains the state of all Fields, scripts, and user defined objects. The user specifies the file name and location within the standard File Save dialog.

Open Loads a .prf instrument setup file from disk that contains the state of all Fields, scripts, and user defined objects. The user must pick a previously stored settings file via the standard File Open dialog.

Clear Delete channel settings, scripts, and user defined objects (e.g. popup menu, edit field, etc).

Reset Same as the Clear button, yet also Resets all instruNet hardware.

Calibrate Calibrates all instruNet hardware. To set up this button to also balance strain gages and load cells at 0 uStrain (or 0 Kg force), press the Setup button in the Record page, press the Calibration button, set the Cal Btn field to Bal Gages, and then exit the dialogs. For more details on Calibration options, please Application Note #67.

The Record Page

The Record Page is used to view recorded (i.e. digitized) waveforms in real-time, store these waves to disk, load them from disk, and scroll through them post acquisition. The Record page, illustrated below, is chosen by pressing the Record tab at the base of the instruNet World window.

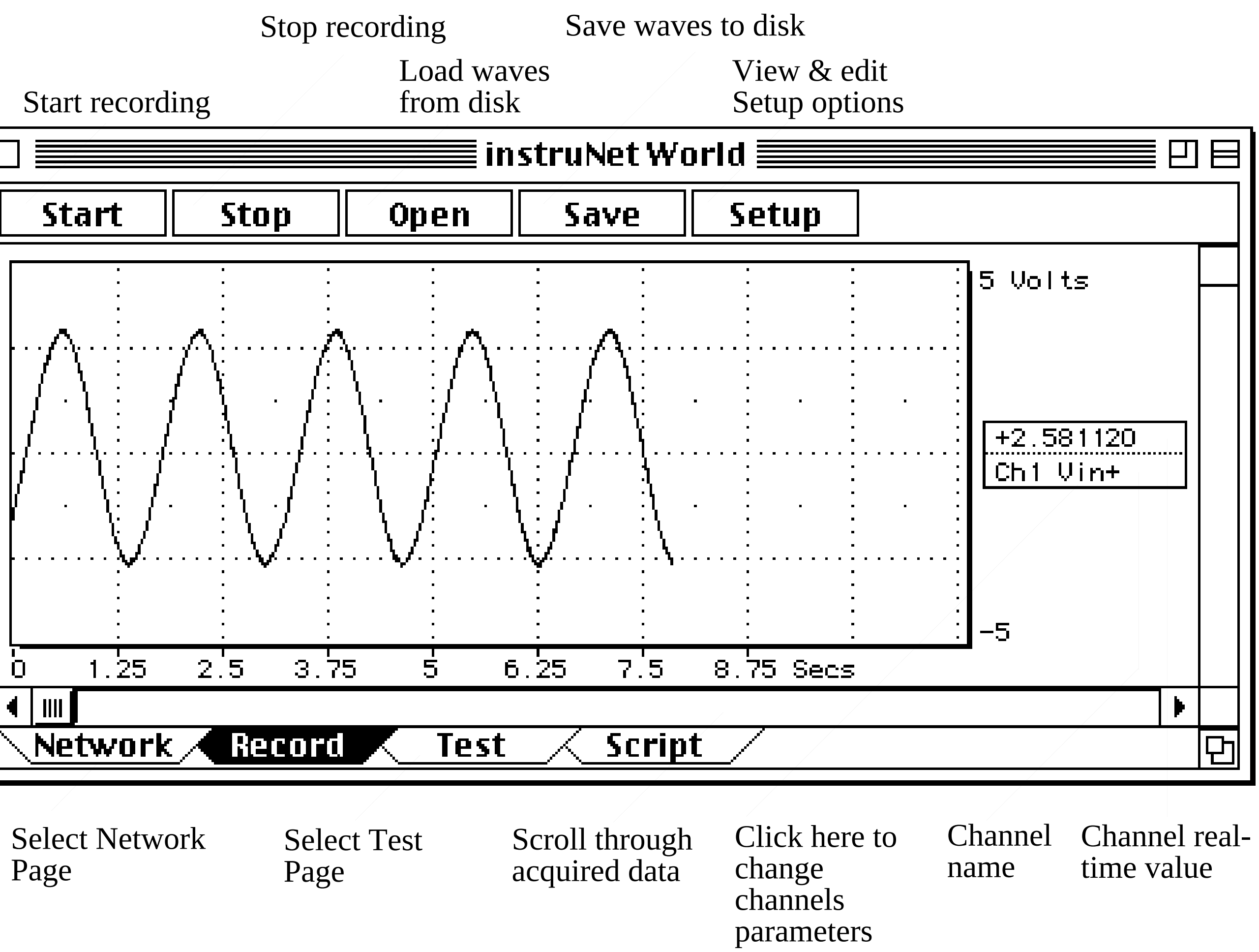


Figure 5.4 The Record Page

Setting Up Channels

A display is shown for each Channel that has previously been turned On. For example, in Figure 5.4, one channel (i.e. "Ch1 Vin+") has been turned On. To learn how to turn Channels On and Off, please consult the "Turning A Channel On" discussion earlier in this chapter. The horizontal scroll bar is used to horizontally scroll through waveforms, post-acquisition, and consequently sets the time of the left edge for all displays. The minimum display height is about 2cm, therefore if more displays exist than there is room, only a subset will be shown, and the user can vertically scroll through the viewed subset with the vertical scroll bar.

Setting Up Displays

To adjust the engineering units value that corresponds to the top and bottom of each display, click on its label (pictured to the left)

at the display right edge. The Display dialog will open, as illustrated below.

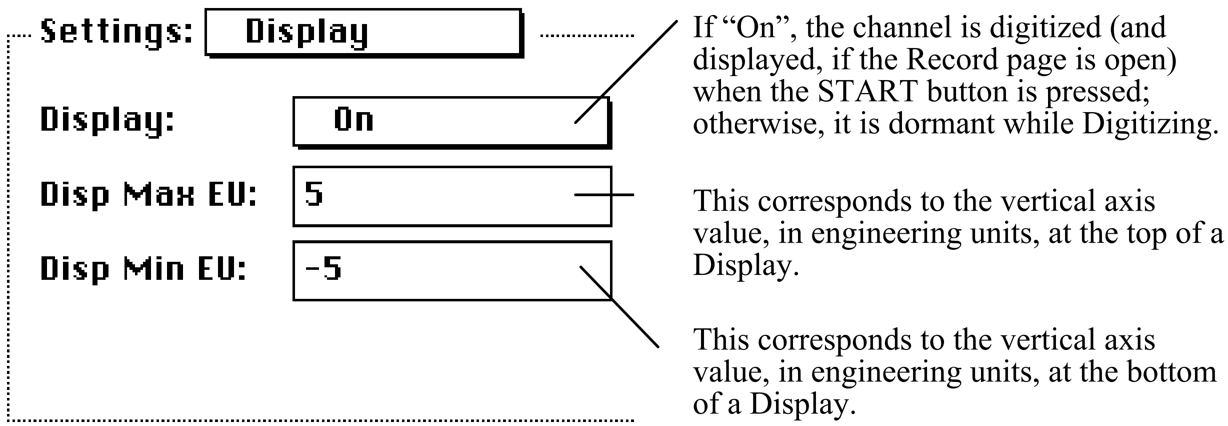


Figure 5.5 Display settings

Oscilloscope or Strip Chart Scan Mode

instruNet supports two Scan Modes, Oscilloscope and Strip Chart, one of which is selected in the Scan Mode popup within the Setup Dialog. Oscilloscope digitizes individual Scans (similar to a real Oscilloscope), whereas Strip Chart links a set of scans together, seamlessly, to form one long waveform (the user does not notice the individual Scans), as illustrated in Figures 5.6 and 5.7. Put a little differently, Oscilloscope waits for a Trigger (specified in the Trigger dialog) before digitizing each Scan, and Strip Chart only waits for a Trigger before digitizing the first Scan.

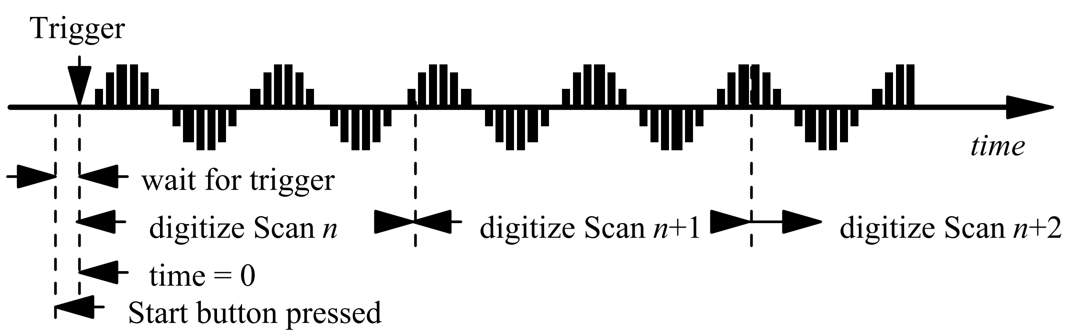


Figure 5.6 Strip Chart Recorder Mode

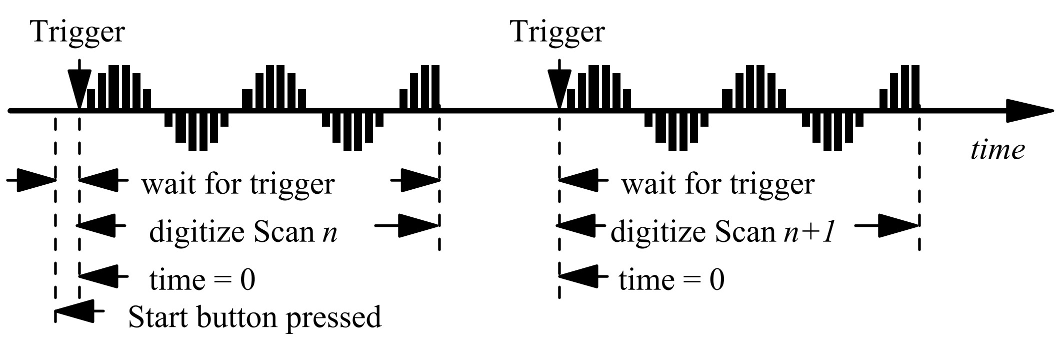


Figure 5.7 Oscilloscope Mode

Oscilloscope mode has two variations, Oscilloscope and Oscillo Queued. Both Oscilloscope and Oscillo Queued acquire and store scans of data in a buffer for processing. In Oscilloscope mode the most recent scan of data in the buffer will always be returned for processing (first in, last out). In Oscillo Queued mode the scans are retrieved from the buffer in sequence (first in, first out).

A typical example of how a data acquisition program using Oscilloscope mode would work is as follows: first the program would wait for the user specified trigger condition and then acquire the first scan of data; next the program would process the acquired scan (i.e. display, analyze, save to disk) while instruNet continued to acquire scans in the background based on the trigger condition, while the processing is going on; next the program would retrieve the most recent scan from the queue, ignoring older scans.

In the Oscilloscope case the scan returned from the buffer will always be the most recently acquired scan, and all other scans in the buffer will be ignored. For example if each scan were 10ms long, the trigger condition was set to none, and the processing took 40ms, then the program would retrieve the first scan, process it and then retrieve the 4th of the 4 queued scans (i.e. the most recent scan) that had accumulated in the buffer while it was processing the first scan. The 3 other older scans would be discarded.

If the example above were done in Oscillo Queued it would work as follows: first the program would wait for the user specified trigger condition and then acquire the first scan of data; next the program would process the acquired scan (i.e. display, analyze, save to disk) while instruNet continued to acquire scans in the background based on the trigger condition; next the program would retrieve the next scan in the queue (i.e. the acquired scan that immediately followed the first acquired and processed scan). In Oscillo Queued mode triggered scans continue to accumulate in the queue and are returned in order.

In Oscillo Queued mode all scans are returned by instruNet for processing but if a lot of processing is required between scans the instruNet buffer can eventually overflow at which point instruNet will return an error message. In Oscilloscope mode the most recent scan is always returned and others are discarded. While scans are discarded in Oscilloscope mode the buffer will not overflow. If the time to process data takes less than the time of a scan then both modes will behave identically. Figure 5.8 below shows the difference between Oscilloscope and Oscillo Queued modes.

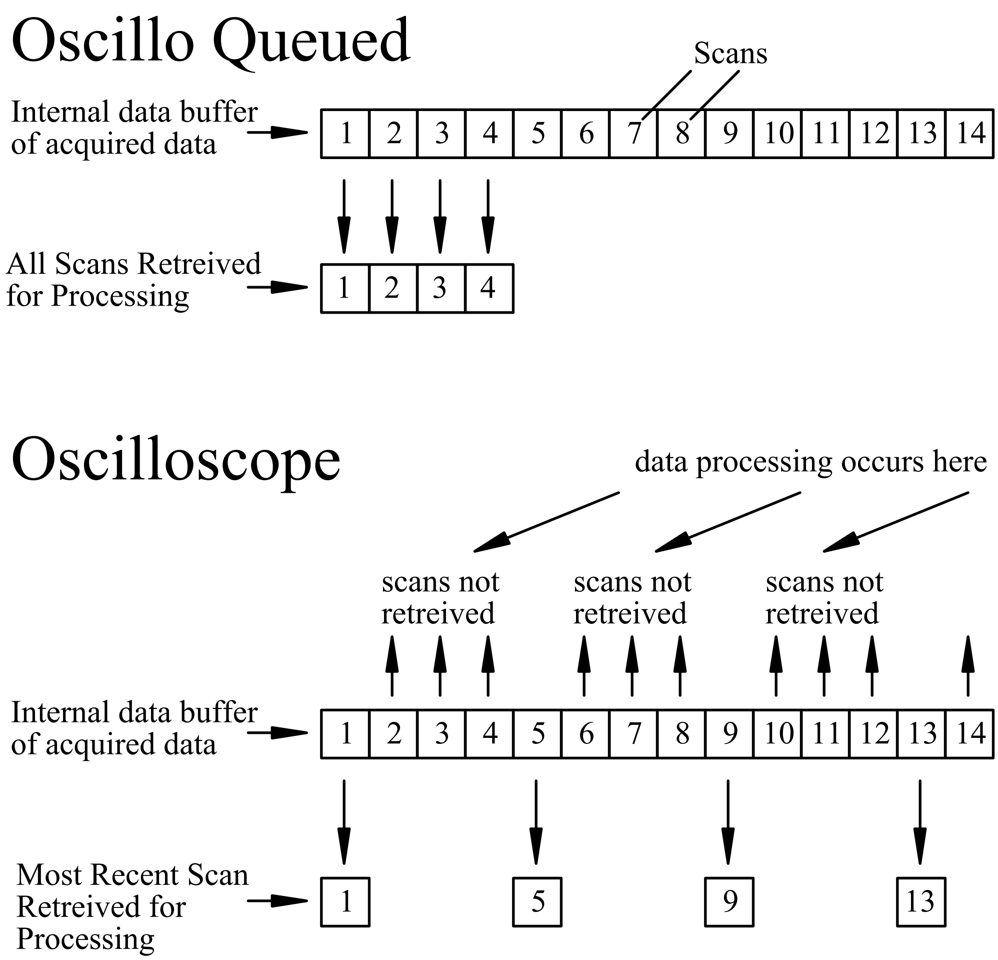


Figure 5.8 Oscilloscope and Oscillo Queued Modes

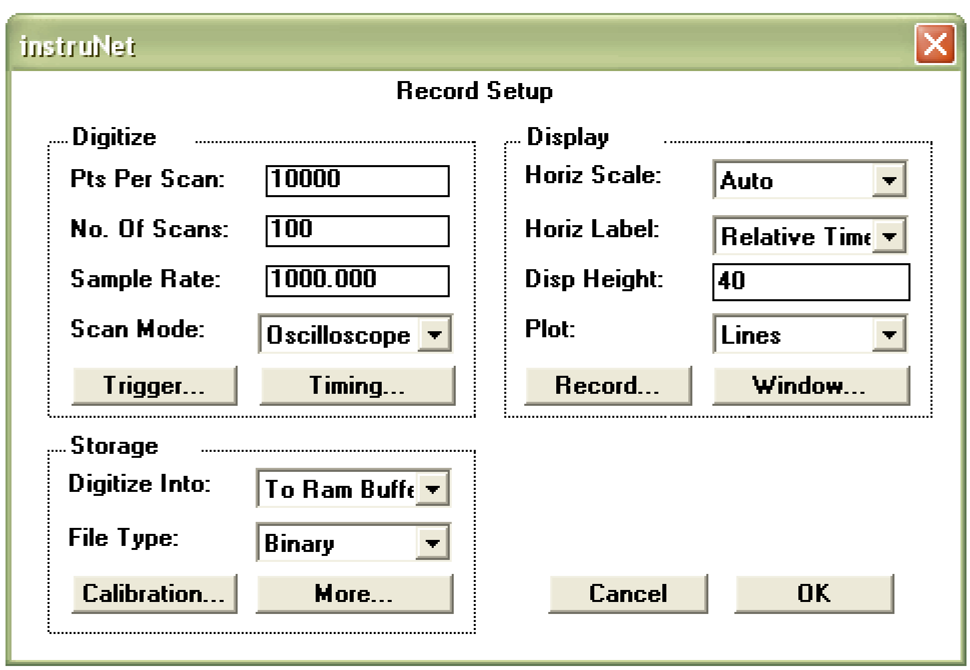


Figure 5.8b The Record Setup Dialog

Record Setup

The Record Setup dialog, illustrated in Figure 5.8b, is used to set the base sample rate, the number of points to be acquired per Scan, the number of Scans to be acquired, the recording mode (i.e. oscilloscope or strip chart recorder), the storage mode, and the display mode. This dialog is described in detail in *Chapter 2, Digitizing Analog Signals into the Computer*.

