EndRun TECHNOLOGIES
"Smarter Timing Solutions"

Ninja *Precision Timing Module*

GPS-Synchronized

User Manual

USM3211-0000-000 Revision 3
August 2021

Ninja

Precision Timing Module User Manual

Preface

Thank you for purchasing the Ninja Precision Timing Module. Our goal in developing this product is to provide an extremely accurate time and frequency standard referenced to Coordinated Universal Time (UTC) that you can deploy quickly, easily and reliably. Your new Ninja is fabricated using the highest quality materials and manufacturing processes available today, and will give you years of troublefree service.

About EndRun Technologies

EndRun Technologies has been dedicated to the development and refinement of the technologies required to fulfill the demanding needs of the time and frequency community since 1998.

The instruments produced by EndRun Technologies have been selected as the time and frequency reference for such rigorous applications as enterprise computer synchronization, research institutions, aerospace, network quality-of-service monitoring, satellite earth stations, and calibration laboratories.

Trademark Acknowledgements

Linux and Windows are registered trademarks of the respective holders.

EndRun Contact Information

Address:	EndRun Technologies 2270 Northpoint Parkway Santa Rosa, CA 95407 U.S.A.
Phone:	(707)573-8633
Fax:	(707)573-8619
Sales:	1-877-749-3878 or (707)573-8633 sales@endruntechnologies.com
Support:	1-877-749-3878 or (707)573-8633 support@endruntechnologies.com

Part No. USM3211-0000-000 Revision 3
August 2021

About This Manual

This manual will guide you through simple installation and set up procedures.

Introduction – The Ninja, how it works, where to use it, its main features.
Basic Installation – How to connect, configure and test your Ninja with your network.
NTP Server and Client Set-Up – Two client sections; one for Unix-like platforms and one for Windows.
Network Protocols - Covers Security, SNMP, HTTP/HTTPS, IPv6 and optional PTP/IEEE-1588.
Console Port – Description of the console commands for use over the network and serial ports.
Options – Description of any optional features that your Ninja might have.

If you detect any inaccuracies or omissions, please inform us. EndRun Technologies cannot be held responsible for any technical or typographical errors and reserves the right to make changes to the product and manuals without prior notice.

Warranty

This product, manufactured by EndRun Technologies, is warranted against defects in material and workmanship for a period of two years from date of shipment, under normal use and service. During the warranty period, EndRun will repair or replace, at its option, products which prove to be defective. Products not manufactured by EndRun Technologies are warranted for ninety days or longer, as provided by the original equipment manufacturer, from date of shipment.

Extended Warranty

EndRun products are supported by a strong, comprehensive standard warranty (see paragraph above). Extended warranties are available to expand the coverage period. The extended warranty can be purchased at the time of order, or during the last year of the standard warranty period.

Limitation of Warranty

The foregoing express warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer or User, Buyer-supplied software or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

TO THE EXTENT PERMITTED BY LAW, THIS WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES, REMEDIES AND CONDITIONS, WHETHER ORAL OR WRITTEN, STATUTORY, EXPRESS, OR IMPLIED. AS PERMITTED BY APPLICABLE LAW, ENDRUN SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Warranty Repair

If you believe your equipment is in need of repair, contact EndRun Customer Support. It is important to contact us first as many problems may be resolved by phone or email. Please provide the serial number of the unit and the nature of the problem. If it is determined that your equipment will require service, we will issue an RMA number and specific shipping instructions.

Ship the unit prepaid in the original container or a container of sufficient strength and protection to EndRun Technologies. EndRun will not be responsible for damage incurred during shipment to us. Buyer shall prepay shipping charges to send product to EndRun and EndRun shall pay shipping charges to return product to Buyer. However, if returned product proves to be operating normally (not defective) then Buyer shall pay for all shipping charges. If Buyer is located outside the U.S.A. then Buyer shall pay all duties and taxes, if any.

Be sure the RMA number is clearly identified on the outside of the shipping container. Our policy is to repair the unit within 5 business days. If it is necessary to order parts or if other circumstances arise that require more than 5 days, an EndRun service technician will contact you.

Loaner units are not included as part of the standard warranty.

Repair After Warranty Expiration

If the warranty period has expired, we offer repair services for equipment you have purchased from EndRun. Contact EndRun Customer Support. It is important to contact us first as many problems may be resolved by phone or email. Please provide the serial number of the unit and the nature of the problem. If it is determined that the equipment has failed and you want EndRun to perform the repairs, we will issue you an RMA number and specific shipping instructions.

Ship the unit prepaid in the original container or a container of sufficient strength and protection to EndRun Technologies. EndRun will not be responsible for damage incurred during shipment to us. Customer is responsible for shipping costs to and from EndRun Technologies. Be sure the RMA number is clearly identified on the outside of the shipping container. After the equipment has been received we will evaluate the nature of the problem and contact you with the cost to repair (parts and labor) and an estimate of the time necessary to complete the work.

Limitation of Liability

The remedies provided herein are Buyer’s sole and exclusive remedies. EndRun Technologies shall not be liable for any direct, indirect, special, incidental or consequential damages, whether based on contract, tort or any other legal theory.

Table of Contents

Chapter *One*

Introduction

The Ninja Precision Timing Module provides highly-precise time and frequency outputs in a compact, portable module. Optimized for size, weight and power (SWaP), the Ninja requires only 35 cubic inches of mounting space and consumes less than 6 watts of power.

Using a proprietary Global Positioning System (GPS) timing receiver, the Ninja supports operation in static or dynamic (shipboard only) platforms. Proprietary multiple satellite optimal ensembling, adaptive 3rd order frequency control and TRAIM algorithms maximize the accuracy, stability, and reliability of the output signals. A variety of top-quality quartz oscillators are available to handle the full range of holdover, phase noise, and short-term stability requirements. Units equipped with our Ultra-Stable OCXO provide unmatched close-in phase noise performance and short-term stability.

The Ninja uses the GPS transmissions to precisely synchronize itself to within 25 nanoseconds of Coordinated Universal Time (UTC) (10 nanoseconds with the Calibration Option). The frequency of the internal oscillator is disciplined to match the frequency of the UTC timescale to less than 1 part in 10^{13} over 24-hour observation intervals. The time and frequency outputs are coherent after initial GPS synchronization, and synchronization is maintained via 20-bit DAC frequency control, rather than phase stepping, to provide excellent short-term stability.

For more detailed information that is not included in this manual, and links to other sites, please visit our website: endruntechnologies.com*. There you can also download firmware upgrades, the latest manuals and other documentation.*

Main Features

Robust GPS Receiver

Ninja is a GPS-based time and frequency standard designed to support mission critical applications. The proprietary GPS timing receiver in Ninja has many safeguards to protect against false GPS signals. The receiver strictly adheres to the GPS Interface Specification and performs low-level integrity checks to protect against weak corrupted signals, jamming, spoofing and accidental GPS control system errors. EndRun's GPS timing receiver technology has evolved to be highly robust against these threats to provide time and frequency outputs you can trust.

Highly-Reliable Design

The Ninja provides high performance and reliability combined with small size and low power consumption. Its internal sub-assemblies are fabricated using state-of-the-art components and processes and are integrated in a solid, high-quality chassis. Since it uses highly efficient ARM microprocessors, the Ninja is convection-cooled and the chassis is sealed to eliminate the maintenance and reliability issues associated with fan-cooled architectures. Air currents flowing over sensitive frequency control components are also eliminated so that the extraordinary short-term stability available from the Ninja oscillators is preserved. Up to nine outputs can be customer configured at the time of order supporting a range of analog and digital outputs. Details on the various outputs are described in ***Chapter 9 - Inputs/Outputs (I/O)*** and ***Appendix J - Specifications***.

Standard Features

The basic Ninja provides an RS-232 serial port and an Ethernet port with a Network Time Protocol (NTP) server. The Ninja can be managed via the network port or the serial port. See ***Chapter 3 - Console Port Control and Status,*** and ***Chapter 7 - NTP*** for more information.

Additional Outputs

Up to nine optional outputs are available via SMA connectors. Outputs supported include 5 MHz and 10 MHz sine waves, 1 PPS, pulse rates to 10 MPPS, time code (AM and DCLS), and an open-collector alarm.

Secure Network Interface

A single 10/100Base-T Ethernet port is provided as a standard feature of the Ninja with a wide variety of protocols including NTP, SNMP with Enterprise MIB, SSH, Telnet, FTP, HTTPS, and SNTP. Refer to ***Chapter 2 - Basic Installation*** for information to help you set up your network interface. The inclusion of SNMP v3 and SSH provides a very secure network interface and allows you to safely perform monitoring and maintenance activities over the network. For security-conscious users, risky protocols such as HTTP/HTTPS, Telnet, Time and Daytime will be disabled by default. To enable, see ***Chapter 5 - Security, Enable/Disable Protocols***. In addition, access via SSH, SNMP and Telnet can be restricted to specific hosts. Refer to ***Chapter 5 - Security, Restrict Access*** for further information.

Free FLASH Upgrades

Firmware and configurable hardware parameters are stored in non-volatile FLASH memory, so the Ninja can be easily and securely upgraded in the field using SSH and SCP or the local RS-232 serial port. Upgrades via FTP and Telnet are also possible although these protocols are not secure. We make all firmware upgrades to our products available to our customers free of charge. For firmware upgrade procedures refer to ***Appendix B - Upgrading the Firmware***.

Time Synchronization Components

The Ninja contains a GPS Subsystem composed of EndRun Technologies' proprietary GPS Receiver and a system quartz oscillator. All quartz oven-controlled oscillators are EndRun Technologies' proprietary design as well, and provide state-of-the-art close-in phase noise and short-term stability. The GPS Subsystem is integrated with a fanless, convection-cooled high performance ARM CPU with an Ethernet port that provides NTP. This is called the Linux Subsystem. Figure 1 shows Ninja's time synchronization components.

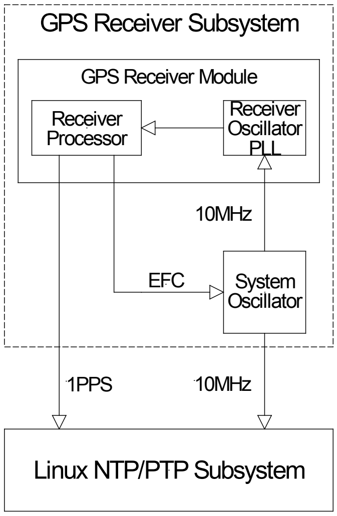


FIGURE 1 - NINJA GPS SYSTEM TIMEBASE

GPS Timing-How It Works

The time and frequency engine in the Ninja receives transmissions from satellites that are operating in compliance with the Navstar GPS Interface Specification known as IS-GPS-200. It specifies the receiver interface needed to receive and demodulate the navigation and time transfer data contained in the GPS satellite transmissions. The GPS navigation system requires a means of synchronizing the satellite transmissions throughout the constellation so that accurate receiver-to-satellite range measurements can be performed via time-of-arrival measurements made at the receiver. For the purposes of locating the receiver, measurements of the times-of-arrival of transmissions from at least four satellites are needed for maximum timing accuracy. Time transfer to a receiver at a known position from a single satellite is supported,

The GPS system designers defined *system time* to be *GPS time.* GPS time is a monotonic time scale consisting of an ensemble of high-performance cesium beam and rubidium vapor atomic frequency standards located in the monitoring stations and satellites. GPS time is measured relative to UTC, as maintained by the United States Naval Observatory (USNO), and maintained synchronous with UTC(USNO) except that it does not suffer from the periodic insertion of leap seconds. Such discontinuities would unnecessarily complicate the system's navigation mission. Contained in the data transmitted from each satellite is the current offset between GPS time and UTC(USNO). This offset is composed of the current integer number of leap seconds difference and a small residual error that is typically less than +/- 10 nanoseconds.

Each satellite in the constellation contains redundant cesium beam or rubidium vapor atomic frequency standards. These provide the timebase for all transmissions from each satellite. These transmissions are monitored from ground stations located around the world and carefully measured relative to GPS time. The results of these measurements for each satellite (i.e. correction to GPS time) are then uploaded to that satellite so that they may be incorporated into the data contained in its transmissions.

The receiver can use this data to relate the time-of-arrival of the received transmissions from that satellite to GPS time and by using the transmitted UTC parameters, to UTC(USNO).

All of this means that during normal operation, the source of the timing information being transmitted from each of the satellites is directly traceable to UTC(USNO). Due to the nature of the GPS spread spectrum Code Division Multiple Access (CDMA) modulation scheme, this timing information may be extracted by a well-designed receiver with a precision of a few nanoseconds. The GPS time and frequency engine in the Ninja does just that.

Where to Use It

Since signals from the GPS satellites are available at all locations on the globe, you may deploy the Ninja virtually anywhere. However, you must be able to install an antenna with good sky visibility, preferably on the rooftop.

Client/Slave Software

Ninja has been designed to operate in conjunction with existing public domain NTP/SNTP client software and may be used in any network environment that is using TCP/IP protocols. Although client software is available for all platforms, for the most precise applications, the Unix-like operating systems are best supported. For more information see ***Chapter 7 - NTP, Setting Up NTP Clients on Unix-like Platforms*** and ***Setting Up NTP Clients on Windows***. There is additional information about NTP Client software at this link:

endruntechnologies.com/products/ntp-time-servers/ntp-client-software

Chapter *Two*

Basic Installation

*This chapter will guide you through the most basic checkout and physical installation of your Ninja Precision Timing Module. See **Chapter 7 - NTP** for instructions on how to configure your unit as an NTP Server. Other chapters and appendices will give you the information needed to configure your installation for the maximum performance in your operating environment.*

Basic familiarity with TCP/IP networking protocols like **`ping`**, **`telnet`** *and* **`ftp`** *is required. Though some familiarity with Linux or other Unix-like operating systems would be helpful, it is not essential. If you satisfy these conditions, the instructions provided herein should guide you to a successful installation. For a brief description of some helpful Linux commands and utilities see **Appendix C - Helpful Linux Information**.*

Checking and Identifying the Hardware

Unpack and check all the items using the shipment packing list. Contact the factory if anything is missing or damaged. The Ninja Precision Timing Module shipment typically contains:

- Ninja Precision Timing Module (part #3211-xxxx-xxx, where x is a variable number)

- Ninja User Manual (part #USM3211-0000-000) on CD (part #5102-0001-000)

- Basic Cable Kit (part # 0648-0002-000) including:
  - RJ-45 to RJ-45 CAT-5 patch cable, 2 meters
  - DB9F-to-DB9F null-modem serial I/O cable

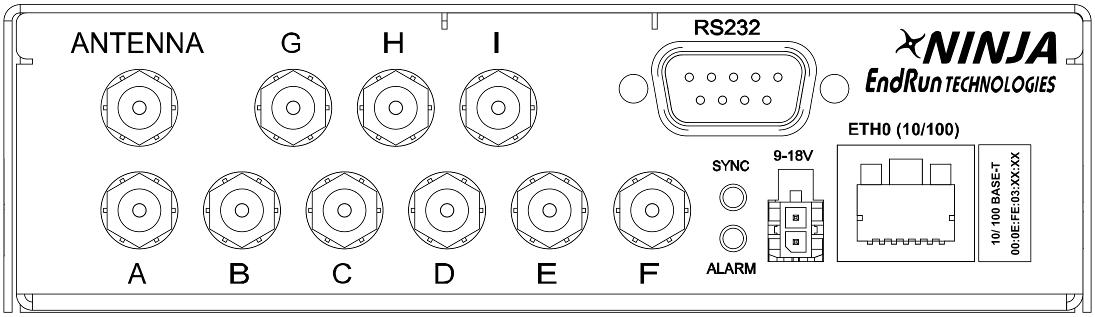
- Starter Kit (part # 0608-0009-000) including:
  - SMA/TNC antenna connector adapter
  - DC power connector and crimp pins
  - 36” DC power cable