Timing Options

The Timing dialog, illustrated in Figure 5.8a, is used to set Fields that determine the digitization length and method. This dialog is opened by pressing the Timing button inside the Setup dialog, and mirrors some of the settings inside the Setup dialog.

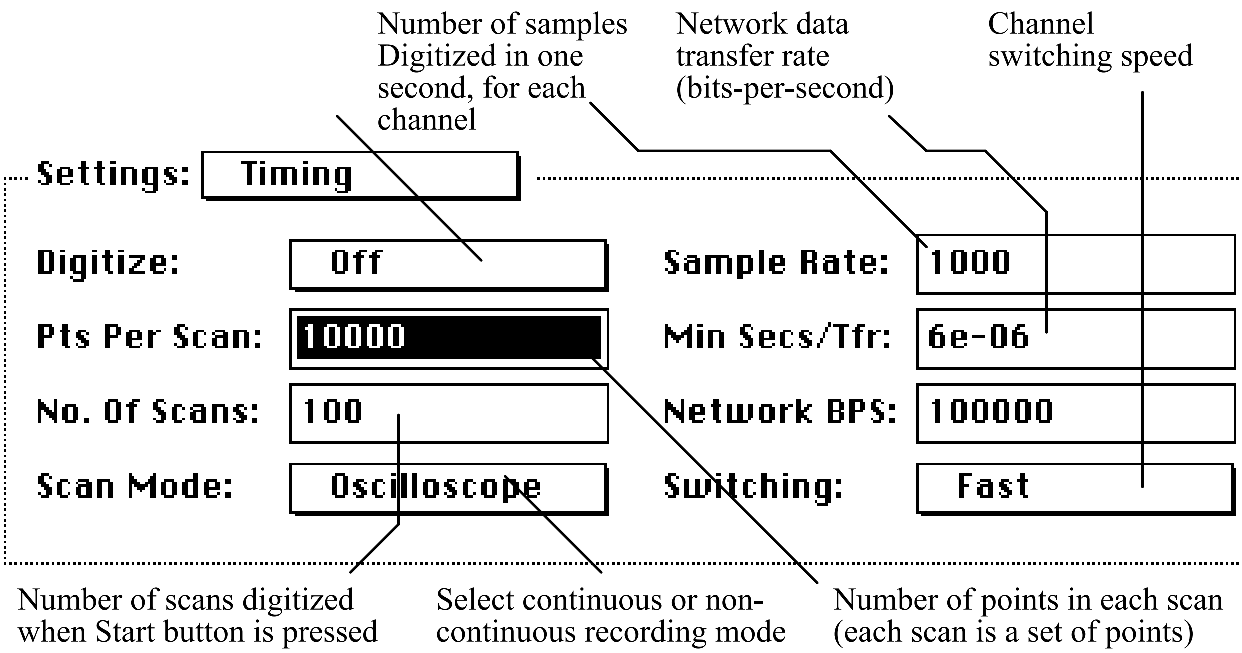


Figure 5.8a Timing Dialog

The Pts per Scan, and the No. of Scans Fields specify the number of points (4bytes per point) that are digitized for each Scan, and the number of scans that are digitized when the Start button is pressed. For example, if you digitize 100 pts/second, with 1000 points per Scan (i.e. 10sec per scan), and 10 scans, then your entire acquisition would consume 100seconds. Pressing the Stop button will halt the digitization process, independent of where it is in its cycle. The Scan Mode field was discussed in detail in the previous *Strip Chart or Oscilloscope Scan Mode* discussion.

The Sample Rate fields specifies the number of points digitized per second per channel. All channels run at this rate unless their % Sample Rate Field in their General settings area has been modified from its default value of 100%. This field enables one to run specific channels at a sample rate less than the master sample rate in the Timing dialog. For example, if the master sample rate is 1000s/sec and a channel's % Sample Rate is set to 25, then the channel will run at 250s/sec. The

Network BPS specifies the network data transfer rate, in units of bits-per-second. On power up, the network is set to the fastest possible rate. The Switching popup is sets the analog channel switching to Fast or Accurate. If Fast is used, the system switches from one channel to another as fast as possible; otherwise, with Accurate, the switching is a little slower, yet provides the amplifiers more time to settle, and is therefore a little more accurate.

Calibrate Options

The Fields in the Calibration Options dialog are used to specify Calibration parameters, as illustrated in Figure 5.8b. This dialog is opened by pressing the Calibration button, inside the Setup Dialog.

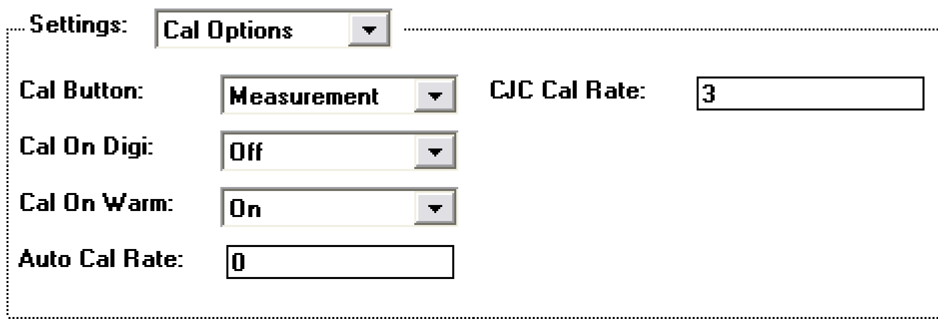


Figure 5.8b Calibrate Options

The Cal Button popup controls the function of the Calibration button in the Network page. It is set to one of:

a. Measurement - instruNet hardware is calibrated when the Network Calibrate button is pressed.
b. Bal Gages - instruNet hardware is calibrated and all strain gages & load cells are balanced when the Network Calibrate button is pressed.
c. Bal Bridges - instruNet hardware is calibrated and all bridges (e.g. strain gages, load cells, potentiometers) are balanced when the Network Calibrate button is pressed.
d. Bal VDividers - instruNet hardware is calibrated and all voltage dividers are balanced when the Network Calibrate button is pressed.

"Balancing" involves reading the voltage across a sensor and placing it into the "Vinit" field within the Constants settings area. This is done to establish a zero point from the sensor, and should be done when the sensor is not receiving a stimulus.

If Cal On Digi is set to On, the instruNet hardware is automatically calibrated when the user presses the Start button and begins to digitize.

If Cal On Warm is set to On, the instruNet hardware is automatically calibrated 10minutes and 25minutes after the instruNet software is first run.

The Auto Cal Rate field sets the rate that instruNet measurement electronics is automatically calibrated, in units of minutes. A full calibration occurs if not digitizing, and a less intensive thermal drift only calibration, described below, occur if digitizing. For example, if this field is set to 5, then instruNet hardware is automatically calibrated every 5 minutes. If set to 0, the Auto-Calibrate feature is disabled. The minimum rate is 0.016 minutes. The primary error from the measurement electronics is an offset voltage of approximately 2uV/C that occurs at the analog voltage measurement input amplifier. This means you will get an offset error of approximately 2uV for every 1 degree C that the amplifier IC heats up. It heats approx 5C above ambient (air around instruNet box) when first turned on and stabilizes after 1hr or so. From here, the IC temperature changes as the room temperature changes (e.g. a 5C increase in room temperature will increase the IC temperature by 5C, which could add voltage measurement offset error of 10uV). When Auto Cal is run, this error is eliminated to 0uV. During digitizing, an abbreviated version of auto-calibration takes place, as described below.

Auto-Calibration While Digitizing
instruNet can automatically calibrate out thermal offset drift errors (e.g. 2uV offset error per 1C change in i100 box temperature) while digitizing (requires iNet32.dll >= 1.40.1.1). These errors are proportional to the temperature change since the last auto-calibration. For example, if we calibrate the i100 at 25C (i100 box temperature), and later the box temperature is 29C, then the Voltage inputs will produce measurement errors between -8uV and +8uV (2uV/C * 4C = 8uV). In many cases; this is not a problem; yet with strain gages, load cells and thermocouples; this could be an issue. In order for this auto-calibration of thermal offset errors (to zero error) while digitizing to operate, all the 40/4KHz analog filters must be off on all channels within the i100xx box being calibrated; and, there must be at least 6ms of time between the total integration time and the sample period. For example, if one digitizes 4 channels with 4ms integration each (16ms total), they would need a sample period of at least 20ms to support this feature. To see which channels are set up for auto-calibrate offset drift while digitizing, set up your recording, press the START button to digitize for several seconds, press STOP, enter "calibrate reportOn" in the BASIC page, press EXECUTE, and view the listed channels. If your list is empty, make sure your AUTO CAL RATE field is >0, make sure all 40/4KHz filters are off, and make sure at least one channel has a range <= +-80mV (large ranges are not affected.

The Cjc Cal Rate field sets the rate that instruNet hardware automatically reads the temperature of the cold junction compensation sensor, in units of minutes, which is used for thermocouple measurements. If set to 0, this feature is disabled. The minimum rate is .016minutes. Reading the cjc sensor can occur while digitizing at slow <1KHs/sec speeds and when the sum of the integration time is less than 40% of the sample period. If the iNet hardware reads the screw temperature (i.e. does a CJC calibrate), and the screw then drifts by X degrees before the next CJC cal, one will incur a thermcouple measurement error by that amount (since the software subtracts the screw terminal temperature when calculating the thermocouple temperature).

For information on how to calibrate while digitizing, how to enable a beep when calibration occurs, and how to minimize thermocouple drift errors, please see Application Note #67.

Display Options

The Fields in the Display Options dialog are used to specify Display attributes, as illustrated in Figure 5.9. This dialog is opened by pressing the Record button, inside the Setup Dialog, and mirrors some of the settings within the Setup dialog.

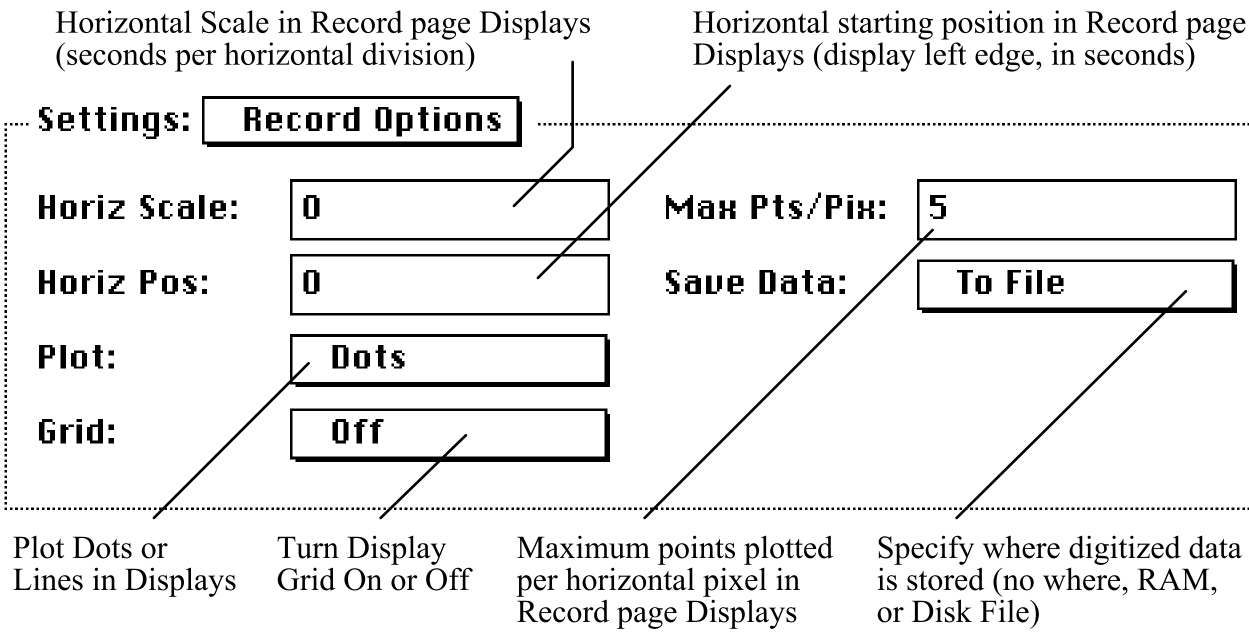


Figure 5.9 Display Options

It is in the Options dialog that one specifies to Plot Dots or Lines, to turn the Grid On or Off, to set the horizontal scale (i.e. time per horizontal division), and to set the maximum plotting density (maximum points plotted per horizontal pixel). Plotting Lines (i.e. connecting points with lines) takes more time than Dots, yet is easier to read if the points are far apart. The Grid take some time to plot as well, and might not be desirable if you are trying to maximize your update rate in Oscilloscope mode. If the horizontal scale (Horiz Scale) is too small (e.g. 10ms/div), the waveform might whip across the screen and be difficult to see in the Strip Chart mode. The maximum points per horizontal pixel (Max Pts/Pix) parameter is useful at plotting only a fraction of the digitized points in cases where you are digitizing ten's of thousands of points and don't want the computer to slow down due to plotting. The Horizontal Position (Horiz Pos) field is very similar to the Horizontal scroll bar in that it defines the time of the left edge of all displays. If Overflow Alert is set to On, an Alert box is shown when a digitize buffer overflows (i.e. it fills faster than it is read); otherwise digitizing proceeds without an Alert.

Saving Data

The Digitize Into option in the Display Options dialog is used to determine whether waveform data is saved to To Ram Buffer, saved to To File, not saved (Off), or determined by the user (User Control). For information on how to save a text file to Excel, post-acquisition, please refer to Appendix III.

Off If the user selects Off, the waveforms are viewed while they are digitized, however you cannot scroll through them post acquisition (since the computer had not saved the data). Since this does not consume memory or disk space, one could digitize many waves at a fast rate for a long time.

To Ram Buffer If one selects To Ram Buffer, each Scan is sent to RAM memory, which is limited by the amount of RAM that has been allocated to the instruNet World application program (or the calling application program in the case of calling the instruNet driver). The buffer in memory holds only one Scan's worth of points, and therefore overwrites the previous Scan with each subsequent Scan. If you are doing Strip Chart work and don't like this, set the No. of Scans field in the Timing dialog to 1, and the Pts Per Scan field to the value required to hold the entire acquisition. If you run out of RAM memory on a Macintosh, you should consider giving your application more by selecting the Application Icon from the Finder, choosing Get Info under File, and then increasing the number in the memory "Minimum Size" field. Each point consumes 4 bytes since they are stored as 32 bit floating point numbers. For example, 100K points would consume 400KBytes. If you have just completed an acquisition with data being sent to RAM and want to save it to disk, press the Save button at the top of the Record page. This will save all waves to disk. You can then load the waves by pressing the Open button.

To File If you want to save all scans to disk, set the Digitize Into option to To File. This will cause all waves to be spooled to disk as they are acquired. If you are doing Strip Chart work, you can then use the horizontal scroll bar to scroll through the entire stream of data. To load in data that was previously saved to disk, press the Open button and then select a previously saved wave in the File Open dialog. Saving to disk is helpful if your waveforms are longer than available RAM.

User Control If User Control is selected in the Digitize Into field, then data will be saved per the settings in the User Ram Buffer, Driver Ram Buffer, and File Settings areas -- which are described later in this manual.

Trigger Options The Trigger dialog is accessed by pressing the Trigger button within the Setup dialog. This dialog is used to specify when digitizing begins after the Start button is pressed, as illustrated in Figure 5.10.

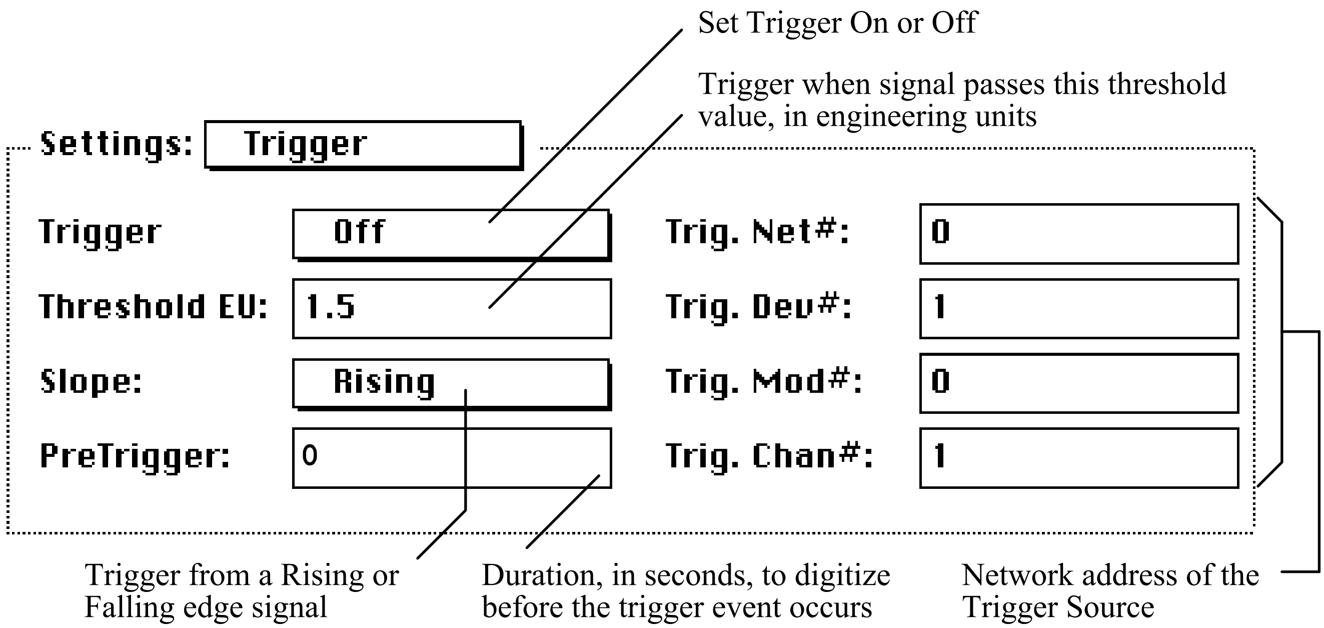


Figure 5.10 Trigger options

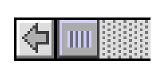
The Trigger field is used to specify no trigger (Off), Automatic trigger, or Normal trigger. Normal trigger will wait forever until the trigger condition is met, whereas Automatic will trigger after several seconds even if the trigger condition is not met. The Threshold EU field is used to set the threshold level, in engineering units, that the incoming signal must pass in order to trigger. The Slope field sets the trigger condition for either a low-to-high pass through the threshold (Rising), or a high-to-low (Falling). The Network number, Device number, Module number, and Channel number fields are used to specify the network address of the input channel that is to be used as the trigger source. For example, {1,1,1,1} would specify the first channel within the first module in the first device tied to the first network.

Getting Ready to Digitize

Before you digitize, you might want to view the settings in the Setup dialog box, which is opened by pressing the Setup button at the top of the Record page. This dialog's fields are described in detail in the previous pages. In summary, one must specify the number of samples digitized per second per channel (i.e. "sample rate"), the number of points per scan, the number of scans, and weather or not the scans are continuous (i.e. Oscilloscope or Strip Chart Recorder mode). The Trigger dialog is used to define when the digitizing begins (e.g. begin when the voltage at Channel #1 exceeds 3Volts). In many cases, there is no Trigger, and the digitization begins when the Start button is pressed.

Digitizing

To begin digitizing, press the **Start** button in the upper-left corner. Waveforms will plot across the screen, in real-time, as they are digitized. To stop digitizing, press the **Stop** button. If the waves had been saved to Ram or Disk (i.e. via the Digitize Into field in Options dialog), then you should be able to scroll through your data with the Horizontal scroll bar.



If Your Sample

If the sample rate is too fast, an alert will appear notifying the user to a buffer

Rate Is Too Fast...

overflow or sample rate problem. This means the computer and/or Controller cannot keep up with the incoming data. In this case, you have several options:

1. Reduce the master sample rate (i.e. Sample Rate field in Setup dialog), or the number of digitized channels.
2. Decrease the % Sample Rate field in a Channel's General settings area to decrease the sample rate for that specific channel. For example, if you run 1 channel at 10% of the master sample rate, then its sample rate will be 1/10 of that of the other channels.
3. Decrease the amount of plotting by reducing the size of the instruNet World window, plotting Dots instead of Lines (i.e. set the Plots field to Dots in the Options dialog), or reducing the plotting density by decrease the Max Pts/Pix field in the Options dialog.
4. Reduce the amount of digital filtering in the Low Pass, High Pass, Band Pass and Band Stop settings areas. Digital filters consume several microseconds to .1us per Filter Order, per point digitized.

Record Buttons

The Record Page sports the following buttons:

Start

Start digitizing channels that have been turned On. Each channel is viewed in its own display. The Digitize Into field in the Setup Dialog determines whether data is saved to RAM or Disk.

Stop

Stop digitizing. If the data had been saved to Ram, then one can now scroll through the most recent Scan, or save the Scan to disk by pressing the Save button. If the data had been saved to Disk in Strip Chart mode, the user can now scroll through the entire set of Scans via the horizontal scroll bar.

Open

The Open button loads waveforms to disk that had been previously saved to disk during acquisition, or by the Save button after acquisition had completed. The standard File Open dialog is used to select the specific wave file. Since waves are saved in their own file, in one directory, it is necessary to only select one file, in order to load the entire set. Once loaded, one can use the horizontal scroll bar to scroll through them.

Save

Saves to disk the current Scan in RAM memory. The user is prompted for a directory name in which to save the set of data, where each wave is saved in its own file, in one common directory. The data is saved in the fast instruNet Binary Merge format, multi-file Binary format, or slow generic *text* format as determined by the File Type field in the Digitizer Channel, within the Driver . If Text Merge is selected in the File Type popup, an additional "Excel Waveform Data.txt" text file is saved that contains all waves in one file, one column of text per wave. This is useful when transferring data to a spreadsheet. Binary Merge saves to disk faster than Binary, yet consumes more RAM memory.

Setup

Opens the Record Setup dialog, as described in the previous pages.

Print

Prints the recorded waveforms to a hard copy printer, at the horizontal scale specified in the Record Setup dialog (i.e. which is mirrored at the front panel horizontal scale popup menu). Multiple pages are printed, as required. For example, if the waves are 100 seconds long, the horizontal scale is set to 1sec/division, and the paper is 20 divisions wide (each grid division is approximately 1 x 1 cm), then 5 pages will be printed. If the horizontal scale is set to AUTO, the entire digitization is printed on one page. Printing requires instruNet World PLUS (iW+).

Test Page

The Test Page, illustrated in Figure 5.12, is used to determine what instruNet hardware is attached to your computer, and to test all instruNet hardware and software. After each test, a report is printed to a miniature text editor within the Test Page. The user can then type notes into this window and save them to disk as a *text* file, to later be opened with the Open button in the Test Page, or a word processor. The Test Page supports the standard text Cut, Copy, and Paste commands, in addition to typing (i.e. Control X, C, and V in Windows). To select the Test Page, press the Test tab at the base of the instruNet World window.

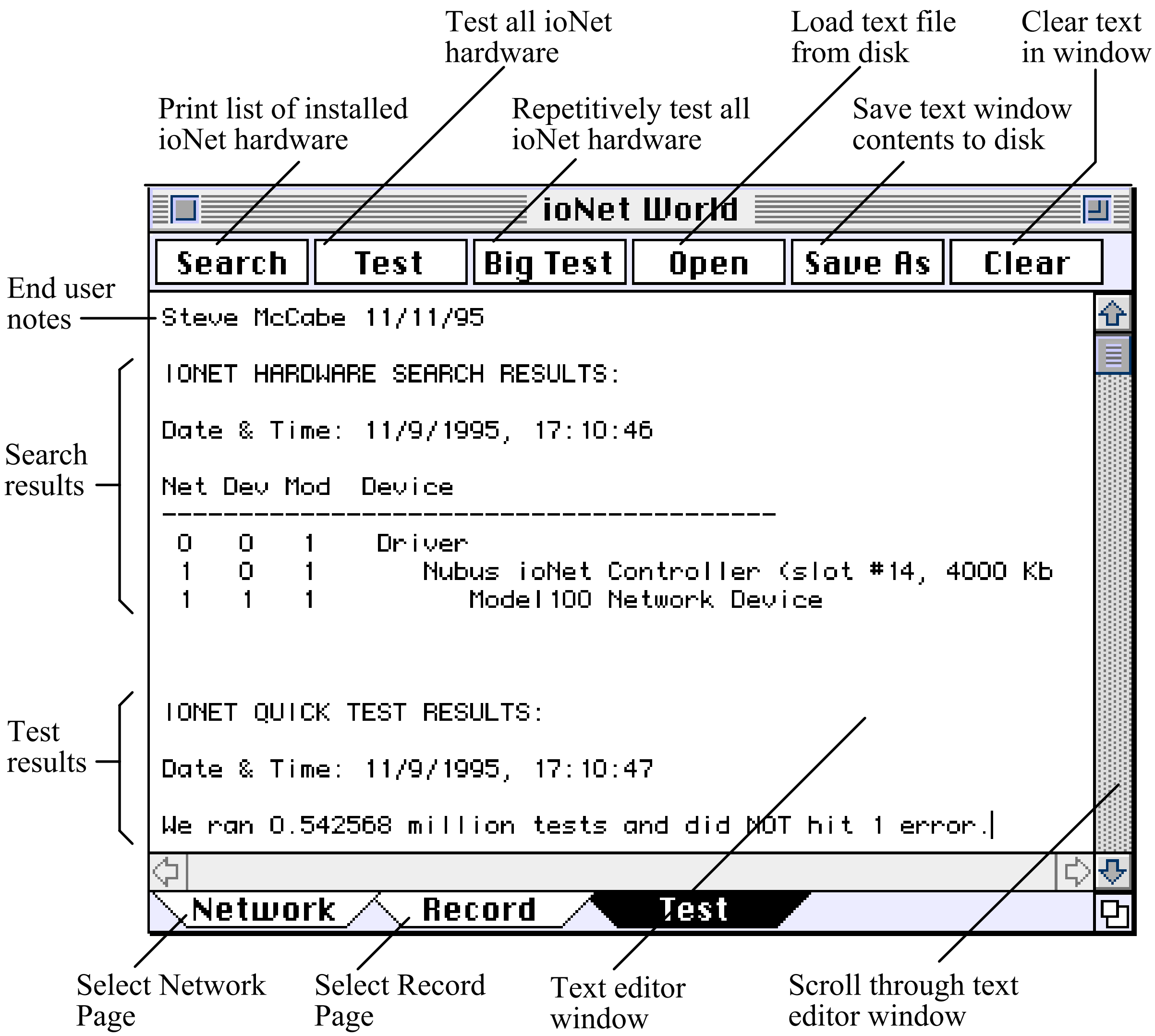


Figure 5.12 instruNet World Test Page

Test Buttons

The Test page sports the following buttons:

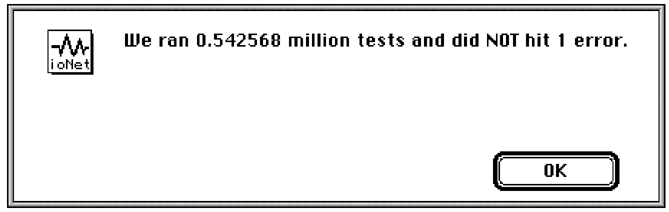
Search

Generates a report, similar to that below, that lists all instruNet hardware attached to your computer. This includes all Controllers and all Devices. If this report disagrees with what you think is out there, please check your cables and consult the "Verify That Your System Is Working Properly" discussion in *Chapter 1*.

```
INSTRUNET HARDWARE SEARCH RESULTS:
Date & Time: 10/23/1995, 19:23:20
Net Dev Mod  Device
----------------------------------------
 0   0   0  Windows XP Version 5.1.2600
 0   0   1    instruNet Driver (DLL 1.36.4, PCI 1.3)
 0   0   1    instruNet World+ license 32-048353-62581
 1   0   1       PCI Controller #iNet-200 (slot #13, 2666KBD, 7us, 95%)
 1   1   1          Device #iNet-100 (SN37526, Cal 8/27/1997, Rev 3, 27.52C, 7us, 4mA)
```

Test

Click the Test button to test all Controllers, Devices, Modules, Channels, and software. This test requires less than a minute and when done displays an alert similar to that shown to the right. A report similar to that shown below is also printed to the Test Page's *text* window. For a list of the error codes please consult *Appendix II instruNet Error Codes.* For tips on trouble shooting, please consult *Appendix I.*




```
INSTRUNET QUICK TEST RESULTS:
Date & Time: 11/8/1995, 17:17: 1
We ran 0.542568 million tests and did NOT hit 1 error.
```

Big Test

Big Test is identical to Test except it runs continuously until you press the mouse button down and hold it down until the test stops. It will run all night if you let it. This is helpful at identifying intermittent problems that occur once in a blue moon.

Open

Loads a text file into the Test page's text editor.

Save As

Saves the contents of the Test page's text editor to disk in a text file.

Clear

Discards the text in the Test Page's text editor.

Report

Prints an internal diagnostic report to be emailed to your instruNet supplier for comment when debugging. After pressing this button, copy the text to the clipboard (Windows Control C), paste it into an email window (Windows Control V), and email to your supplier. A Diagnostic Report can also be generated by pressing the "SAVE DEBUG REPORT TO DISK" button within an error alert. For details, please see Application Notes #69 and #121.

Print

Prints the Test page text to a hard copy printer. Requires instruNet World PLUS (iW+).

The Script Page

To learn more about the iW+ Script page, please select "instruNet World+ Manual" in the "Help" menu within instruNet World, or see Internet URL www.instrunet.com/plus.

Menubar Reference

This manual focuses on the buttons at the top of the instruNet World pages, yet these functions can *also* be accessed in the menubar. For documentation on the menubar and on instruNet World PLUS "iW+" (a version of instruNet World with more features), please select "instruNet World+ Manual" in the "Help" menu within instruNet World, or see Internet URL www.instrunet.com/plus.

Chapter 6, Hardware Reference

This chapter includes specifications for the following instruNet hardware products:

instruNet Controllers
- Model 200 instruNet Controller for Windows 95/98/Me/Nt/2k/Xp PCI bus or Macintosh/PCI bus
- Model 230 instruNet Controller for Windows 95/98/Me/2k/Xp (not NT) PC-Card bus (16bit PCMCIA)

instruNet Network Devices
- Model 100, 100B, and 100HC Analog/Digital I/O Network Device

instruNet Network Accessories
- Model 300 Power Adaptor
- Model 311.x, 312.x, 322.x Power Supplies
- Model 330 Electrical Isolator

Model 200 PCI instruNet Controllers

The Model 200 instruNet Controller boards attach to personal computers via an expansion slot to drive an instruNet network, and to provide several Digital Timer I/O channels at a 34-pin connector, as illustrated in Figure 6.1a. Table 6.1 describes the pins at the Controller 34-pin connector.