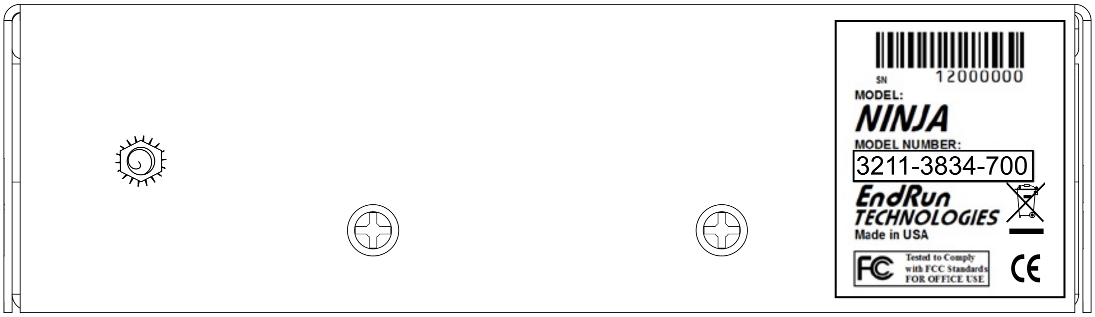
Ninja can ship with many different options. The most common are:

- External AC Power Supply (part #0623-0004-000)

- Antenna Kit (part # 0610-0009-001) including:
  - GPS antenna
  - Pipe/clamps for outside antenna mounting
  - 50’ TNC/TNC RG-59/U coaxial cable assembly

Ninja Physical Description





The drawings above show the Ninja Precision Timing Module front and rear-panels with all optional outputs (A through I) being used. (Unused outputs would be plugged.) For more information see ***Chapter 9 - Inputs/Outputs and Appendix J - Specifications***. Descriptions below briefly describe the standard and optional I/O connectors:

Sync LED — This amber LED flashes to indicate synchronization status.

Alarm LED — This red LED illuminates briefly at power-up, and thereafter whenever a serious fault condition exists.

GPS Antenna Input — This SMA connector mates with the downlead cable from the external antenna.

RS-232 Port — This DB9M connector provides the RS-232 serial I/O console interface to the Ninja. This console allows you to initialize and maintain the Ninja. See ***Chapter 3 - Console Port Control and Status*** for more information and ***Appendix J - Specifications*** for the RS-232 pin assignments.

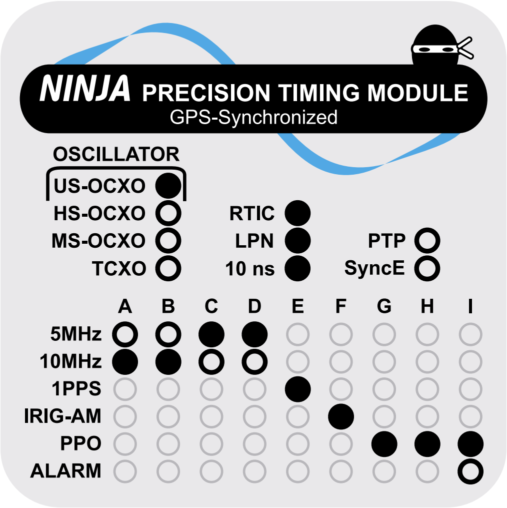
10/100Base-T Jack — This RJ-45 connector mates with the Ethernet twisted pair cable from the network. It is labeled “ETH0” with the MAC address. There is one green LED that indicates network activity. This port provides a console interface to the Ninja. See ***Chapter 3 - Console Port Control and Status*** for more information.

DC Power Input Jack	This 2-position jack provides connection to the DC power source. See details in ***Appendix J - Specifications.***
Spare Outputs	These nine SMA connectors are labeled with their connector identifier A through I. There is a legend label on the top of the unit showing the signal (if any) on each connector. Various optional outputs are listed below. For details, see ***Chapter 9 - Inputs/Outputs*** and ***Appendix J - Specifications***:

5 MHz Sine Wave
10 MHz Sine Wave
1 Pulse Per Second (PPS)
IRIG Amplitude-Modulated (AM) Time Code
Programmable Pulse Output (PPO) with DC-Shift Time Code
Open-Collector Alarm

Configuration Label

The label on top of each unit shows the configuration. The label below indicates 10-Nanosecond Calibration with an Ultra-Stable OCXO and the RTIC Option, two 10-MHz and two 5-MHz low-phase-noise (LPN) outputs, a 1PPS Output, an IRIG-B AM Output and three Programmable Pulse Outputs (PPO).



US-OCXO	●
HS-OCXO	○
MS-OCXO	○
TCXO	○
RTIC	●
LPN	●
10 ns	●
PTP	○
SyncE	○

	A	B	C	D	E	F	G	H	I
5MHz	○	○	●	●	◌	◌	◌	◌	◌
10MHz	●	●	○	○	◌	◌	◌	◌	◌
1PPS	◌	◌	◌	◌	●	◌	◌	◌	◌
IRIG-AM	◌	◌	◌	◌	◌	●	◌	◌	◌
PPO	◌	◌	◌	◌	◌	◌	●	●	●
ALARM	◌	◌	◌	◌	◌	◌	◌	◌	○

Performing a Site Survey

Using the front panel status LED indicators, it's easy to find out if your Ninja will work in your desired location:

1. Temporarily mount the antenna on the roof. Make sure that it is not blocked by large metallic objects closer than one meter. See ***Appendix E - Installing the GPS Antenna*** for more information.

2. Connect the TNC plug on the end of the antenna cable to the supplied TNC/SMA adapter. Connect the SMA plug end of the adapter to the antenna input jack on the Ninja.

3. Connect the DC power or AC power option (see details in ***Connecting the DC Power*** or ***Connecting the AC Power Option***).

Initially upon power up:

1. The unit will light the Alarm LED for about 10 seconds.

2. Then it will continuously light the Sync LED.

3. When the unit locks onto a GPS signal and begins to decode the timing data and adjust the system oscillator, the Sync LED will flash very rapidly (about a 6 Hz rate) until the data is fully decoded and the system oscillator is fully locked to the GPS frequency.

NOTE

If your Ninja has an OCXO oscillator (MS-OCXO, HS-OCXO, or US-OCXO), then it will require a 5 minute warm-up period before it begins searching for a GPS signal. If your Ninja has a TCXO oscillator then it will start searching for a GPS signal within a minute.

4. Then the Sync LED will pulse at precisely a 1 Hz rate, synchronized to UTC seconds.

At this point, the GPS Subsystem is fully synchronized, and you may proceed to permanently mounting the chassis and antenna in their desired locations. If you are unable to achieve GPS lock after 24 hours, then contact EndRun Customer Support for assistance.

NOTE

In order for the Ninja to synchronize with maximum accuracy to UTC, the delay for the cable and all devices between the antenna and the GPS receiver must be compensated. See ***Appendix E - Installing the GPS Antenna, Calibrate Your Receiver*** for more information.

Installing the Ninja

FCC NOTICE

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Mount the Ninja

Mount the unit in the desired location. After mounting the unit and connecting the antenna cable, verify that it still acquires and tracks a GPS signal. Refer to the baseplate drawing in ***Appendix J - Specifications*** for the mounting hole locations.

CAUTION

Power cable is used as a disconnection device. To de-energize equipment, disconnect the power cable. Do not install the Ninja where the operating ambient temperature might exceed 122°F (50°C).

The Ninja internal temperature must not exceed 70°C, as measured by the built-in temperature sensor accessible via the “oscctrlstat” serial port command. Internal temperature will remain in safe range if all conditions are met:

A. Base plate is in good thermal contact with external enclosure.

B. Ambient air temperature surrounding Ninja GPS Timing Module enclosure is < 50°C.

C. Adequate clearance around Ninja enclosure allows for free-convection around cover.

D. No additional thermal sources via adjacent mechanical contact.

Condition A must be met. If condition B and/or C and/or D cannot be met as stated, use built-in temperature sensor to verify adequate operating margins.

Connecting the DC Power

CAUTION

Connect the "+12VDC" terminal to the positive output of the DC power source. Connect the "GND" terminal to the negative output of the DC power source. The DC power source voltage must not exceed +18V. This until will not operate if the +/- connections are reversed; however it will not be damaged by a reverse connection. Note that the GND terminal is connected to the chassis inside the unit.

Do not install the Ninja where the operating ambient temperature might exceed 122°F (50°C).

SHOCK/ENERGY HAZARD

Install in Restricted Access Location.