Digital I/O Connector
(34pin male header,
.1"grid, .025" posts)

+5V	34	33	+5V
restricted	32	31	restricted
+12V	30	29	+12V
restricted	28	27	restricted
-12V	26	25	-12V
Ch10 Dout	24	23	Ch10 Din
Ch9 Dout	22	21	Ch9 Din
Ch8 Dout	20	19	Ch8 Din
Ch7 Dout	18	17	Ch7 Din
Ch6 Dout	16	15	Ch6 Din
Ch5 Dout	14	13	Ch5 Din
Ch4 Dout	12	11	Ch4 Din
Ch3 Dout	10	9	Ch3 Din
Ch2 Dout	8	7	Ch2 Din
Ch1 Dout	6	5	Ch1 Din
Gnd	4	3	Gnd
Gnd	2	1	Gnd

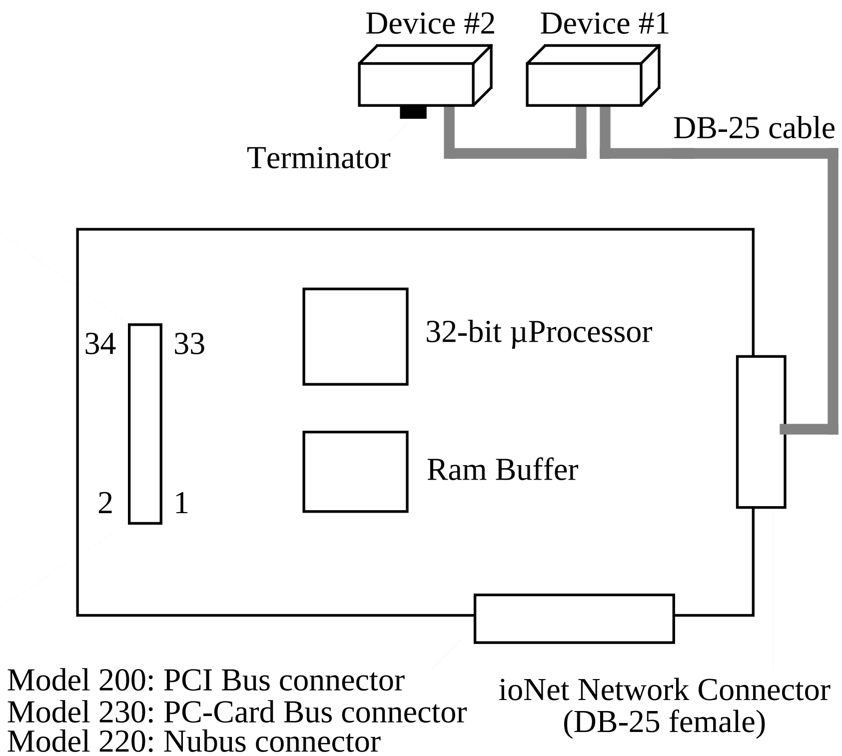


Fig 6.1a Model 200 instruNet Controller Connector Layouts.

Specifications The physical specifications for the Model 200 Controllers are as follows:

Digital I/O Connector:	34-pin male ribbon connector*
Network Connector:	DB-25 female connector
Network Data Throughput:	100Kbits/sec to 4,000Kbits/sec
Dimensions:	4" x 7"
Operating Temperature Range:	0 to 70˚C
Storage Temperature Range:	-25 to +85˚C
Relative Humidity:	Up to 90% (non-condensing)
Power Consumption:	+5V @ 1A max
(does not include network)	+12V @ 10mA max
	-12V @ 10mA max

Controller 34-pin Connector I/O Terminals	Specifications (typical at 25°C)	
Ch1..10 Din The {Ch1 Din...Ch10 Din} digital input terminals are used to sense a digital high or digital low state; or to measure the period of a changing digital signal. The inputs are protected against high voltages up to +6Volts and down to -6V. To measure the logic level or the period of an input signal, connect it directly to a Controller Din terminal and then connect the signal source ground to a Controller Gnd terminal. For details, please see *Ch2, Working with Controller Digital Timer I/O Channels*.	**Digital Inputs** Digital Input Port Input Levels **Period Measurement** Measured Period Range Resolution	 1 non-latching input bit per channel V_{IH} = 3.5V min ... 6V max V_{IL} = .75V max ... -6V min I_{IH} = -120µA, V_i = 3.5V I_{IL} = -.6mA max. $V_{HYSTERESIS}$ = .5V 1Cycle: 3us-16ms ± .25us or 3ms-32sec ± 1ms >1Cycle: 3us - 4.1sec ± .25us or 3ms-16Ksec ±1ms 1Cycle: 16bits >1Cycle: 24bits
Ch1..10 Dout The {Ch1 Dout...Ch10 Dout} digital output terminals can be set to a high or low state; or set up to output a digital clock signal. To wire a Dout terminal to a device, connect the Dout terminal to the device input, and the Controller Gnd terminal to the device Ground. For details, please see *Ch2, Working with Controller Digital Timer I/O Channels*.	**Digital Outputs** Digital Output Port Logic Output Levels **Clock Outputs** Duty Cycle Range Period Range Time Base Accuracy Output Signal	 1 latching output bit per channel TTL-compatible V_{OH} = 2V min ... 5V max V_{OL} = 0.5V max ... 0V min I_{OH} = -12mA max. I_{OL} = 24mA max. .01% to 100% 5µsec to 536 sec (programmable) +- 0.01% TTL Compatible
+5V, -12V, +12V These terminals can be used to power external devices. **The maximum allowed current, listed to the right, must not be exceeded; otherwise damage might occur to the Controller or computer.** To power an external device, one must attach a wire from one of the Controller power terminals to the external device's power input, and also attach a wire from the Controller's Gnd terminal to the external device ground.	**Power Available to User** +5V +12V -12V	 300 mA max. 50 mA max. 50 mA max.

Table 6.1 Model 200 Controller 34-pin Connector Technical Specifications .

*The Model 230 PC-Card Controller does not include the 34-Pin Digital I/O Connectors, and its signals.

Model 230 PC-Card instruNet Controller

The Model 230 PC-Card (16bit PCMCIA) instruNet Controller attaches to a personal computer via a 16bit PC-Card slot to drive an instruNet network, as illustrated in Figure 6.2. The i230 has the same capabilities as the i200 PCI controller, except it requires an external power supply (e.g. iNet-312-8), it does not provide 10 counter/timer channels, and it is not compatible with Windows NT.

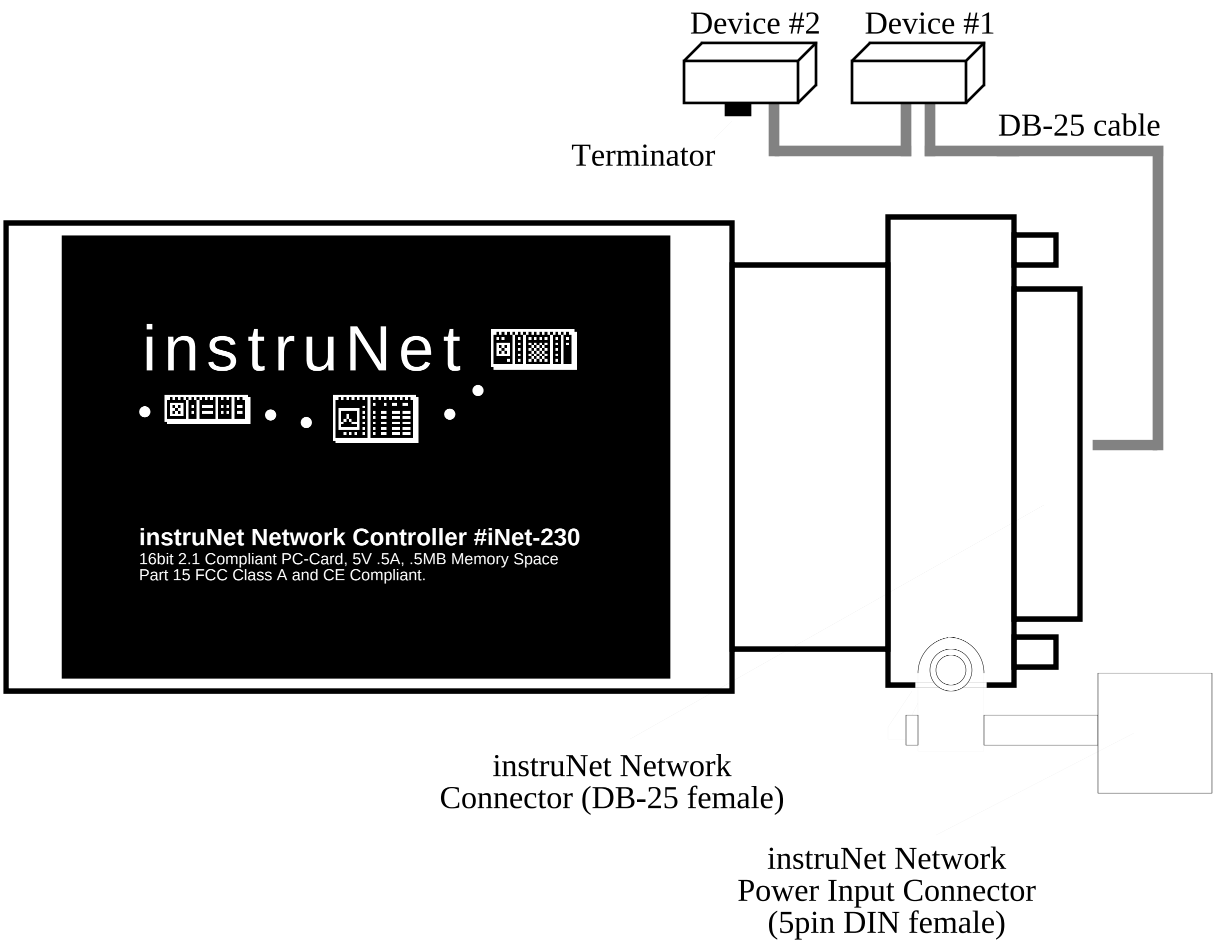


Fig 6.1b Model 230 instruNet Controller .

Specifications The physical specifications for the Model 230 Controller is as follows:

Network Connector:	DB-25 female connector
Network Power Input Connector:	5pin DIN female connector
Network Data Throughput:	100Kbits/sec to 4,000Kbits/sec
Dimensions:	2.1" x 3.4"
Temperature Range:	Operating 0 to 70˚C, Storage -25 to +85˚C
Relative Humidity:	Up to 90% (non-condensing)
Power Consumption:	+5V @ .5A max

Model 100, 100B, and 100HC Network Devices

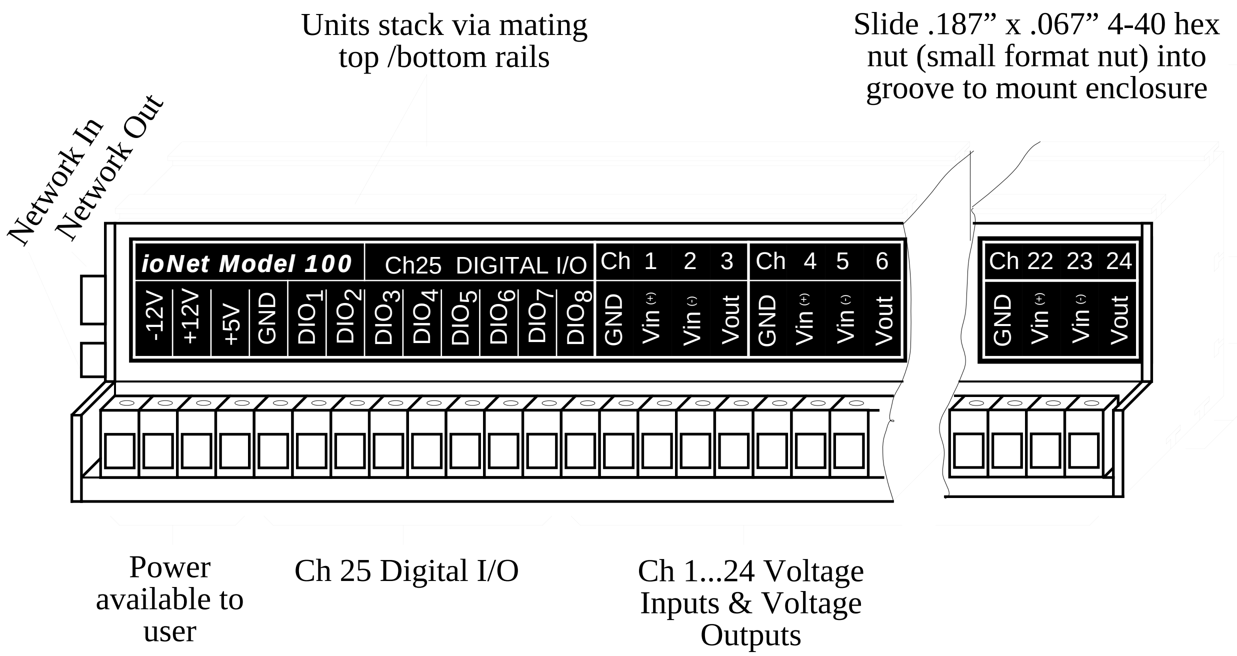


Fig 6.2 Model 100xx Network Device Connector Layout.

Model 100xx Connector I/O Terminals	**Specifications[1]**	
Ch 25 Digital I/O	**Digital I/O**	
This is a digital I/O port consisting of 8 individual lines, each of which can be configured as either a digital input channel or a digital output channel. When configured as an input, a channel can be used to sense a digital high (3.2V to 10Volts) or digital low (0 to 1Volts). When configured as an output a channel can be set high (2.0V min) or low (.5V max). The inputs are protected against high voltages up to and +12Volts and down to -12V.	Digital Input/Output	8 non-latching input bits and 8 latching output bits at 8 bidirectional screw terminals
	Input Levels	V_{IH} = 3.2V min ... 12V max V_{IL} = 1.0V max ... -12V min I_{IH} = -200µA, V_i = 3.2V I_{IL} = -.5mA max.
	Output Levels	V_{OH} = 2V min ... 5V max I_{OH} = -.5mA max. I_{OL} = 500mA max, V_O=1.7V I_{OL} = 50mA max, V_O = .7V

Table 6.2 Model 100, 100B, and 100HC Technical Specificati

Model 100xx Connector I/O Terminals	Specifications[1]

Ch 1...24 Vin+/Vin-

These are voltage input ports that can be wired for any of the sensors described in Chapter 5 Sensor Reference. When wiring single-ended inputs the Vin+ and Vin- terminals function identically.

Analog Inputs

A/D Conversion Time	4µs min
A/D Resolution	14-bit
Voltage Input Range	+- 5V +-.6V +- 78mV max, +- 80mV typ +- 8mV max, +- 10mV typ
Amplifier Gain	1, 8, 64, 512
Number of Channels	16se/8di
System Throughput	166Ksamples/second max

Measurement Accuracy

Voltage Range	Integration (seconds)	Channel Switching	Voltage Accuracy
±5V	1ms	(either)	±.75mV
	0sec	Accurate	±1.5mV
	0sec	Fast	±2.5mV
±0.6V	1ms	(either)	±75µV
	0sec	Accurate	±150µV
	0sec	Fast	±225µV
±80mV	1ms	(either)	±15µV
	0sec	Accurate	±45µV
	0sec	Fast	±60µV
±10mV	1ms	(either)	±10µV
	0sec	Accurate	±30uV
	0sec	Fast	±50µV

Signal To Noise Ratio	78dB
Differential Linearity	+- 1.5 LSB
Integral Linearity	+- 2 LSB
Input Over voltage Protection	+- 15 V (power on or off)
Input Impedance	10MΩ ±1%, 3pf
Common Mode Voltage (CMV)	+- 5V min.
Common Mode Rejection (CMR)	+- 80dB
Temperature Drift	Gain: +- 5ppm/˚C of FSR Offset: Self-cal'ed to 0
Time Stability	Gain: 27ppm/1yr typ Offset: Self-cal'ed to 0

Ch 1...24 Vout

These are voltage output ports that can be used for purposes such as stimuli and sensor excitation.

Analog Outputs

D/A Resolution	8-bit
Number of Channels	8
Output Voltage Range	#100/100B: ±5V, 4mA, .001uF #100HC: ±5V, 15mA, .01uF
Output Protection	Short-to-ground continuous
Output Settling Time	4µs (to +-1/2 LSB,+-5V step)
Analog Output Accuracy	+-0.4%
Digital Coupling	+-20mV
Gain Drift	+- 10ppm/˚C of 5V FSR
Offset Drift	+- 5µV/˚C

Table 6.2 Model 100, 100B, and 100HC Technical Specifications.

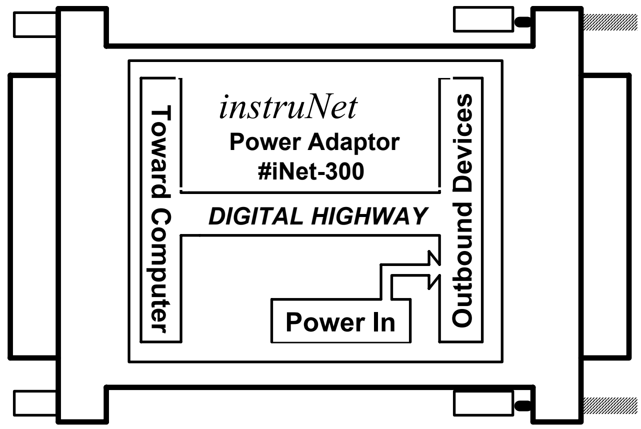
Model 100xx Connector I/O Terminals	Specifications[1]	
+5V, +12V, -12V These screw terminals can be used to power external devices. The maximum allowed current specified in the table to the right should not be exceeded, else the controller and/or the computer could be damaged. To power an external device, run a wire from a voltage output terminal (e.g. "+5V) to the device and also run a wire from one of the ground terminals (i.e. "GND") to the device ground.	+5V +12V -12V	100 mA max. 30 mA max. 30 mA max.
Network Interface The Model 100 cables to a Controller and/or other Network Devices via DB-25 cables.	Network Utilization: Compatibility: Network Connector:	Occupies 1 physical address in instruNet Network All instruNet Controllers and Network Devices DB-25 male connector (input), DB-25 female connector (out)
Physical/Environmental All I/O signals cable to the Model 100 via screw terminals.	I/O Connector: Dimensions: Operating Temperature Storage Temperature Relative Humidity:	Screw Terminals 1.8" x 4.2" x 9" 0 to 70˚C -25 to +85˚C Up to 90% (non-condensing)
Power Power consumed by the Model 100 Network Device without additional Network Devices attached, without a terminator, and without outputs loaded. Total system power consumption is the sum of the power consumed by the Controller and each attached Network Device.	Power Consumption:	+5V @ 180mA max +12V @ 80mA max -12V @ 80mA max

Table 6.2 Model 100, 100B, and 100HC Technical Specifications.

[1]Measurement Accuracy
0-70°C, no condensation, #iNet-100xx Rev 3, temperature has not changed since self-calibration, Accuracies are typical within 2 standard deviations. Accuracy is effected by the Channel Switching field (i.e. set in the Setup dialog to Accurate or Fast) and the integration time. If the Channel Switching field is set to Accurate, the signal is given more time to settle after the channel multiplexor is switched. Accurate switching has a lower maximum sample rate than Fast switching.

Integration time is independently programmable for each channel, and reduces noise by averaging many samples. Also, if more than 125 samples are averaged, a dithering noise generator automatically turns on to add noise to the input signal. This increases the accuracy further by using more a/d codes to establish the input volt

Model 300 Power Adaptor



instruNet controllers provide power to external devices; however this power is sometimes inadequate due to too many devices on the network, or a network length that induces an unacceptable voltage drop between the controller and the devices. Subsequently, one can insert a Model 300 Power Adaptor inline the instruNet network, preferably close to the Devices receiving power, and connect an external power source (e.g. #iNet-311.x, 312.x, 322.x Power Supply) to the Model 300 External Power Source input connector. Multiple Model 300's can be placed in one network (e.g. place one Model 300 every 4 Devices, in a 16 Device network).

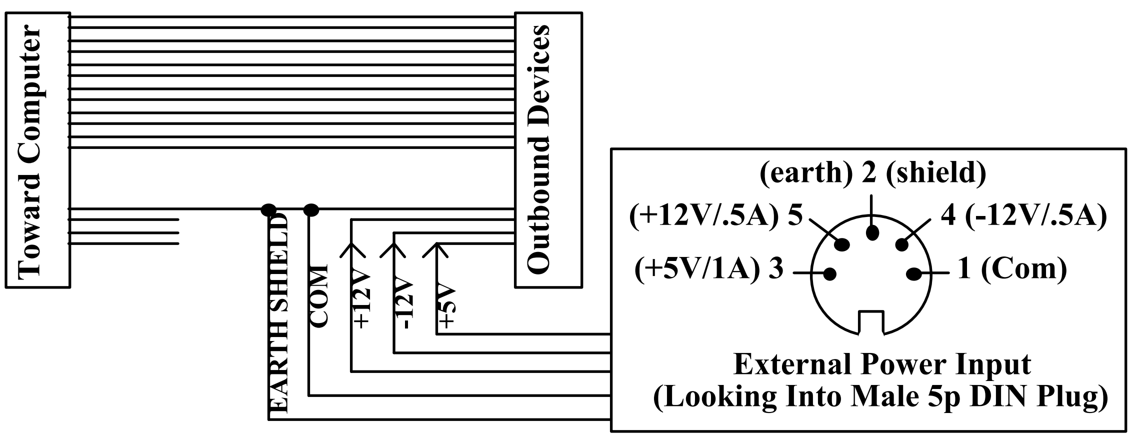


Fig 6.3 Model 300 Power Adaptor Block Diagram

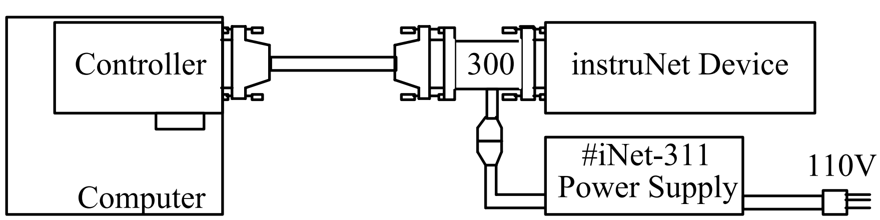


Fig 6.4 Example Application of Model 300 Power Adaptor

Model 311.x, 312.x, 322.x Power Supplies

The following power supplies provide a 5pin Din output connector compatible with the instruNet #iNet-300 Power Adaptor, #iNet-330 Electrical Isolator, and #iNet-230 PC-Card. The 5pin DIN connector pin out is as follows: 1: Com, 2: n/c or earth GND, 3: +5V, 4: -12V, 5: +12V.

Part #	Plug	VAC Input	Output	RECOMMENDATION
#iNet-312.8	USA 3-Prong	120/220	5V/2A, +-12V/.8A	
#iNet-312.8eu	Euro 2-Prong	120/220	5V/2A, +-12V/.8A	
#iNet-311.2	USA 3-Prong	120	5V/.7A, +-12V/.24A	*i311.x and i322.x*
#iNet-311.5	USA 3-Prong	120	5V/2A, +-12V/.5A	*are discontinued -*
#iNet-322.5	Euro 2-Prong	220	5V/2A, +-12V/.5A	*buy i312.8 instead*

Power Plugs

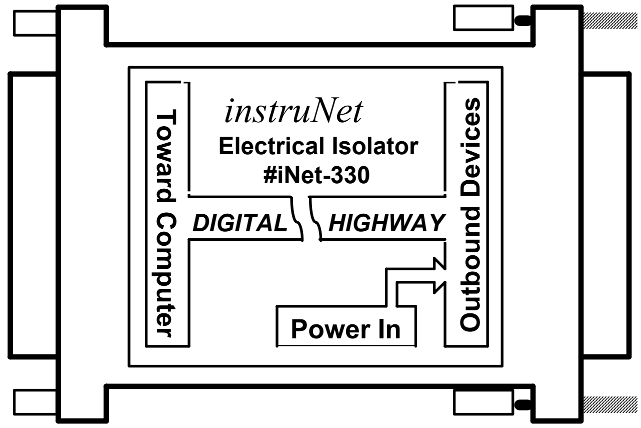
USA 3 Prong Plug
USA 3 prong plugs mate with USA 110VAC wall sockets.

EURO 2 Prong Plug
Euro 2 prong CE plugs include a receptacle for an earth pin and they mate with 220VAC wall sockets in Germany, France, Italy, Spain (yet not UK).

Japan 110VAC Plug
Connecting to 110VAC Sockets in Japan requires replacing the iNet-3xx.y plug with a Japanese plug.

If the power plug does not match your wall socket...
If the power plug doesn't fit, we recommend that it be removed and replaced with one that does.

Model 330 Electrical Isolator



The Model 330 Electrical Isolator provides 1000 Volts of optical isolation at one point within an instruNet network. This is often used to eliminate ground loops between the computer and items under test, and to reduce noise that is transmitted from the computer to sensors. When measuring small voltages (e.g. <10mV), optical isolation is sometimes critically important. The Model 330 Electrical Isolator is very similar to the Model 300 Power Adaptor, described earlier, except for its isolation capability. An external power source (e.g. #iNet-312.8) must be connected to the 5pin DIN Power Source Connector, to provide power to outbound devices. Multiple Model 330's can be placed in one network (e.g. place one Model 330 every 4 Devices, in a 16 Device network).

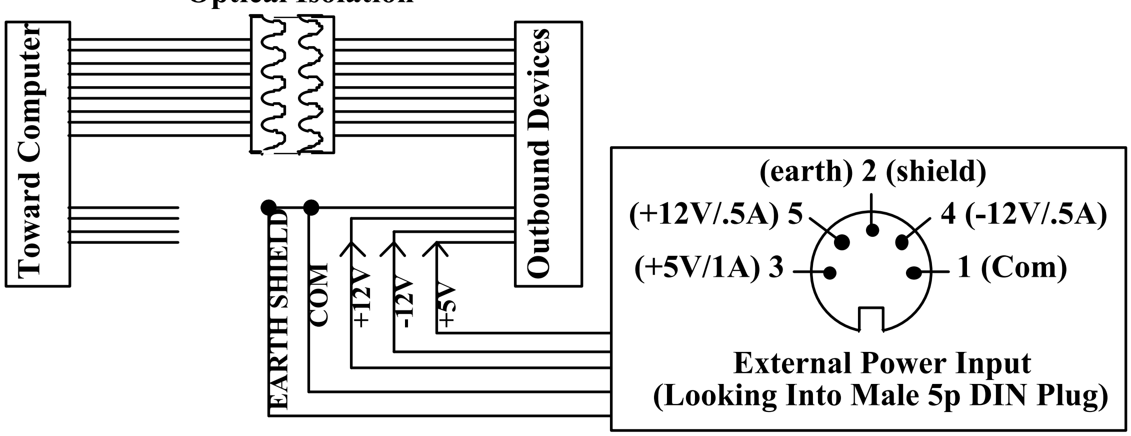


Fig 6.5 Model 330 Electrical Isolator Block Diagram

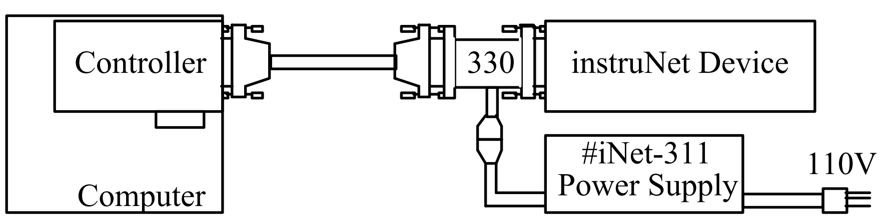


Fig 6.6 Example Application of Model 330 Electrical Isolator

Isolating From Earth Ground

The External Power Input pins 1 and 2 are both tied to the outbound instruNet device ground (case and pcb). If this is also tied to the 3rd prong of the power supply AC input connector, and you want to isolate instruNet from this earth ground, then you would need to place a 2-to-3 prong Adaptor at the power supply input connector (or rip the 3rd prong out of the plug with a pair of strong pliers).

Wild Grounds

If the outbound devices are not seen by instruNet World, try temporarily holding a wire between the outer shell of the isolator inbound and outbound DB-25 connectors. If the outbound devices are then seen by instruNet World (after Reset), then this indicates that a very high frequency exists between the two grounds and is causing problems. The Model 330 is designed to sustain a signal between the two grounds that is 50V peak-to-peak with a 1000V/us slew rate (e.g. slews 1Volt in 1ns); and this is adequate is most applications.

Slower Speeds

Since the optical isolators introduce delays, the Model 330 reduces the speed of the instruNet network from 4M bits/sec to 1.33M bits/sec, on short networks. In a worse case scenario, this would reduce the maximum sample rate 3 to 1.

Chapter 7, Channel Reference

The following table summarizes the Channels provided by each instruNet hardware Device, the Settings offered by each channel, and the Fields within each Settings group.

For documentation on channel settings unique to instruNet World PLUS (e.g. Panel Meters, Control), please select "instruNet World+ Manual" in the "Help" menu within instruNet World, or see Internet URL www.instrunet.com/plus.

Hardware Device	Channel	Setting Group	Fields
Model 200 PCI, Model 230 PC-Card Controllers	Ch1...10 Digital Timer I/O	General	Value I/O, Units Label, User Name % Sample Rate
		Display	Digitize On/Off, Display Min, Display Max, View Show/Hide
		Timer	Function, Clk Period, Clk Out Hi, Measure, Meas. Resol., Meas. Cycles
	Ch 11 Time	General	Time Secs, Units Label, User Name % Sample Rate
		Display	On/Off, Max EU, Min EU
	Ch 12 Digitizer	General	Value, Units Label, User Name, % Sample Rate
		Timebase	Digitize Off/On, Pts Per Scan, No. Of Scans, Min sec/tsfr, Sample Rate, Scan Mode, Switching, Network BPS
Continued...		Trigger	Trigger Off/On, Threshold EU, Slope, Trig. Net#, Trig. Dev# Trig. Mod#, Trig. Chan#

Table 7.1 Channel Reference.

Hardware Device	Channel	Setting Group	Fields
Model 100 Analog & Digital I/O System	Ch1 Vin+, Ch2 Vin-, Ch4 Vin+, Ch5 Vin-, Ch7 Vin+, Ch8 Vin-, Ch10 Vin+, Ch11 Vin-, Ch13 Vin+, Ch14 Vin-, Ch16 Vin+, Ch17 Vin-, Ch19 Vin+, Ch20 Vin-, Ch22 Vin+, Ch23 Vin-	General	Value Input, Units Label, User Name % Sample Rate
		Hardware	Sensor, Wiring, Low Pass, Integrate Range
		Constants	Ro, Rshunt, Vout, Vinit, alpha delta/Rlead, GF, υ _Poisson
		Display	Digitize On/Off, Display Min, Display Max, View Show/Hide
		Lowpass Filter	Filter type, PassB Ripple, StopB Attn Filter Order, PassB F1 Hz, StopB F1 Hz
		Highpass Filter	Filter type, PassB Ripple, StopB Attn Filter Order, PassB F1 Hz, StopB F1 Hz
		Bandpass Filter	Filter type, PassB Ripple, StopB Attn Filter Order, PassB F1 Hz, StopB F1 Hz, PassB F2 Hz, StopB F2 Hz
		Bandstop Filter	Filter type, PassB Ripple, StopB Attn Filter Order, PassB F1 Hz, StopB F1 Hz, PassB F2 Hz, StopB F2 Hz
		File	File, Digitize, File Name, Command Scan Num, 1st Pt Num, Num Pts
		Driver Ram	Digitize On/Off, Buffer Addr, Ptr Byte Size, Scan Num In, Pt Num In, Scan Num Out, Pt Num Out
		User Ram	Digitize On/Off, User Ptr, Ptr Bytes Size, Scan Num In, Pt Num In, Scan Num Out, Pt Num Out
	Ch3, Ch6, Ch9, Ch12, Ch15, Ch18, Ch21, Ch24 Vout	General	Value Output, Units Label, User Name, % Sample Rate
		Display	Digitize On/Off, Display Min, Display Max, View Show/Hide
	Ch25 DIO	General	Value Output, Units Label, User Name, % Sample Rate
		Display	Digitize On/Off, Display Min, Display Max, View Show/Hide
		Hardware	Digital Out, Direction

Table 7.1 Channel Reference.

Chapter 8, Settings Reference

This chapter summarizes the Settings field groups used by the instruNet field hierarchy. Each group may be used many times; for example, the Display group contains 4 fields (Digitize On/Off, Display Min, Display Max, View Show/Hide) and is used with 25 Channels in the Model 100 hardware device. Each Settings group contains 1 to 8 fields, as described in the following table. And each field can be set to a different value, as described in this table.

For Programmers:
Each Setting Group has an associated settingType value, which are passed to the iNet() subroutine in the 'settingGroupNumOrType' argument. These are listed in the 1st column of the table in subscript form. For example, the settingType of the Filter Settings group is -11. These are also listed in the ion_settingGroupType area of the interface .h file.

Each Field has an associated fieldNum value, which are be passed to the iNet() subroutine in the 'fieldNum' argument. These is listed in the 2nd column of the table in subscript form. For example, the fieldNum of the PassB Ripple field within the Filter Settings group is 2. These are also listed in the fldnum_.... areas of the interface .h file.

Some Fields have several possible values which are passed to the iNet() subroutine in the 'ptrToArg' argument. These are listed in the 3rd column of the table in subscript form. For example, the fieldValue for the Elliptic option of the Filter field within the Filter Settings group is 2. These are also listed in the fldnum_.... areas of the interface .h file.