Use 22 AWG copper wire only.

Branch circuit must have circuit breaker, 2A or less

Power must be sourced via two-pole disconnect device.

Connecting the AC Power Option

CAUTION

Ground the unit properly with the supplied power cord.

The socket outlet should be installed near the equipment and be easily accessible.

Power cord is used as a disconnection device. To de-energize equipment, disconnect the power cord.

Do not install the Ninja where the operating ambient temperature might exceed 122°F (50°C).

NOTE

The Ninja chassis with AC Power Option must be grounded using either the bottom mounting holes or the rear panel ground lug.

Connecting and Configuring Ethernet

Connect one end of the CAT-5 patch cable supplied with your Ninja to the RJ-45 connector labeled 10/100BASE-T. Connect the other end of the patch cable to your network through a ‘straight’ port on your switch. Do not connect it to a ‘crossover’ port on your switch.

By factory default, the Ninja will attempt to configure the Ethernet automatically via the Dynamic Host Configuration Protocol (DHCP). The Ninja will attempt to set the netmask, its IP address, the IP address of the default gateway, the domain name and the IP addresses of any nameservers, if the DHCP server is configured to provide them. You may optionally configure the Ninja to also set its hostname via DHCP, if your DHCP server is configured to provide it. You can do this by running a simple script called **`netconfig`** after your unit is up on the network.

If your network *does* use DHCP for host configuration, and you are in a hurry to get your Ninja up and running, you may proceed to ***Verifying Network Configuration*** to make sure that the network parameters were set up correctly. Otherwise, it is recommended that you read the following sections on the use of the RS-232 serial I/O port now, since they will help you in debugging any problems that you may encounter with the automatic configuration via DHCP.

If your network *does not* use DHCP, you will need to configure your Ethernet using the RS-232 serial I/O port. The following sections contain brief descriptions on how to do that.

Configuring Ethernet with the Serial Port

To configure your Ethernet with the serial port, after logging in as the *root* user, you must run a simple script called **`netconfig`**. This script will prompt you for the needed information and perform some syntax checking on your inputs. Then it will create or modify the appropriate files needed to configure the Ethernet interface. The following sections will guide you in setting up communications with the Ninja using its RS-232 serial I/O port.

Connect the RS-232 Serial I/O Port

To test serial communications with the Ninja you will need either a VT100 compatible terminal or a terminal emulation program running on your computer. We will refer to either of these as “terminal” for the remainder of this instruction.

1. Disconnect power from the Ninja.

2. Connect one end of the DB9F-to-DB9F null modem adapter cable to the serial I/O jack on the Ninja.

3. Connect the other end of the DB9F-to-DB9F null modem adapter cable to the terminal. If the serial I/O port on your terminal does not have a DB9M connector, you may need to use an adapter. Refer to ***Appendix J - Specifications*** for details on the signal wiring. *If you are using a computer for your terminal, remember which port you are using because you will need to know that in order to set up your terminal software.*

NOTE

You must use an RS-232 null-modem cable or adapter if you are connecting the Ninja to another computer. The cable included in the shipping kit is a null-modem cable.

If your computer does not have a serial port, you can use a USB port with a USB-RS232 converter similar to Gearmo GM-FTDI-8. First, connect the USB converter to your computer, then connect the converter to the null-modem cable. Finally, connect the null-modem cable to the Ninja.

Test the Serial Port

You must configure your terminal to use the serial I/O port you used in ***Connect the RS-232 Serial I/O Port*** above. You must also configure your terminal as shown below:

- Baud Rate: 19200
- Data Bits: 8
- Parity: None
- Stop Bits: 1
- Handshaking / Flow Control: OFF (both hardware and software)
- Terminal Emulation (if any): VT100 (or similar) or Linux

After configuring these parameters in your terminal, apply power to the Ninja. After about 20 seconds, your terminal should display something similar to this:

```
*****************************************************************************
*   6010-0085-000 v1.00  NinjaXXX Bootloader Mon Feb 24 19:59:00 UTC 2020   *
*****************************************************************************


Current default Kernel/Root File System:  FACTORY/FACTORY


You can:

  Override the default kernel and/or root file system boot configuration,
  and/or
  Reset the root password to the factory password,

By typing these commands:

  bootcfg=*#     (* = 0 or 1 to select FACTORY or UPGRADE kernel,
                  # = 0 or 1 to select FACTORY or UPGRADE root file system)

  pwrst=xxxxxxxx (xxxxxxxx is reset code obtained from EndRun Tech Support)

Begin typing within 5 seconds to extend the boot timeout.


Booting current default Kernel/Root File System:  FACTORY/FACTORY
```

These lines are the Linux bootloader boot prompts. These prompts will timeout after five seconds and the factory default Linux kernel and the factory default Ninja root file system will be loaded. When the Linux kernel is loaded from FLASH memory into RAM a long list of kernel-generated, informational messages is displayed as the kernel begins execution and the various device drivers are initialized. When the boot process completes, the Ninja login prompt is displayed:

```
**********************************************************************
*          Welcome to NinjaPTM console on:  NinjaPTM.your.domain
*          Tue Feb 20  2013 21:47:03 UTC
**********************************************************************

NinjaPTM login:
```

Here you may log in as “sysuser” with password “Praecis” or you may log in as the “root” user with password “endrun_1”. When logged in as “sysuser”, you may check status information and view log files but you will not be able to modify any system settings or view secure files. In order to perform system setup procedures, which includes configuring the IP network settings, you must log in as the “root” user. After correctly entering the password at this prompt,

```
password:
```

the sign on message is shown. It identifies the host system as Ninja and shows the software part number, version and build date. The out-of-the-box hostname is set to “Ninja”, and the domainname is set to “your.domain”.

```
NinjaPTM 6010-0086-000 v 1.00 Tue Feb 25 14:17:44 UTC 2020
NinjaPTM (root@NinjaPTM:~)->
```

This last line is the standard Ninja prompt. After configuring the unit, you should change the passwords using the Linux **`passwd`** command issued from the prompt.

If you do not see characters displayed by your terminal program within 30 seconds after the unit is powered up, you must troubleshoot your setup. An incorrectly wired cable or incorrect port setting in your terminal emulation program are the most common problems. Refer to ***Appendix J - Specifications*** for the signal connections for the Ninja.

Once you have successfully established communications with the Ninja, you may proceed to configure the network parameters using **`netconfig`** (see below). Then you can communicate with the Ninja over the network using **`ssh`** and synchronize your network computers to UTC using NTP.

Using netconfig to Set Up Your IP

NOTE

If you want to use the HTTPS Interface, then be sure to configure the name server IP address during the **`netconfig`** process. The HTTPS Interface will not operate properly if this is configured incorrectly. Only one name server is required, but two gives some redundancy.

The following shows the beginning of the **`netconfig`** interactive script:

```
*******************************************************************************
****************    NinjaPTM IPV4/IPV6 Network Configuration   ****************
*******************************************************************************
*                                                                             *
*   This script will configure the TCP/IPV4/IPV6 network parameters for your  *
*   Ninja.  We will first configure IPV4 and then IPV6.  Your Ninja           *
*   has one ethernet interface, called eth0.                                  *
*                                                                             *
*   You can also choose to unconfigure IPV4 or IPV6 on eth0.                  *
*                                                                             *
*   You will be able to reconfigure your system at any time by typing:        *
*                                                                             *
*   netconfig                                                                 *
*                                                                             *
*   The settings you make now will not take effect until you reboot your      *
*   Ninja, so if you make a mistake, just re-run this script before           *
*   rebooting.                                                                *
*                                                                             *
*   You will be prompted to enter your IPV4/IPV6 network parameters now.      *
*                                                                             *
*******************************************************************************
*******************************************************************************

Configure IPV4?
  (Answer yes to continue on and reconfigure eth0 for IPV4.)
  (Answer no to "unconfigure" eth0 for IPV4.  Only the
   IPV4 loopback interface will be setup.) ([y]es, [n]o):
```

After configuring your Ethernet interface, you should shutdown the Ninja and reboot it by issuing this command at the prompt:

```
NinjaPTM(root@NinjaPTM:~)-> reboot
```

Verify Network Configuration

If you are using the RS-232 serial I/O port to communicate with Ninja, you will be able to see the kernel-generated boot messages when the unit reboots. You should note the lines

```
Configuring eth0 as 192.168.1.120...
```

if you have set up a static IP address, or these lines

```
Attempting to configure eth0 by contacting a DHCP server...
```

if you are using DHCP. These appear near the end of the kernel generated boot messages.