Settings Group settingType	Field fieldNum	Field Description fieldValue
Bandpass Filter -10		Defines a digital bandpass filter. For more information, please refer to *Chapter 2, Working with Digital Filters*.
	$Filter_1$	Selects filter model = {Off_1, $Elliptic_2$, Chebyshev P_3, Chebyshev S_4, or $Butterworth_5$ }.
	PassB $Ripple_2$	Specifies maximum allowable passband ripple, in dB.
	StopB $Attn_3$	Specifies the minimum stop band attenuation, in dB.
	Filter $Order_4$	Displays filter order. Automatically determined by the filter design, which is based on the specified criteria.
	PassB F1 Hz_5	Specifies lower cutoff frequency of the band that is passed. Frequencies between PassB F1 Hz_5 and PassB F2 Hz_7 are passed.
	StopB F1 Hz_6	Frequencies below StopB F1 Hz_6 are attenuated.
	Pass B F2 Hz_7	Specifies high cutoff frequency of the band that is passed. Frequencies between PassB F1 Hz_5 and PassB $F2_7$ Hz are passed.
	Stop B F2 Hz_8	Frequencies above StopB F2 Hz_8 are attenuated.
Bandstop Filter -11		Defines a digital bandstop filter. For more information, please refer to *Chapter 2, Working with Digital Filters*.
	$Filter_1$	Selects filter model = {Off_1, $Elliptic_2$, Chebyshev P_3, Chebyshev S_4, or $Butterworth_5$ }.
	PassB $Ripple_2$	Specifies maximum allowable passband ripple, in dB.
	StopB $Attn_3$	Specifies the minimum stop band attenuation, in dB.
	Filter $Order_4$	Displays filter order. Automatically determined by the filter design, which is based on the specified criteria.
	PassB F1 Hz_5	Frequencies below PassB F1 Hz_5 are passed.
	StopB F1 Hz_6	Specifies lower cutoff frequency of the band that is stopped. Frequencies between StopB F1 Hz_6 and Stop B F2 Hz_8 are stopped (i.e. attenuated).
	Pass B F2 Hz_7	Frequencies above PassB F2 Hz_7 are passed.
	Stop B F2 Hz_8	Specifies higher cutoff frequency of the band that is stopped. Frequencies between StopB F1 Hz_6 and Stop B F2 Hz_8 are stopped (i.e. attenuated).

Table 8.1 Channel Setting Group Reference

Settings Group settingType	Field fieldNum	Field Description fieldValue
Constants -4		Defines constants used to calculate engineering units when measuring sensors (e.g. RTD, strain gage, etc). For more information, please refer to *Chapter 3, Connecting To Sensors.*
	Ro_1	Specifies value of bridge completion resistor, unstrained strain gage, load cell w/ 0 force, potentiometer, or resistance of an RTD at 0°C. Units are ohms.
	$Rshunt_2$	Specifies value of shunt resistor in voltage divider or current measurement circuit. Units are ohms.
	$Vout_3$	Sets excitation voltage in bridge/voltage divider circuits. Units are Volts.
	$Vinit_4$	Specifies voltage across an unstrained bridge (e.g. strain gage, load cell); the voltage of a pot in the CCW position; or the voltage out of an accelerometer at 0 G's.
	$alpha_5$	Specifies temperature coefficient of an RTD at 0°C (typically .00385 for American RTD's, and .00392 for European RTD's), as specified by the manufacturer of the RTD (ohms/ohms/C units); maximum force (Kg) in the case of a load cell; or maximum acceleration (G's) in the case of an accelerometer.
	delta, $Rlead_6$	When connecting to an RTD, delta is the Callendar-Van Dusen delta constant (typically 1.492). This constant is specified by the manufacturer of the RTD. When doing measurements using quarter-bridge and half-bridge circuits, Rlead specifies the lead resistance of the wires connecting the sensor (i.e. strain gage) to the bridge.
	GF_7	Specifies the gage factor of a Strain Gage as specified by the manufacturer of the gage (i.e. relates resistance change to strain); V/V per Kg of force in the case of a Load Cell; and V per G in the case of an accelerometer.
	υ_$Poisson_8$	Specifies Poisson's ratio in axial strain gage measurements.
Directory -19		Contains fields that specify where and how information is saved to, and retrieved from, disk. For the most part, this is only used by programmers. For more information, please refer to Chapter *4, Programming*.
	Path $Name_1$	Specifies the path name for the directory/folder that is used when loading or saving files via the Save or Open buttons in the Record page.
	New $Name_2$	When the Save button is pressed in the Record page to save waveforms to disk, a new directory/folder is created for the new files. If the New Name field is set to Prompt $User_1$, the user is prompted for the new folder's name and location, otherwise, if New Name is set to Auto $Generate_2$, the folder is automatically created and named.
	Save $Fields_3$	When the Save button is pressed in the Record page to save waves to disk, the user has the option of saving the network Fields with the waveforms. Save Fields is set to On_1 if Fields are to be saved, and Off_2 if they are not.
	Load $Fields_4$	If Load $Fields_4$ is set to On_1, the network Fields are loaded from disk when one presses the Open button in the Record page = {On_1, or Off_2}.

	File Type$_5$	Specifies the file type for waveforms saved to disk via the Save button in the Record page = { Binary$_1$, Text (ASCII)$_2$, Text Merge (ASCII)$_3$, Binary Merge$_4$,}. Binary and Binary Merge are fast and compact, whereas Text is compatible with other programs such as word processors and spreadsheets. Text Merge saves an additional "Excel Waveform Data.txt" text file that contains all waves in one file. Binary Merge saves all channels in 1 file and spools to disk very fast; whereas Binary saves each channel in its own file and uses less RAM memory.
	Command$_6$	Writing New Directory$_1$ to the Command$_6$ field causes a new directory/folder to be created. Writing Show Dialog$_2$ to the Command$_6$ field causes the File Open dialog to appear, which enables one to modify the Path Name field. Command$_6$ is used primarily by programmers.

Table 8.1 Channel Setting Group Reference

Settings Group settingType	**Field** fieldNum	**Field Description** fieldValue
Display -7		Defines the vertical scale of Record page Displays, and allows one to enable a channel for digitizing.
	Digitize$_1$	If Digitize$_1$ is set to On$_1$, the channel is enabled for digitizing when the Start button is pressed in the Record page = {On$_1$, or Off$_2$}. When turned On within instruNet World, Input channels are Digitized and Plotted; whereas Output channels are only plotted (one must use the iW+ Control calculate field or Control script to drive output channels). When turned On while programming (e.g. C/BASIC), input channels are enabled for digitizing and their data is sent to data buffers; whereas output channels are placed into the digitize list and fed with data from their data buffers at the digitize sample rate.
	Display Max$_2$	Specifies the Engineering Units value that corresponds to the top line of a Record page or Probe dialog snapshot display.
	Display Min$_3$	Specifies the Engineering Units value that corresponds to the bottom line of a Record page or Probe dialog snapshot display.
	View$_4$	Show$_1$ or Hide$_2$ Display in Record page while digitizing.
Driver Ram -15		Defines a buffer, in RAM memory, that is maintained by the instruNet Driver. This is used by programmers to hold digitized data. For more information, please refer to Chapter *4, Programming*.
	Digitize$_1$	If Digitize$_1$ is set to On$_1$, the channel is enabled for digitizing, and the digitized data is sent to the Driver Ram buffer at runtime = {On$_1$, or Off$_2$}.
	Buffer Addr$_2$	This is the address, in RAM memory, of the Driver Ram Buffer; 0 if not used (off). The Buffer size corresponds to 1 Scan of data points, where each point is stored in a 32bit floating point number (4 bytes per point).

	Ptr Byte Size$_3$	This is the size of the Driver Ram Buffer in Bytes. This is often equal to 4 * PtsPerScan.
	Scan Num In$_4$	Scan Number of last data point pushed into Driver Ram Buffer = {1... numScans}.
	Pt Num In$_5$	Point Number of last data point pushed into Driver Ram Buffer = {1... PtsPerScan}
	Scan Num$_6$	Scan Number of last data point pulled out of Driver Ram Buffer by Access_Digitized_Data_In_Ram_Buffer() = {1...numScans}.
	Pt Num Out$_7$	Point Number of last data point pulled out of Driver Ram Buffer by Access_Digitized_Data_In_Ram_Buffer() = {1...PtsPerScan}.
File $_{-17}$		Specifies a file that contains digitized data. This is used primarily by programmers. For more information, please refer to *Chapter 4, Programming*.
	File$_1$	If File$_1$ is set to On$_1$ the channel is linked to the file = {On$_1$, or Off$_2$}.
	Digitize$_2$	If Digitize$_2$ is set to On$_1$ the channel is enabled for digitizing, and the digitized data is sent to the Driver Ram buffer at runtime = {On$_1$, or Off$_2$}.
	File Name$_3$	Specifies the file name that is linked to the channel. This is often the same as the channel name (e.g. "Ch1 Vin+").
	Command$_4$	Writing to the Command$_4$ field causes the following to occur (use the Record window Open/Save commands to connect with datasets on disk): File > Ram buf$_2$ — Transfer Num Pts$_7$ (number of points) from the Scan Num$_5$ scan, starting at PointNum$_6$, from the File to the Driver Ram Buffer. If FileType is Binary Merge, this occurs for all channels stored in the file for this network. File > User buf$_4$ — Same as File > Ram buf$_2$, yet to User Ram Buffer. Ram buf > File$_3$ — Transfer the data in the Driver Ram Buffer to the file on disk. If FileType is Binary Merge, this occurs for all channels stored in the file for this network. User buf > File$_5$ — Same as Ram buf > File$_3$, yet from User Ram Buffer. Get File Info$_6$ — Get information about the File. Load {Scan Num$_5$ PointNum$_6$} with the last point in the file, and load Num Pts$_7$ with the number of points per scan .
	Scan Num$_5$	Refers to a scan number = {1...numScans}.
	PointNum$_6$	
	Num Pts$_7$	Refers to the number of points in the last Scan = {1...PtsPerScan}.
General $_{-6}$		Specifies general information about a channel, such as its name and value.

	Value I/O$_1$	Specifies realtime value of the channel, in engineering Units. If the channel is a voltage input, this reflects the realtime voltage at the input terminals. If the channel is a voltage output, this is the voltage at the output terminals.
	Units Label$_2$	Specifies vertical label that is shown in the Record page and Probe dialog displays (e.g. "Volts", "C").
	User Name$_3$	Specifies the name of the channel that is shown in the displays (e.g. "Pressure1", "Temp2").
	% Sample Rate$_4$	Specifies the channel's sample rate as a percentage of the master sample rate that is defined in the Timing dialog (press Timing in Setup dialog) = {.001...100}. For example, if the master sample rate is 1Ks/sec, and Ch1's % Sample Rate$_4$ is set to 25, then Ch1 will digitize at 250s/sec.
Hardware $_{-3}$ *with Voltage Input Channels*		Specifies parameters that control a Voltage Input channel. For more information, please refer to *Chapter 2, Working with Sensors*, and *Chapter 3, Connecting To Sensors*.
	Sensor$_1$	Specifies type of sensor attached to Voltage input terminals = { Voltage$_1$, Current$_2$, Resistance$_3$, Strain Gage$_4$, RTD$_5$, Types J$_6$, K$_7$, T$_8$, E$_9$, R$_{10}$, S$_{11}$, B$_{12}$, N$_{13}$ Thermocouple, Thermistor$_{14,}$ Load Cell$_{16}$, Potentiometer$_{17}$, Accelerometer$_{18}$ }.
	Wiring$_2$	Specifies wiring at Voltage input terminals = { Vin+ - Vin-$_1$, Vin - Gnd$_2$, Shunt Resistor$_3$, Voltage Divider$_4$, Bridge$_5$, Quarter Bridge$_6$, Half Bridge Bend$_7$, Half Bridge Axial$_8$, Full Bridge Bend$_9$, Full Bridge Axial I$_{10}$, Full Bridge Axial II$_{11}$ }.

Table 8.1 Channel Setting Group Reference

Settings Group settingType	**Field** fieldNum	**Field Description** fieldValue
Hardware *Continued*	Low Pass$_3$	Specifies lowpass analog filter cutoff frequency in Hz (e.g. 0Hz = off, 40Hz, 4000Hz).
	Integrate$_4$	Specifies amount of integration (i.e. smoothing) in units of seconds. Signal are sampled and averaged for the duration specified in this field. For details, please see Application Notes #58 and #112.
	Range$_5$	Specifies maximum input range for the measured voltage, in units of Volts. For the most accurate readings, specify the smallest range possible without causing the measured voltage to hit the bound. For example if your maximum expected input range is +-50 mV select a range of +-.3V (i.e. set the Range$_5$ field to .3).
Hardware $_{-5}$ *with Digital I/O*		Specifies parameters that control a Digital I/O channel (e.g. Ch25 Dio on the Model 100 Device). For more information, please refer to *Chapter 2, Working with Digital I/O Channels.*

	Digital Output$_1$	Specifies the logic level of bits that have been set up as digital outputs in the Direction$_2$ field. Please see *Chapter 2, Working with Digital I/O Channels* for details on how this is done.
	Direction$_2$	Sets the direction of the various digital bits as an input (0) or an output (1).
Highpass Filter $_{-9}$		Defines a digital highpass filter. For more information, please refer to *Chapter 2, Working with Digital Filters*.
	Filter$_1$	Selects filter model = {Off$_1$, Elliptic$_2$, Chebyshev P$_3$, Chebyshev S$_4$, or Butterworth$_5$ }
	PassB Ripple$_2$	Specifies maximum allowable passband ripple, in dB.
	StopB Attn$_3$	Specifies the minimum stop band attenuation, in dB.
	Filter Order$_4$	Displays filter order. Automatically determined by the filter design, which is based on the specified criteria.
	PassB F1 Hz$_5$	Frequencies above PassB F1 Hz$_5$ are passed.
	StopB F1 Hz$_6$	Frequencies below StopB F1 Hz$_6$ are attenuated.
	Pass B F2 Hz$_7$	Not used.
	Stop B F2 Hz$_8$	Not used.
Lowpass Filter $_{-8}$		Defines a digital lowpass filter. For more information, please refer to *Chapter 2, Working with Digital Filters*.
	Filter$_1$	Selects filter model = {Off$_1$, Elliptic$_2$, Chebyshev P$_3$, Chebyshev S$_4$, or Butterworth$_5$ }
	PassB Ripple$_2$	Specifies maximum allowable passband ripple, in dB.
	StopB Attn$_3$	Specifies the minimum stop band attenuation, in dB.
	Filter Order$_4$	Displays filter order. Automatically determined by the filter design, which is based on the specified criteria.
	PassB F1 Hz$_5$	Frequencies below PassB F1 Hz$_5$ are passed.
	StopB F1 Hz$_6$	Frequencies above StopB F1 Hz$_6$ are attenuated.
	Pass B F2 Hz$_7$	Not used.
	Stop B F2 Hz$_8$	Not used.

Table 8.1 Channel Setting Group Reference

Settings Group settingType	Field fieldNum	Field Description fieldValue
Display Options -18		Specifies how waveforms are displayed and stored while digitizing. For more information, please refer to *Chapter 2, Digitizing Analog Signals into the Computer* and *Chapter 5, Display Options*.
	Horiz Scale1	Specifies the horizontal scale of the displays in the Record page and Probe dialog (i.e. Seconds per horizontal division). Use 0.0 to enable automatic scale selection, which is based on the sample rate and Scan size.
	Horiz Pos2	Specifies the horizontal position of the displays in the Record page and Probe dialog (i.e. seconds associated with the display left edge). This, in effect, is linked to the horizontal scrollbar.
	Plot3	Specifies whether to plot one dot per data point, or to connect data points with lines = { Lines1, or Dots2 }. Lines1 requires more processor time.
	Grid4	Turns display grid on or off = { On1, or Off2 }.
	Max Pts/Pix5	Sets the maximum number of points plotted in each vertical column of pixels.
	Digitize Into6	Specifies where digitized data is saved = { Off1, ToRamBuffer2, ToFile3, or UserControl4 }.
	Overflow Alrt7	Specifies whether or not the instruNet driver shows an alert when the Digitizer buffer overflows = { On1, or Off2 }.
Timer -14		Specifies the function of the Digital Timer I/O Channels on the Model 200 instruNet Controllers (not iNet-230). For more information, please refer to *Chapter 2, Working with Controller Digital Timer I/O Channels.*
	Function1	Sets the channel function as one of = { Digital In1, Digital Out2, Clock Output3, Period Measurement4 }. See *Chapter 2, Working with Controller Digital Timer I/O Channels* for details.
	Clk Period2	Sets the cycle time, in seconds, when doing Clock Output.
	Clk Out Hi3	Sets the high time, in seconds, when doing Clock Output.
	Measure4	When doing Period Measurement, this specifies if the cycle time or the high time is measured = { Cycle Time1, or High Time2 }.
	Meas. Resol.5	When doing Period Measurement, this specifies the measurement accuracy to one of = { .25µs1, or 4ms2 }.
	Meas. Cycles6	When doing Period Measurement, this specifies the number of cycles or high times that must elapse during the measured duration = {0...255}.

Table 8.1 Channel Setting Group Reference

Settings Group settingType	**Field** fieldNum	**Field Description** fieldValue
Timing -12		Specifies parameters for the DSP based digitizing via the Model 200 and 230 instruNet Controllers. For more information, please refer to *Ch2, Digitizing Analog Signals into the Computer*, & *Ch 5, Timing Options*.
	Digitize1	Turns digitizing on or off for all channels = { On1, or Off2 }. This field is automatically set to On1 when the Start button is pressed and Off2 when the Stop button is pressed.
	Pts Per Scan2	Specifies the number of points digitized for each Scan, at the master Sample Rate5.
	No. of Scans3	Specifies the number of Scans that are digitized when the Start button is pressed.
	Scan Mode4	Specifies whether consecutive Scans are continuous with respect to each other = { Strip Chart1, Oscilloscope2, Oscillo Queued3 }. Please refer to *Chapter 5, Strip Chart and Oscilloscope Scan Modes*, for details.
	Sample Rate5	Specifies the master sample rate, in units of samples-per-second-per-channel. If the specified rate is too fast, instruNet will adjust to the fastest possible rate. All channels run at this rate, unless their % Sample Rate4 field is requesting a slower rate.
	Min sec/tsfr6	Specifies the minimum acceptable time to transfer a 16bit value on the instruNet network, in units of seconds.
	Network BPS7	Specifies the instruNet network data transfer rate in bits per second = {100,000 ...4,000,000}. This is automatically set to the fastest possible rate when instruNet is reset. Long network cables and/or many network Devices sometimes require a slower rate.
	Switching8	Specifies analog channel switching to run Fast2 or Accurate1. If Fast is used, the system switches from one channel to another as fast as possible; otherwise, with Accurate, the switching is a little slower, yet provides the amplifiers more time to settle, and is therefore a little more accurate.

Table 8.1 Channel Setting Group Reference

Settings Group settingType	**Field** fieldNum	**Field Description** fieldValue
Trigger -13		Specifies the Trigger condition that must be met before digitizing (after the Start button is pressed). For more information, please refer to *Chapter 2, Digitizing Analog Signals into The Computer*, & *Ch 5, Trigger Options*.
	$Trigger_1$	Sets the trigger mode to one of: { Off_1, $Auto_2$, or $Norm_3$ }. Off_1 specifies no trigger, $Norm_3$ mandates that the digitizing cannot begin until the trigger condition is met, and $Auto_2$ waits for the trigger condition yet digitizes anyway if the condition is not met within several seconds.
	Threshold EU_2	Specifies the trigger threshold in engineering units (EU).
	$Slope_3$	Specifies the direction the waveform must cross the threshold in order to trigger = {$Rising_1$, or $Falling_2$}.
	$Expansion_4$	
	Trig. $Net\#_5$	Specifies the Network number of the trigger channel.
	Trig. $Dev\#_6$	Specifies the Device number of the trigger channel.
	Trig. $Mod\#_7$	Specifies the Module number of the trigger channel.
	Trig. $Chan\#_8$	Specifies the Channel number of the trigger channel.
User Ram -16		Defines a buffer, in RAM memory, that is maintained by the programming end user. This buffer is used to hold digitized data. For more information, please refer to Chapter *4, Programming*.
	$Digitize_1$	If $Digitize_1$ is set to On_1, the channel is enabled for digitizing, and the digitized data is sent to the User Ram buffer at runtime = {On_1, or Off_2}.
	User $Addr_2$	This is the address, in RAM memory, of the User Ram Buffer; 0 if not used (off). The Buffer size must be enough to hold 1 Scan of data points, where each point is stored in a 32bit floating point number (4bytes/point).
	Buff Byte $Size_3$	This is the size of the User Ram Buffer in Bytes. This must be greater than or equal to 4 * PtsPerScan.
	Scan Num In_4	Scan Number of last data point pushed into User Ram Buffer = {1... numScans}.
	Pt Num In_5	Point Number of last data point pushed into User Ram Buffer = {1... PtsPerScan}
	Scan Num_6	Scan Number of last data point pulled out of User Ram Buffer by Access_Digitized_Data_In_Ram_Buffer() = {1...numScans}.
	Pt Num Out_7	Point Number of last data point pulled out of User Ram Buffer by Access_Digitized_Data_In_Ram_Buffer() = {1...PtsPerScan}.

Table 8.1 Channel Setting Group Reference

Appendix I, Troubleshooting

If your instruNet system is not operating properly, use the information in this chapter to isolate the problem. If the problem appears serious enough to warrant technical support, please contact your instruNet supplier.

Identifying Symptoms and Possible Causes

Use the Troubleshooting information in the following table to try to isolate the problem. This table lists general symptoms and possible solutions for problems with instruNet hardware and software.

Symptom	Possible Cause	Possible Solution
Computer crashes when instruNet World is run.	The Controller board is not seated properly, or it is damaged.	Check the installation of the Controller hardware and instruNet software per directions in *Chapter 1*. Try rebooting the computer. Also, refer to the next section of this chapter entitled "... your controller is not seen by the instruNet World Software" paying careful attention to the discussion about making sure the card is properly seated in its connector. Also, please refer to Application Note #121.
Digitized waveforms appear to be invalid.	An open connection may exist on a channel.	Check the wiring to the input terminals.
	The sensor is not wired correctly.	Refer to Chapter 5 for instructions on wiring sensors.
	The Field settings are not correct.	Refer to *Chapter 5 Sensors Reference*, *Chapter 7 Channel Reference* & *Chapter 8 Settings Reference*. It is recommended that Chapter 2 instruNet World Tutorial be completed before attempting to wire sensors.
	A Channel is configured as a single-ended input while the transducer is a differential type, or vice-versus.	Check the wiring to instruNet and the Wiring settings. Check the transducer type.
Computer will not boot	The Controller board is not seated properly, or is damaged.	Check the installation of the Controller hardware and instruNet software per the directions in *Chapter 1*.

instruNet World opens, yet does not see Controller (i.e. only 3 rows of data are displayed in Network).	The Controller board is not seated properly, or is damaged.	Please refer to the next section of this chapter for information on this scenario.
	Driver is not compatible with your computer.	Install latest version of driver, available for free at www.instrunet.com. Beware that if the driver files are also in the same directory as the application (e.g. in the same directory as "instruNet World"), the application may use those driver files instead of the one's in your operating system directory; therefore, make sure that both are current (or delete the driver files in the same directory as the application).
instruNet World opens ok, yet does not see an instruNet 100 Device	instruNet 100 device is not connected to the instruNet controller by an instruNet cable, or instruNet cable is not well seated.	Use a flash light to make sure connectors at the controller end and device are well seated. Some computers have a small corridor through which the instruNet cable must pass through to get to the instruNet controller DB-25 connector, and in some cases, this hole is not large enough, and subsequently requires reducing the size of the instruNet cable molding with a knife.
	instruNet Terminator not attached to the end of the network.	Check Terminator.
	In i100 and i200 units manufactured before 9/1/97: Blown power fuse (e.g. 5V, -12V or +12V) in the Device, or in the Controller (which could have been caused by plugging in devices while the power is on).	To test if your fuses are ok, measure the voltages at the 5V, -12V and +12V screw terminals on the instruNet 100. For example, to measure the +5V voltage, place one voltmeter lead on the "+5V" screw terminal and the other lead on the "GND" screw terminal. If the voltage is off by more than +/1V, a fuse is probably blown in either the instruNet 100 or the Controller card. Please contact your instruNet supplier if this is the case.
	Bad instruNet Cable.	Try another cable.
	Broken Controller or Device.	Consult your instruNet supplier.

Table AI-1 Troubleshooting

If the instruNet PCI Controller Board is not seen by the instruNet World software...

After installing the instruNet software per Chapter 1, one can easily verify its installation by running the instruNet World software. If on Windows, select "instruNet World" in the "instruNet" group within the START menu to run instruNet World. When this software first opens, a list of instruNet resources are listed in a spreadsheet like format. If only 3 rows of information are displayed, then the software only sees the driver; if 15 rows are displayed then the Controller (e.g. PCI) is seen as well; and if 40 rows are displayed, then an instruNet 100 network device is also seen. Also, to verify that these three items are installed, one can click on the TEST tab at the bottom of the window, and press the SEARCH button to display a list of installed resources (e.g. instruNet Driver, i200 Controller, i100 Device).

If the Controller (e.g. pci or pcmcia card) is not seen on a Computer, then please proceed with the following steps:

1. Use a flash light to make sure the board is well seated in its connector. With some computers, it is difficult to insert the tab at the bottom of the I/O fence into its receptacle. Sometimes, the I/O fence at the back of the computer is not registered with respect to the motherboard, and tightening the I/O fence screw causes the card to enter its connector at an angle. If this happens, leave the screw loose and make sure the card is properly aligned in its connector so that the pcb finders correctly align with their mating receptical pins.

2. Turn the computer power off for 10 seconds, turn it back on to boot the computer, and then run "instruNet World" to see if the card is found. If it is not found and you are running on a Windows computer, please power the computer off, and then on again, since re configuring the cards internally sometimes requires two power off/on cycles (sometimes due to plug-and-play arbitration going on inside the computer).

3. If running under Windows, make sure "iNet32.DLL" Version ≥ 2.0 is installed in the System directory, within the Windows directory (i.e. "Windows\System\iNet32.dll" on a Win 95/98/Me computer, or "Windows\System32\iNet32.dll" on a Win Nt/2k/Xp computer).

If on a PPC Macintosh, make sure "instruNet Driver (ppc)" Version ≥ 1.32.12 is installed in the System's Extensions folder.

If running under Windows Nt/2k/Xp, make sure you log on as the Administrator before running the Setup.exe instruNet installer program, and then restart your computer after running the Setup.exe.

The latest software is available for download, free of charge, at www.instrunet.com.

Beware that if the driver files are also in the same directory as the application (e.g. in the same directory as "instruNet World"), the application may use those driver files instead of the one's in your operating system directory; therefore, make sure that both are current (or delete the driver files in the same directory as the application).

4. If on a Windows computer: If you are running a video accelerator, try turning it off, and see if this helps.

5. If your computer has several ISA or PCI cards that are not necessary for computer operation (e.g. sound card, fax/modem card, scsi card, etc) then it is recommended that you unplug them one at a time (while power is off), boot the computer (for each case) and run the instruNet software.

6. It is possible the controller is broken. To detect this, try a different computer or a different controller.

7. With older computers, it is sometimes possible that an old ISA card driver is conflicting with a new PCI card. A conflict resolution program such as "First Aid", or reinstalling the OS software, might remedy the problem.

8. If running under Windows with an instruNet PCI card, and instruNet Version ≤ 1.22 (1998) was previously installed on this computer, and you are now running >1.22, then it might be necessary to remove some debris from the older software. To do this, run The "System" Control Panel, select "Device Manager", select "View Devices by connection", expand "PCI bus", if you see "? PCI Card" select it & press the Remove button, and exit "System" Control Panel. Then, reboot your computer, and when it asks for a PCI driver, Navigate via the Browse button to "Program Files \ instruNet \ Internal \ PCI Win95 Driver \ inet95.inf". This inf file is installed on your computer when you run the >1.22 instruNet Setupex.exe file. Then run "instruNet World" software and look for >14 rows in the Network page, indicating that it found the pci controller (if only 3 are shown, it means that it found the Driver software, yet not a controller card).

9. Email a diagnostic report to your supplier, as noted in the following discussion.

10. Please refer to the Troubleshooting Application Notes at www.instruNet.com.

Emailing a Diagnostic Report to your Supplier

To Email an instruNet Diagnostic Report to your Supplier: Run instruNet World software, press the TEST tab, press the REPORT button to print the diagnostic report to the window, copy the text to the clipboard (Windows Control C), paste the text into an email window (Windows Control V) and then email to your instruNet supplier for comment. A Diagnostic Report can also be generated by pressing the "SAVE DEBUG REPORT TO DISK" button within an error alert. If there is a conflict with other pci/pcmcia cards, then a emailing a registry report (another type of report) via Application Note #184 might also be helpful.

Appendix II, Error Codes

The table below lists error codes returned by the instruNet driver along with possible causes and solutions to the problem. These are instruNet error codes, and are very different from your operating system error codes.

Error Code	Error Label	Possible Cause	Possible Solution
0	iNetErr_-None	no error	The operation was successfully completed.
1	iNetErr_- General	potentially anything	Try doing things differently and hope it goes away.
2	iNetErr_-Controller-NotInitialized	instruNet has not been initialized	Check cables. Try pressing the Reset button. See *Chapter 1*.
3	iNetErr_-InitFailed	instruNet initialization failed	Check network cables and termination. Check software installation. Try pressing the Reset button.
4	iNetErr_-DeviceNum-OutOfRange	Device number (deviceNum) is out of range	Make sure all hardware Devices are connected and powered on. Press Search button in Test page for list of registered Devices. Make sure the specified 'deviceNum' is correct. See *Chapter 1*.
5	iNetErr_-ChannelNum-OutOfRange	Channel number is out of range	Make sure the specified 'chanNum' channel number parameter is correct. See *Chapter 7*.
6	iNetErr_-FieldNum-OutOfRange	Field number is out of range	Make sure the specified 'fieldNum' field number parameter is correct. See *Chapter 8*.
7	iNetErr_-ControllerNot-Found	instruNet Controller not found	Make sure the specified 'netNum' network number parameter is correct. Press Search button in Test page for list of registered Networks. See *Chapter 1, Hardware Installation*.
8	iNetErr_-FieldDoesNotExist	specified {net, device, module, chan, setting, field} does not exist	Make sure the specified 'netNum', 'deviceNum', 'moduleNum', 'chanNum', 'settingNum', 'fieldNum' parameters are correct. See *Chapter 8*.
9	iNetErr_-BadfieldNat-iveDataType	bad data type	Make sure the specified 'netNum', 'deviceNum', 'moduleNum', 'chanNum', 'settingNum', 'fieldNum' parameters are correct.
10	iNetErr_-BadField-ReadType	bad read type	Make sure the specified 'netNum', 'deviceNum', 'moduleNum', 'chanNum', 'settingNum', 'fieldNum' parameters are correct.
11	iNetErr_-Timeout-AtReadBegin	time out at controller, crashed controller	Press Reset Button.
12	iNetErr_-TimeoutAt-WaitFor-ReadDone	time out at controller (crashed controller?)	Press Reset Button.
13	iNetErr_-Controller-IsInWeeds	crashed controller	Press Reset Button.
14	iNetErr_ill-egalDataType	bad data type	Make sure the specified 'netNum', 'deviceNum', 'moduleNum', 'chanNum', 'settingNum', 'fieldNum' parameters are correct. See *Chapter 8*.
15	iNetErr_-FailedCopy-DataTest	failed the CopyWaveData() test or bad instruNet Driver file	Try reinstalling installing Driver file and make sure you install correct version of Driver.
16	iNetErr_-Compressor-HitError	compressor hit error	Press Reset Button.
17	iNetErr_-FailedRam-Test	failed board ram test	Power Computer off, then on. Press Big Test button in Test page. Disconnect network cable from Controller. Controller might need service.
18	iNetErr_-RanOutOf-Memory	instruNet Driver ran out of memory	Try giving calling application program more memory. Try reducing the number of points per scan. Try a computer with more RAM.

19	iNetErr_-AlertFailed	the routine that shows an alert failed	Try reinstalling installing Driver file and make sure you install correct version of Driver.
20	iNetErr_-CtrlrRom-NotBooting	instruNet Controller's ROM does not seem to boot up (poss problem: controller, bus, rom)	Power Computer off, then on. Press Big Test button in Test page. Disconnect network cable from Controller to see if that fixes it. Controller might need service.
21	iNetErr_-CtrlrRam-NotBooting	instruNet Controller's driver in RAM does not seem to boot up (poss problem: controller, bus, ram, rom, download from uC, bad driver downloaded)	Power Computer off, then on. Press Big Test button in Test page. Disconnecting network cable from Controller might help. Controller might need service. Latest version of instruNet Driver might help.
22	iNetErr_-DriverDown-loadFailed	the download of the uController driver into uC ram failed (driver may be bad, or hardware is bad) (the keys did not match).	Power Computer off, then on. Press Big Test button in Test page. Disconnecting network cable from Controller might help. Controller might need service. Latest version of instruNet Driver might help.
23	iNetErr_-CtrlrRW-TestFailed	failed during controller rw test in Test_DualPort_Ram()	Power Computer off, then on. Press Big Test button in Test page. Disconnecting network cable from Controller might help. Controller might need service. Latest version of Driver might help.
24	iNetErr_-Interface-BlockTest-Failed	Interface block between uController and host computer is invalid	Power Computer off, then on. Press Big Test button in Test page. Disconnecting network cable from Controller might help. Controller might need service. Latest vers. of instruNet Driver might help.
25	iNetErr_-IncCounter-TestFailed	Controller failed CounterInc test	Power Computer off, then on. Press Big Test button in Test page. Disconnecting network cable from Controller might help. Controller might need service. Latest vers. of instruNet Driver might help.
26	iNetErr_-EchoCmd-ToStatus-TestFailed	Controller failed EchoCmdToStatus test	Power computer off, then on. Press Big Test button in Test page. Disconnecting network cable from Controller might help. Controller might need service. Latest vers. of instruNet Driver might help.
27	iNetErr_-Controller-BootTest-Failed	Controller failed Test_A_Booted_Controller test	Power Computer off, then on. Press Big Test button in Test page. Disconnecting network cable from Controller might help. Controller might need service. Latest vers. of instruNet Driver might help.
28	iNetErr_-Controller-FailedToBoot	Controller failed to Boot.	Power Computer off, then on. Press Big Test button in Test page. Disconnecting network cable from Controller might help. Controller might need service. Latest vers. of instruNet Driver might help.
29	iNetErr_-Controller-CmdFailed	Controller failed to execute command	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
30	iNetErr_GUI	error related to graphical user interface	Try reinstalling installing Driver file and make sure you install correct version of Driver.
31	iNetErr_-QSPI_Busy	QSPI is busy running	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
32	iNetErr_-QSPI_Halted	QSPI hit HALT error	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
33	iNetErr_-QSPI_Arg-OutOfRange	QSPI argument out of range	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
34	iNetErr_-QSPI_-TimeOutErr	QSPI hit time out error	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
35	iNetErr_-FlakyNet-work	QSPI is acting flaky	Check instruNet network cables. Make sure a network terminator is properly installed. See *Chapter 1, Hardware Installation*.
36	iNetErr_-CouldNot-LocateDriverFile	could not find instruNet Driver file in system folder	Make sure you installed the correct instruNet Driver file at the correct location on your computer. See *Chapter 1, Software Installation*.
37	iNetErr_-netNumOut-OfRange	netNum is out of range	Make sure the specified 'netNum' network number parameter is correct. Press the Search button in the Test page for a list of registered Networks. See *Chapter 1, Hardware Installation*.
38	iNetErr_-SettingGro-upNumOut-OfRange	setting GroupNum is out of range	Make sure the specified 'settingGroupNum ' setting group number parameter is correct. Press the Search button in the Test page for a list of registered Networks and Devices. See *Chapter* 7.
39	iNetErr_-UnitType-OutOfRange	deviceType is out of range	Check instruNet network cables. Make sure a network terminator is properly installed. See *Chapter 1, Hardware Installation*.

40	iNetErr_-DriverDid-NotSetErr-Code	Driver did not get a chance to set the error code; therefore Driver, or interface to Driver, is in trouble	Check the interface to Driver. Make sure you installed the correct instruNet Driver file at the correct location on your computer. See *Chapter 1, Software Installation*.
41	iNetErr_-SettingGroup-TypeOutOf-Range	settingGroupType is out of range	Make sure the specified 'settingGroupType' setting group type parameter is correct. Press the Search button in the Test page for a list of registered Networks and Devices. See *Chapter 8.*
42	iNetErr_-ModuleNum-OutOfRange	Module number is out of range	Make sure the specified 'moduleNum' module number parameter is correct (it is usually 1). Press the Search button in the Test page for a list of registered Networks and Devices.
43	iNetErr_-Intention-NumOutOf-Range	Intention number is out of range	Make sure the specified 'intention' intention number parameter is correct. See list of valid 'ion_intention' values in interface .h file.
44	iNetErr_-ReadOnly-Field	Cannot write to this field, read only	Make sure the specified 'netNum', 'deviceNum', 'moduleNum', 'chanNum', 'settingNum', 'fieldNum' parameters are correct. See *Chapter 8*.
45	iNetErr_-WriteOnly-Field	Cannot read from this field, write only	Make sure the specified 'netNum', 'deviceNum', 'moduleNum', 'chanNum', 'settingNum', 'fieldNum' parameters are correct. See *Chapter 8*.
46	iNetErr_-FieldValue-OutOfRange	Tried to set a field with a value that is too high or low	Make sure the specified 'netNum', 'deviceNum', 'moduleNum', 'chanNum', 'settingNum', 'fieldNum' parameters are correct. See *Chapter 8*.
47	iNetErr_-ArgTypeOut-OfRange	ArgType parameter is out of range	Make sure specified 'argType' parameter is correct. See list of valid 'instruNetDataType' values in interface .h file.
48	iNetErr_-BadKeyIn-Field-Hierarchy	A BAD key was found in the field hierarchy data -- internal data might be corrupted	Check network cables. Make sure a network terminator is properly installed. Latest version of instruNet Driver might help. Try pressing Reset button. Resetting computer might help.
49	iNetErr_-Max_LT_-MinInField-Hierarchy	A maximum value is less than a minimum value in the field hierarchy -- internal data might be corrupted	Latest version of instruNet Driver might help. Try pressing Reset button. Resetting computer might help.
50	iNetErr_-Hierarchy-FieldData-In-Trouble	Hierarchical field data is in trouble -- internal data might be corrupted	Latest version of instruNet Driver might help. Try pressing Reset button. Resetting computer might help.
51	iNetErr_-Channel-NameInvalid	The channel name is in trouble -- internal data might be corrupted	Latest version of instruNet Driver might help. Try pressing Reset button. Resetting computer might help.
52	iNetErr_-tempUnits_-outOfRange	temperature scale {C,K,F} out of range	Check temperature measurement hardware and software.
53	iNetErr_-sensorType_-outOfRange	sensor type out of range	Check sensor hardware and software.
54	iNetErr_-CircBufErr	circular digitizing data buffer error	Try a slower sample rate, or ask instruNet to do less while digitizing.
55	iNetErr_-DataBuffer-Overflow	circular digitizing data buffer overwrote data before it was read	Try a slower sample rate, or ask instruNet to do less while digitizing.
56	iNetErr_-PulledToo-MuchOn-LastPull	circular digitizing data buffer error where pulled too much on last pull	Try a slower sample rate, or ask instruNet to do less while digitizing.
57	ERequired_-fbx_DCIIR	At least one cutoff frequency (passband or stopband) is needed for each transition band of bandpass and bandstop digital filters	Check digital filter Frequency Cutoff fields.
58	EFreqToo-Large_fx_-DCIIR	Cutoff frequency must be less than half the sampling rate	Check digital filter Frequency Cutoff fields and make sure they are lower than half the sample rate.
59	EFreqsNot-Ascending_-DCIIR	Cutoff frequency negative or frequencies not in ascending order	Check digital filter Frequency Cutoff fields.
60	ERequired_-fx_DCIIR	Missing one or more cutoff frequencies	Check digital filter Frequency Cutoff fields.
61	ERequired_-adelx_DCIIR	Missing passband ripple and or stopband attenuation	Check digital filter Ripple and Attenuation fields.
62	EInvalidArg_-DCIIR	Invalid argument	Check digital filter fields.
63	EOrderToo-High_DCIIR	Necessary or specified filter order is too high -- maximum order is %d	Reduce the digital filter Attenuation field value, or increase the Ripple field value.
64	EEven_-ndeg_DCIIR	Filter order must be even for bandpass and bandstop filters -- order being increased by 1	Reduce the digital filter Attenuation field value, or increase the Ripple field value.
65	EOrderToo-Low_DCIIR	Specified filter order is too low -- order being automatically increased	Increase the digital filter Attenuation field value, or decrease the Ripple field value.

66	EActualOrder-_DCIIR	Required filter order = %d (%s biquadratic section%s)	Check digital filter fields.
67	iNetErr_-InterfaceCom-piledBadly	a variable type in interface file (e.g. INET_INT.C) is bad	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
68	iNetErr_-BadInterface-Key	the 'key' field passed to driver is bad	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
69	iNetErr_-BadAddrPa-ssedToDriver	bad address passed to driver	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
70	iNetErr_-BadStatic-VarInDriver	bad static variable in driver	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
71	iNetErr_-BadInteger-MathInDriver	bad integer math in driver	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
72	iNetErr_-BadChannel-Type	bad channel Type	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
73	iNetErr_-CppCompiler-DidBad	Cpp compiler failed	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
74	iNetErr_-MemMngr_-Failed	Memory Manager failed	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
75	iNetErr_-Toolbox_-Failed	Toolbox failed	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
76	iNetErr_-CrtRect_-Failed	CrtRect failed	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
77	iNetErr_-DlogCode_-Failed	Dialog Code failed	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
78	iNetErr_-DrvrNeeds-Fpu_Failed	Driver file needs FPU	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer. Check Interface to, and installation of, instruNet Driver.
79	iNetErr_-iirCode_-Failed	iir code failed	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer. Check Interface to, and installation of, instruNet Driver.
80	iNetErr_-sprintf_Failed	sprintf failed	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer. Check Interface to, and installation of, instruNet Driver.
81	iNetErr_-DigitizeInit	Digitize initialization failed	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
82	iNetErr_-SPEoff	instruNet network error	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
83	iNetErr_-Halton	instruNet network error	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
84	iNetErr_-QPTQP	instruNet network error	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
85	iNetErr_-qBusy	instruNet network is busy before digitize	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
86	iNetErr_-abort	we aborted early	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
87	iNetErr_-cBusy	controller is busy doing something	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
88	iNetErr_-cNotFin	controller did not finish the command	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
89	iNetErr_-CodeGen	code generation segment error	Latest version of instruNet Driver might help.
90	iNetErr_-CPTQPbad	instruNet network error	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.

91	iNetErr_-Compiler	compiler error	Latest version of instruNet Driver might help.
92	iNetErr_-bOverflow	driver or user ram buffer overflow	Try reducing the sample rate, making displays much smaller, plotting less PointsPerPixel.
93	iNetErr_-nonCompOS	non-compatible operating system	Latest version of instruNet Driver might help.
94	iNetErr_bad-ChPtsPerScan	bad channel Points-Per-Scan value	Try a different Points-Per-Scan (i.e. press Timing button) or '% sample rate'.
95	iNetErr_-DestBuff -NotFound	destination buffer not found	Latest version of instruNet Driver might help. Try pressing Reset button or Resetting computer Check Interface to, and installation of, instruNet Driver.
96 to 200	Please search for "iNetErrorCode" in file "… program files \ instrunet \ Programming Interfaces \ Visual C .Net \ Source_Bak \ iNet_int.h" for details.		
10000 to 10255	Controller Status Register Error	this is an error code from the instruNet controller Status register	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.
-1 to -32000	OS Error	this is an error code from the Operating System	Press Reset Button. Try doing things differently and hope it goes away. Latest version of instruNet Driver might help.

Table AII - Error Codes

Appendix III, Working With Spreadsheets

instruNet supports the acquisition of data into instruNet World with Export to a spreadsheet, post-acquisition; and also supports the digitizing of data directly into an Excel spreadsheet, in realtime, via the "Direct To Excel" program described in Chapter 2.

MERGED.TXT

	A	B	C	D
1	instruNet waves in merged TEXT form #2			
2				
3	Notes:			
4	Date & Time:	2/18/97	22:03:00	
5	ChanName:	Ch1 Vin+	Ch4 Vin+	Ch7 Vin+
6	UserName:	Ch1 Vin+	Ch4 Vin+	Ch7 Vin+
7	vUnits:	Volts	Volts	Volts
8	hUnits:	Secs	Secs	Secs
9	PtsPerScan:	10	10	10
10	eforeLastScan:	0	0	0
11	PtsInLastScan:	10	10	10
12	1stPtTime:	0.00E+00	0.00E+00	0.00E+00
13	SamplePeriod:	1.00E-03	1.00E-03	1.00E-03
14	netNum:	1	1	1
15	deviceNum:	1	1	1
16	moduleNum:	1	1	1
17	chanNum:	1	4	7
18	BytesPerRow:	-1	-1	-1
19	FirstDataByte:	-1	-1	-1
20	key:	21940	21940	21940
21	1904secs:	2.939E+09	2.939E+09	2.939E+09
22	DataType:	5	5	5
23	Internal1:	5.00E+00	5.00E+00	5.00E+00
24	External1:	5.00E+00	5.00E+00	5.00E+00
25	Internal2:	-5.00E+00	-5.00E+00	-5.00E+00
26	External2:	-5.00E+00	-5.00E+00	-5.00E+00
27	Secs	Ch1 Vin+	Ch4 Vin+	Ch7 Vin+
28	0.00E+00	2.35E-03	-1.23E-03	1.83E-03
29	1.00E-03	2.35E-03	-2.45E-03	1.83E-03
30	2.00E-03	2.35E-03	-1.23E-03	6.12E-04
31	3.00E-03	3.57E-03	-1.23E-03	1.83E-03
32	4.00E-03	2.35E-03	-1.05E-05	1.83E-03
33	5.00E-03	2.35E-03	1.21E-03	6.12E-04

Exporting a file to a spreadsheet, post-acquisition, is done by first acquiring data into instruNet World software, saving it to disk in the form of a text file, and then opening that text file with a spreadsheet, as illustrated to the left. To do this, one would:

1. Run instruNet World.
2. Select channels for digitizing in the Network Page.
3. Press the Record tab.
4. Press the Setup button to open the Setup dialog. Select Digitize Into: To Ram Buffer, select File Type: Text Merge, select the desired Sample Rate, select the desired Pts Per Scan, set the No of Scans to 1, set Scan Mode to Oscillo, and press OK to exit the dialog.
5. Press the Network tab, press the Save button to save these settings to disk, and then press the Record tab to return to the Record page.
6. Press the Start button to begin recording, and the Stop button to stop recording.
7. Press the Save button in the Record page (or, if running instrunet World+, one can select Save Waveforms To Excel in the Record menu) to save the data to disk in a text file named "Excel Waveform Data.txt". This file places each channel in its own column, and looks something like the illustration to the left when opened with a spreadsheet program.

Columns are separated with TAB characters, and rows are separated with CARRIAGE RETURN characters.

Appendix IV, Working With Application Software

DASY*Lab*

instruNet is compatible with DASYLab software Version ≥ 4.01.11 and Version >1.27 of the instruNet Driver. DASYLab with instruNet runs on a Windows ≥ 95 computer and is fully 32bit compatible.

DASYLab is the easy-to-use data acquisition software application. Its outstanding analysis and display features make it the ideal tool for many types of measurement and control applications. Designed as an open system, DASYLab contains drivers for more than 250 different data acquisition devices as well as software interfaces and extension toolkits. With its unique structure, DASYLab is able to acquire data up to 1 MHz into the PC's RAM, stream data to disk at up to 400 kHz and display data online at up to 150 kHz. For details, please select "Programs / instruNet / Application Software / DasyLab /" in the Windows START menu.

LabVIEW

instruNet Drivers for the Macintosh & Windows Version of LabVIEW $\geq$ 4 are available. The part numbers for these low cost drivers are listed below. Please consult their documentation for information on how to link instruNet to LabVIEW. In summary, they provide LabVIEW icons for many of the instruNet functions described in Ch4 Programming, and provide examples of their use. For details, please select "Programs / instruNet / Application Software / LabVIEW /" in the Windows START menu.

#iNet-380	instruNet Drivers for Mac & Windows LabVIEW
#iNet-380-10x	Ten more iNet-380's after buying 1st one
#iNet-381	instruNet LabVIEW Driver 1year Update

HP VEE

InstruNet is compatible with HP Vee Version $\geq$4.0 on Windows. For details, please select "Programs / instruNet / Application Software / HP Vee /" in the Windows START menu.

TestPoint

InstruNet is compatible with TestPoint Version $\geq$3.0 on Windows. For details, please select "Programs / instruNet / Application Software / TestPoint /" in the Windows START menu.