If you are using DHCP and are not using the RS-232 serial I/O port, you will have to check the DHCP configuration information maintained by your DHCP server to determine the expected IP address and log in to the Ninja using **`ssh`** (or **`telnet`** if enabled) to verify successful DHCP configuration. Refer to the subsequent topics in this section ***Using Telnet*** and ***Using SSH***, for details on logging into the Ninja that way. Once you have logged in, you may perform the following checks.

If you are not using DHCP, the IP address shown should match the static IP address which you entered during the **`netconfig`** procedure. If so, log in as “root” at the login prompt and check the other configuration parameters using **`ifconfig`**:

```
NinjaPTM(root@NinjaPTM:~)-> ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.120  netmask 255.255.255.0  broadcast 192.168.1.255
        ether 00:0e:fe:03:12:6c  txqueuelen 1000  (Ethernet)
        RX packets 365  bytes 35713 (34.8 KiB)
        RX errors 0  dropped 54  overruns 0  frame 0
        TX packets 24  bytes 1790 (1.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
        device interrupt 25  base 0x8000

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 286  bytes 60084 (58.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 286  bytes 60084 (58.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Pay particular attention to the settings shown for **eth0**, in particular the **Mask**: setting, which should match that which is appropriate for your network. Now check the remaining configuration parameters using **route**:

```
NinjaPTM(root@NinjaPTM:~)-> route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         192.168.1.1     0.0.0.0         UG    1      0        0 eth0
loopback        *               255.0.0.0       U     0      0        0 lo
localnet        *               255.255.255.0   U     0      0        0 eth0
```

Here you are interested in the default gateway address. It should match the appropriate one for your network. If so, then the Ethernet interface of your Ninja has been successfully configured to operate on your network and you are ready to check operation of the Ninja over the network. If not, you should recheck your configuration and/or repeat the **netconfig** procedure.

If you have configured a nameserver(s) for your network, you may check that by issuing this command:

```
NinjaPTM(root@NinjaPTM:~)-> cat /etc/resolv.conf
search your.domain
nameserver 192.168.1.1
nameserver 192.168.1.2
```

Which displays the contents of the */etc/resolv.conf* file containing the domain name you entered previously using **netconfig**, and the nameserver IP address(es) to use for that domain.

Check Network Operation

With your Ninja network parameters properly configured, you are ready to test the setup using **ping** from a server or workstation that is able to access the network connected to the Ninja. Alternatively, you could **ping** one of your servers or workstations from the Ninja prompt to test the setup.

Once you have successfully established network communications with Ninja, you may perform all maintenance and monitoring activities via **ssh**. The companion utility, **scp** provides a secure means of transferring files to and from Ninja. Both of these protocols are supported in the Ninja via the OpenSSH implementations for Linux. Refer to ***Chapter 5 - Security, OpenSSH*** for more information about the secure shell protocol.

You may also use **telnet** and **ftp** to perform maintenance and monitoring activities. The Ninja provides both client and server operation using **telnet**. (See ***Using Telnet*** below.) For security reasons, only client operation is supported using **`ftp`**. You may also monitor Ninja via the HTTPS interface (see ***Using HTTPS*** below).

Using SSH

When establishing a **`ssh`** connection with your Ninja, logging in directly as *root* is permitted. When you log in as *root* via a **`ssh`** session with the Ninja, this banner will be displayed:

```
********************************************************************************
*   Welcome to the NinjaPTM SSH console on:  host.your.domain
********************************************************************************

root@M192.168.1.120’s password:
```

Here you may log in as “root” with password “endrun_1”. After correctly entering the password the sign on message is shown. It identifies the host system as Ninja and shows the software part number, version and build date:

```
NinjaPTM 6010-0086-000 v 1.00 Tue Feb 25 14:17:44 UTC 2020
NinjaPTM (root@NinjaPTM:~)->
```

This last line is the standard Ninja Precision Timing Module prompt. After configuring the unit, you should change the passwords using the Linux **`passwd`** command issued from the prompt.

Issuing **`exit`** will close the **`ssh`** session.

Using Telnet

If you want to use **`telnet`** you must first enable it. See ***Chapter 5 - Security, Enable/Disable Protocols*** for instructions. When establishing a **`telnet`** connection with your Ninja, logging in directly as *root* is not permitted. This is a security measure that makes it slightly more difficult to gain access by simply trying passwords, since it is also necessary to know the name of a user. When you initiate a **`telnet`** session with the Ninja, this banner will be displayed:

```
*********************************************************************************
*        Welcome to NinjaPTM telnet console on:  host.your.domain
*********************************************************************************

host login:
```

Here you may log in as “sysuser” with password “Praecis”. When logged in as “sysuser”, you may check status information and view log files but you will not be able to modify any system settings or view secure files. After correctly entering the password at this prompt,

```
Password:
```

the sign on message is shown. It identifies the host system as NinjaPTM and shows the software part number, version and build date:

```
NinjaPTM 6010-0086-000 v 1.00 Tue Feb 25 14:17:44 UTC 2020
NinjaPTM (root@NinjaPTM:~)->
```

This last line is the standard Ninja Precision Timing Module prompt. After configuring the unit, you should change the passwords using the Linux **`passwd`** command issued from the prompt.

To gain *root* access, you must now issue the “super user” command at the prompt:

```
NinjaPTM(root@NinjaPTM:~)-> su root
```

You will then be prompted for the password, which is “endrun_1”, and be granted *root* access to the system. To leave “super user” mode, issue the command **`exit`**. Issuing **`exit`** again will close the **`telnet`** session.

Using HTTPS

If you want to use HTTPS you must first enable it. See ***Chapter 5 - Security, Enable/Disable Protocols*** for instructions. You may monitor the status of the Ninja via the HTTPS interface. For security reasons, you may not change any settings. See ***Chapter 4 - HTTP/HTTPS*** for more information.

IMPORTANT

SSH, Telnet, SNMP and HTTPS are all enabled with default passwords. To ensure security, change the passwords or disable the protocols.

To change the passwords for SSH, Telnet and HTTPS use the Linux **`passwd`** command. To change the passwords/community strings for SNMP see ***Chapter 6 - SNMP***.

To disable Telnet, SSH, SNMP and HTTPS see ***Chapter 5 - Security, Disable Protocols***.

Connecting Instruments to the Ninja

SMA jacks provide the means of connecting your equipment to the Ninja. The Ninja can provide up to nine optional outputs. Options are:

Digital outputs (1PPS or PPO and IRIG-DC)
Amplitude-modulated (AM) time code
Open-collector alarm
5 MHz and 10 MHz sine waves

The AM time code and the digital outputs are capable of driving properly terminated coaxial cables. These signals are DC-coupled and sourced from Advanced CMOS (ACMOS) drivers which are able to maintain output TTL levels into a 50-ohm load.

The low-phase noise, spectrally pure sine wave outputs are capable of driving 1 Vrms into a 50-ohm load. Care should be taken not to short circuit these outputs or to connect them to other voltage sources.

If your unit is equipped with the optional Alarm Output, care should be taken not to directly connect this open-collector output to a voltage source. A series current-limiting resistor of at least 1k ohms in value should be used. The pull-up voltage must not exceed 40V.

Refer to ***Chapter 9 - Inputs/Outputs, Output Options*** and to ***Appendix J - Specifications, Optional Outputs*** for more information these signals.

Chapter *Three*

Console Port Control and Status

This chapter describes Ninja control and status commands used via the Linux console. The console is accessed via the Ethernet port or the RS-232 serial port. Ninja supports several application-specific commands for configuration and for monitoring the performance and status of the Linux and GPS Subsystems.

You do not need knowledge of Linux commands in order to operate Ninja. However, Ninja does support a subset of the standard Linux commands and utilities and it uses the **`bash`** *shell, which is the Linux standard, full-featured shell. A wealth of information is available from a variety of other sources on Linux.*

Ninja-specific commands will be described in this chapter. For a brief description of some of the most useful Unix/Linux commands, see ***Appendix C - Helpful Linux Information****.*

Console Ports

Two interface ports are available on Ninja. One is a 10/100Base-T Ethernet port and the other is an RS-232 serial port. A network cable and a serial cable are provided with each Ninja shipment. The serial cable is wired as a null-modem adapter and can be used to connect Ninja to the serial port on your computer. Detailed specifications on the ports, including the RS-232 pinout, are in ***Appendix J - Specifications***.

General Linux Operation

You do not need to know Linux in order to operate Ninja. However, for those interested, the command shell used by Ninja is the Linux standard: **`bash`**. All commands and file names are case sensitive, which is standard for Unix-like operating systems. For a brief description of some of the most useful Unix/Linux commands, see ***Appendix C - Helpful Linux Information***.

If you are unfamiliar with Unix-like operating systems, and you would like to be able to more closely monitor or optimize the performance of your Ninja then you should consult good Linux reference books or the Linux Documentation Project at::

http://www.tldp.org

Available User Commands

COMMAND	FUNCTION
accessconfig	Interactive script that guides you in configuring **`telnet, ssh`** and **`snmpd`** access to Ninja that is limited to specific hosts. The resulting */etc/hosts.allow* and */etc/hosts.deny* files are saved to the non-volatile FLASH disk. Factory default configuration allows access by all hosts.
antfltmask	Prints the current setting for the Antenna Fault Mask. See the **`setantfltmask`** command.
caldelay	Prints the calibration delay. See the **`setcaldelay`** command.
clearalmanac	For use with a GPS simulator. See ***Appendix I - Operation with a GPS Simulator***.
cpuio	Returns the current settings for any optional outputs such as the Programmable Pulse Output (PPO).
cpuioconfig	An interactive utility that allows you to modify the settings for any optional outputs, such as the Programmable Pulse Output (PPO).
cpustat	Prints the current Linux CPU core temperature, system load as percent of maximum and free memory available.
dumpalmanac	Prints the current GPS almanac data for all satellites in Yuma format.
dumpephemeris	Prints the last received GPS ephemeris data for all satellites in Rinex 3.01 format.
faultstat	Prints the summary of all system fault states in a user-friendly format.
gpsdynmode	Prints the GPS dynamic mode currently in effect. See the **`setgpsdynmode`** command.
gpsionoinfo	Prints the GPS Ionospheric Model Almanac parameters per the IS-GPS-200.
gpslastfix	Prints the last computed GPS position fix.
gpsrefpos	Prints the GPS reference position. See the **`setgpsrefpos`** command.
gpsrefpos_ecef	Prints the GPS reference position in Earth-Centered Earth-Fixed format.
gpsreset	Sends the reset command to the GPS receiver.
gpsstat	Prints the GPS Subsystem status information.
gpstrkstat	Prints the GPS satellite tracking status. Azimuth, elevation and signal level (C/No) are shown for each satellite.
gpsutcinfo	Prints the GPS UTC Almanac parameters per the IS-GPS-200. Also shows the current calculated GPS-UTC offset, which includes leap seconds and a small sub-second offset.
gpsversion	Prints the GPS Receiver firmware and FPGA version information.

help help command	Prints help for all Ninja-specific (not Linux) commands. Prints command-specific help. For example: **`help gpsstat`**.
inetdconfig	Interactive script that allows you to configure the list of protocol servers which are started by the **`inetd`** server daemon running in Ninja.
inhibitoutputsmode	Prints the current InhibitOutputsMode setting which controls the timing output signals (time codes and pulse rates) prior to initial lock to GPS. See the **`setinhibitoutputsmode`** command.
ionostat *(optional)*	Prints the status of the optional RTIC. See ***Chapter 12 - Real-Time Ionospheric Corrections*** for more information.
kernelversion	Prints the Linux operating system kernel version.
logrinex	Controls logging of raw GPS pseudorange and carrier phase observations in Rinex 2.11 format.
netconfig	Interactive script that allows you to configure the IP network subsystem of Ninja.
ntpconfig	Interactive script that guides you in configuring the NTP Subsystem. Allows configuration of MD5 authentication and broadcast/multicast mode. All parameters are retained in non-volatile FLASH disk storage.
ntpstat	Prints the values of several key parameters indicating the status of the NTP daemon. These include the current offset between the NTP-steered system clock and the GPS Subsystem clock, and the current counts of received packets, sent packets and dropped packets. In addition the current sent packet rate is shown.
oscctrlstat	Prints the system oscillator disciplining parameters.
passwd	Used to change the password for the user that you are logged in as. This is a Linux command.
rcvrserialnumber	Prints the serial number of the GPS Receiver.
rcvrstat	Prints the status of the GPS Receiver.
resetlastgpswn	For use with a GPS simulator. See ***Appendix I - Operation with a GPS Simulator***.
resetleaphistory	For use with a GPS simulator. See ***Appendix I - Operation with a GPS Simulator***.
rticmode *(optional)*	Prints the mode of the RTIC Option - either ON or OFF. See ***Chapter 12 - Real-Time Ionospheric Corrections*** for more information.
setantfltmask	Command to enable or mask the Antenna Fault. See the **`antfltmask`** command.
setcaldelay	An interactive utility that allows you to change the clock calibration delay. See the **`caldelay`** command.
setgpsdynmode	Command to set the dynamic mode of operation of the GPS Subsystem. See the **`gpsdynmode`** command.

setgpsrefpos	Interactive utility that prompts you for an accurate reference position, performs syntax and argument validity checking then passes the position to the GPS Subsystem. See the **`gpsrefpos`** command.
setinhibitoutputsmode	Sets the InhibitOutputsMode setting which controls the timing output signals (time codes and pulse rates) prior to initial lock to GPS. See the **`inhibitoutputsmode`** command.
setrticmode *(optional)*	Sets the RTIC mode - either ON or OFF. See ***Chapter 12 - Real-Time Ionospheric Corrections*** for more information.
setsigfltmask	Command to enable or mask the Signal Loss Fault. See the **`sigfltmask`** command.
settfomfltlvl	Command to change the TFOM fault level
sigfltmask	Prints the current setting for the Signal Loss Fault mask. See the **`setsigfltmask`** command.
syskernel	Prints the currently booted linux kernel, either 0 or 1, where 0 is the factory-installed kernel and 1 is the upgraded kernel.
sysosctype	Prints the installed system oscillator type, which is one of HP-TCXO, MS-OCXO, HS-OCXO or US-OCXO.
sysrootfs	Prints the currently loaded linux root file system image, either 0 or 1, where 0 is the factory-installed root file system, and 1 is the upgraded root file system.
sysstat	Prints detailed NTP status information. Included is the offset of the NTP-steered system clock to the GPS Subsystem clock, the NTP daemon leap indicator bit values, the TFOM, the time of the most recent update and the current leap seconds value.
systemio	Returns the current settings for installed system outputs such as 1 PPS and time code.
systemioconfig	An interactive utility that allows you to modify the settings for the system options.
systimemode	Prints the time mode settings in effect for the optional Time Code Output. See the **`systimemodeconfig`** command.
systimemodeconfig	Interactive utility that guides you in configuring the time mode settings for the optional Time Code Output. Allows setting to the GPS or UTC timescale. See the **`systimemode`** command.
sysversion	Prints the Linux root file system version information.
tfomfltlvl	Prints the current setting for the TFOM Fault Level.
triggerppo *(optional)*	Starts execution of the Trigger PPO function. See ***Chapter 9 - Inputs/Outputs (I/O), Programmable Pulse Output (PPO) Option*** for more information.
updatekernelflag	Command to update the flag stored in FLASH that is read by the Linux bootloader at boot time to select operation with either the factory-installed or upgraded kernel.

updaterootflag	Command to update the flag stored in FLASH that is read by the Linux bootloader at boot time to select operation with either the factory-installed or upgraded root file system.
upgradekernel	Command that performs the Linux Kernel upgrade process.
upgradercvr	Command that performs the GPS Receiver upgrade process.
upgradercvrfpga	Command that upgrades the FPGA resident on the GPS Receiver.
upgraderootfs	Command that performs the Linux Root File System upgrade process.

Detailed Command Descriptions

accessconfig

This command starts an interactive script that will allow the root user to configure access limitation via **`telnet`**, **`ssh`** and **`snmp`** to Ninja. By default, the unit is configured to allow access by all users. If you need to limit **`telnet`**, **`ssh`** or **`snmp`** access, e.g. for security reasons, you must run this script as root from either the RS-232 serial I/O port or from a **`telnet`** or **`ssh`** session.

This script modifies these files: */etc/hosts.allow* and */etc/hosts.deny*. These are non-volatilely stored in the FLASH disk */boot/etc* directory. You must reboot Ninja after running this script for the changes to take effect.

Command: **`accessconfig`**
Ninja reply: Interactive script is started.

antfltmask

This command displays the current setting for the Antenna Fault Mask.

Command: **`antfltmask`**
Ninja reply: **`Antenna Fault is ENABLED`**

caldelay

This command displays the current calibration delay setting for the GPS antenna cable. The allowable calibration delay range is ±500,000 nanoseconds. See ***Appendix E - Installing the GPS Antenna, Calibrate Your Receiver*** for more details.

Command: **`caldelay`**
Ninja reply: **`+0 nanoseconds`**

clearalmanac

This command is for use with a GPS simulator   Refer to ***Appendix I - Operation with a GPS Simulator*** for more information.

cpuio

This command displays the current settings for any optional outputs, such as the Programmable Pulse Output (PPO). See ***Chapter 9 - Inputs/Outputs, Output Options*** for more information.

Command: `cpuio`

Ninja reply:

```
CPU I/O E - 1 PPS OUTPUT is Installed --
  Current Setting = (See systemio command)

CPU I/O F - AM TIME CODE is Installed --
  Current Setting = (See systemio command)

CPU I/O G - PROGRAMMABLE PULSE OUTPUT is Installed --
  Current Setting = 1M PPS
```

cpuioconfig

This command starts an interactive shell script that will allow the root user to change the settings of any installed, user-selectable, optional outputs. See ***Chapter 9 - Inputs/Outputs, Programmable Pulse Output (PPO) Option*** for more information.

Command: `cpuioconfig`

Ninja reply: Interactive shell script is started.

cpustat

This command shows a group of key values for monitoring the health of the Linux CPU and operating system status. The format is:

```
YYYYMMDD.HH:MM:SS LLL% FREEkB +TT.TC
```

Where:

YYYY is the year of the UTC timestamp of the most recent update.

MMDD is the month and day-of-month of the UTC timestamp of the most recent update.

HH:MM:SS is the hour, minute and second of the UTC timestamp of the most recent update.

LLL% is the percentage of maximum load as returned using the Linux **`vmstat`** command.

FREEkB is the available free memory in kilobytes as returned using the Linux **`vmstat`** command.

+TT.TC is the temperature in degrees centigrade of the Linux CPU die temperature.

Command: `cpustat`

Ninja reply:

```
20130116.22:24:00  23% 320056kB  +67.9C
```

dumpalmanac

This command prints the current GPS almanac data for all satellites in Yuma format.

Command: **`dumpalmanac`**
Ninja reply: Example shown below, for satellite 01

```
****** Week 1890 almanac for PRN-01 *******
ID:                          01
Health:                      000
Eccentricity:                5.217074882e-03
Time of Applicability(s):    5.038080000e+05
Orbital Inclination(rad):    9.637917876e-01
Rate of Right Ascen(r/s):   -7.828898418e-09
SQRT(A)  (m 1/2):            5.153603027e+03
Right Ascen at Week(rad):   -1.158766985e+00
Argument of Perigee(rad):   +4.515030086e-01
Mean Anom(rad):             -2.085680962e+00
Af0(s):                     +1.621245974e-05
Af1(s/s):                   +0.000000000e+00
week:                        1.890000000e+03
```

dumpephemeris

This command prints the current GPS ephemeris data for all satellites in Rinex 3.01 format.

Command: **`dumpephemeris`**
Ninja reply: Example shown below, for satellite 01

```
G01 2016 03 30 20 00 00 +1.58743932843e-05 +1.13686837722e-12 +0.00000000000e+00
     +8.00000000000e+01 -1.13437500000e+01 +4.30446501253e-09 -2.12893752067e+00
     -5.90458512306e-07 +5.20769448485e-03 +8.78982245922e-06 +5.15364822960e+03
     +3.31200000000e+05 -1.26659870148e-07 -1.15738910133e+00 -5.58793544769e-09
     +9.63788168503e-01 +2.12843750000e+02 +4.50864037440e-01 -7.82032574796e-09
     +3.27513642258e-10 +1.00000000000e+00 +1.89000000000e+03 +0.00000000000e+00
     +2.00000000000e+00 +0.00000000000e+00 +5.12227416039e-09 +8.00000000000e+01
     +3.24258000000e+05 +4.00000000000e+00 +0.00000000000e+00 +0.00000000000e+00
```

faultstat

This command returns the summary of all system and receiver fault states in a user-friendly format. An example is shown below. For details on the various faults see ***Appendix G - System Faults***.

Command: **faultstat**

Ninja reply:

```
System Fault Status:
System Oscillator DAC--------------------------> OK
GPS Signal-------------------------------------> OK
GPS Receiver FPGA Configuration----------------> OK
GPS Receiver FLASH Writes----------------------> OK
Local Oscillator Synthesizer Tuning------------> OK
Local Oscillator Synthesizer-------------------> OK
GPS Reference Time-----------------------------> OK
GPS Receiver Oscillator------------------------> OK
GPS Antenna Short------------------------------> OK
GPS Antenna Open-------------------------------> *FAULT*
GPS Receiver Oscillator PLL--------------------> OK
NTP Polling------------------------------------> OK
GPS Communication------------------------------> OK
```

gpsdynmode

This command displays the current GPS Subsystem dynamic mode of operation. It has two possible settings: OFF or ON. When it is OFF, it is assumed that Ninja is installed in a stationary location. When it is ON, it is assumed that Ninja is installed on a moving platform. Dynamic mode is intended for shipboard applications only.

When the dynamic mode is OFF, Ninja will use its accurate reference position to implement Timing Receiver Autonomous Integrity Monitoring (TRAIM) for the utmost in reliability during any GPS system faults. In addition, single satellite operation is possible once an initial accurate position has been determined.

When the dynamic mode is ON, only a very minimal TRAIM algorithm is in effect because the accurate reference position is not static. In addition, a minimum of four satellites must be visible and only 3-D position fixes are used. When the dynamic mode is ON, the source reported for the accurate reference position by **gpsrefpos** is set to DYN. Dynamic mode is intended for shipboard applications only.

Command: **gpsdynmode**

Ninja reply:

```
OFF
```

gpsionoinfo

This command provides the current and previously received GPS ionospheric model coefficients. Also shown are the week number and time-of-week of the almanac transmissions that contained the two sets of model coefficients.

Command: **gpsionoinfo**

Ninja reply:

```
GPS Ionosphere Model Almanac Parameters:
  WN_a = 1890  T_oa = 503808
  Alpha0 = +1.676e-08  Alpha1 = +7.451e-09  Alpha2 = -1.192e-07  Alpha3 = -5.960e-08
  Beta0  = +1.106e+05  Beta1  = +1.638e+04  Beta2  = -2.621e+05  Beta3  = -6.554e+04
  WN_a = 1890  T_oa = 405504
  Alpha0 = +1.676e-08  Alpha1 = +7.451e-09  Alpha2 = -1.192e-07  Alpha3 = -5.960e-08
  Beta0  = +1.106e+05  Beta1  = +1.638e+04  Beta2  = -2.621e+05  Beta3  = -6.554e+04
```

gpslastfix

This command provides the last computed GPS position. When tracking four or more satellites, the GPS Receiver may provide a 3D-position fix, otherwise an overdetermined time-only solution will be computed. The last-fix position is unaveraged and typically less accurate than the reference position, but it does provide a good indication that the receiver is working properly. Position is provided in latitude, longitude and height above the WGS-84 ellipsoid. PDOP, HDOP, VDOP and TDOP are also shown. OORCnt is the out-of-range counter which may be non-zero when Ninja has been moved more than 200 meters from its previous location without being placed in UNKNOWN mode using the **setgpsrefpos** command. It could also be non-zero during a GPS system anomaly, in which case Ninja TRAIM algorithms will be operating to maintain the integrity of the system timing.

Command: **gpslastfix**

Ninja reply:

```
LAST POSITION FIX = N38d24m54.33s W122d45m10.99s +00003.5 meters
PDOP:  2.90 HDOP:  1.48 VDOP:  2.49 TDOP:  2.03 OORCnt:   0
```

gpsrefpos

This command displays the current GPS Subsystem reference position. The source of the position, which is one of UNK (unknown), DYN (dynamic), USR (user entered) or AVG (24 hour average of GPS fixes) is displayed first. The WGS-84 latitude and longitude in degrees, minutes, seconds format and the height above the WGS-84 reference ellipsoid in meters follow. Also, the current RMS PDOP of the average position and the current count of averages are shown. If the reference position source is not AVG, then these two values will be zero.

Command: **gpsrefpos**

Ninja reply:

```
CURRENT REFERENCE POSITION = AVG N38d24m54.30s W122d45m10.95s +00004.1 meters
RMSPDOP:  2.4 AvgCnt: 28800
```

gpsrefpos_ecef

This command displays the current GPS Subsystem reference position in Earth-Centered Earth-Fixed format. The source of the position, which is one of UNK (unknown), DYN (dynamic), USR (user entered) or AVG (24 hour average of GPS fixes) is displayed first. The WGS-84 X, Y, Z coordinates in meters are displayed next. Then, the current RMS PDOP of the average position and the current count of averages are shown. If the reference position source is not AVG, then these two values will be zero.

Command: **gpsrefpos_ecef**

Ninja reply:

```
CURRENT REFERENCE POSITION ECEF = AVG X: -2707225.0 m Y: -4208361.2 m Z: +3941650.2 m
RMSPDOP:  2.4 AvgCnt: 28800
```

gpsreset

This command sends a reset command to the GPS receiver processor.

Command: **gpsreset**

Ninja reply: **OK**

gpsstat

This command allows you to query the status of the GPS Subsystem. During normal operation, the system polls the GPS Subsystem every 10 seconds. The results of this poll are used to steer the system clock and are saved to a log file. This command parses and formats the data contained therein and prints this fixed-length string having these fields:

```
LKSTAT TFOM = ? YEAR DOY HH:MM:SS LS LF S N AGC OSCDAC SN.R FLTS
```

Where:

LKSTAT is the tracking status of the GPS Subsystem, either LOCKED or NOTLKD.

TFOM = ? is a value between 3 and 9 and indicates clock accuracy.
A detailed explanation of TFOM is in ***Appendix A - TFOM***.

YEAR is the year of the UTC timestamp of the most recent update.

DOY is the day-of-year of the UTC timestamp of the most recent update.

HH:MM:SS is the hour, minute and second of the UTC timestamp of the most recent update.

LS is the current number of leap seconds difference between the UTC and GPS timescales (18 at the time of this writing).

LF is the future (at the next UTC midnight) number of leap seconds difference between the UTC and GPS timescales (18 at the time of this writing).

S is the Signal Processor State, one of 0 (Acquiring), 1 (GPS Locking), 2 (GPS Locked), 9 (Warming Up).

NN is the number of GPS satellites being tracked, 0 to 12.

AGC is the RF front-end Automatic Gain Control 8-bit DAC value. Typical range is 125 to 200, with larger numbers implying higher gain being used.

OSCDAC is the system oscillator Oscillator Voltage Control 20-bit DAC value, 0 to 1048575 with larger numbers implying higher oscillator frequency. Typical range is 320000 to 680000.

SN.R is the received GPS Carrier Signal-to-Noise Ratio, 0.00 to 99.9, measured in dB in a 1Hz bandwidth. Typical range is 39 to 50.

FLTS is the fault status for the GPS Subsystem. This is a numeric value consisting of four hexadecimal characters where each bit indicates a particular system fault. Assertion of any of these bits will light the Alarm LED. Bit definitions are shown below. Decoding the bits can be difficult for non-programmers. For a more user-friendly method of reading the fault status use the **`faultstat`** command. For details on each system fault see ***Appendix G - System Faults***.

	Bit 3	Bit 2	Bit 1	Bit 0
Char 0	FLASH Writes	GPS Rcvr FPGA Configuration	GPS Signal	System Oscillator DAC
Char 1	GPS Receiver Oscillator	GPS Reference Time	Synthesizer	Synthesizer Tuning
Char 2	NTP Polling	Rcvr Osc PLL	Antenna Open	Antenna Short
Char 3	N/A	N/A	N/A	GPS Comm

The example reply below indicates that there has been a period without tracking a GPS signal that exceeded the time-out period, that there was a FLASH Write Fault and that there is an Antenna Short Fault.

```
Command:      gpsstat
Ninja reply:
LOCKED TFOM = 4 2001 092 04:48:56 13 13 2  7 151 328605 41.6 010A
```

gpstrkstat

This command displays the current GPS Subsystem satellite tracking status. A list of twelve satellite numbers along with azimuth, elevation and C/No is displayed for each receiver channel. Satellite number 0 is an invalid number and indicates that no satellite is being tracked on that channel. Valid satellite numbers range from 1 to 32. Azimuth and elevation are in degrees and C/No is in dB.

```
Command:      gpstrkstat
Ninja reply:  Ch SV Azimuth   Elev C/No
               1 23 -108.41 +15.70 41.7
               2 11 -118.21 +45.58 46.9
               3 22 +107.41 +21.04 37.9
               4 14  +52.10 +29.76 40.4
               5 32  -40.36 +58.18 45.2
               6  1  -79.14 +55.53 46.6
               7 31 +127.87 +62.60 47.3
               8  0   +0.00  +0.00  0.0
               9  0   +0.00  +0.00  0.0
              10  0   +0.00  +0.00  0.0
              11  0   +0.00  +0.00  0.0
              12  0   +0.00  +0.00  0.0
```

gpsutcinfo

This command displays the IS-GPS-200 almanac parameters which are used to relate GPS time to UTC. The first line of output contains the current (LS) and future (LSF) leap second values and the GPS week number (WN_lsf) and day of week (DN) at the end of which the future leap second will take effect. This could be in the past if a leap second insertion has recently taken place. Leap second events typically occur every one-and-a-half to three years on either June 30 or December 31.

The second line of output contains the parameters for calculating the small residual offset between the GPS master clock ensemble and UTC(USNO). This is typically less than 10 nanoseconds. The remaining output shows the current value of the GPS-UTC offset.

```
Command:        gpsutcinfo
Ninja reply:    GPS UTC Almanac Parameters:
                LS = 18  LSF = 18  WN_lsf = 1694  DN = 7
                a0 = +9.313226e-10  a1 = -1.243450e-14  WN_t = 1727  t_ot =  61440
                Current (GPS - UTC) Offset:
                GPS - UTC = (18 + 3.810e-09) s @ WN = 1726, TOW = 434757
```

gpsversion

This command displays the firmware and hardware versions of the GPS Receiver.

```
Command:        gpsversion
Ninja reply:
F/W 6010-0088-000 Ver 1.00 - FPGA 6020-0018-000 Ver 0085 - Sep 9 11:08:23 2019
```

help

This command displays a list of Ninja commands (not Linux commands). To get help on a particular command you would type **help**, followed by the command.

Command: **help**
Ninja reply: Ninja commands are displayed.
Command: **help gpsstat**
Ninja reply: Information specific to the **gpsstat** command is displayed.

inetdconfig

This command starts an interactive script that allows you to configure the list of protocol servers which are started by the **inetd** super-server daemon running in Ninja. Three protocol servers may be configured: Time, Daytime, and Telnet. By default, the unit is configured to start all of these protocol servers. If you need to disable start-up of some or all of these, e.g. for security reasons, you must run this script as *root* from either the RS-232 serial I/O port or from a **telnet** or **ssh** session.

This script modifies the */etc/inetd.conf* file, which is non-volatilely stored in the FLASH disk */boot/etc* directory. You must reboot Ninja after running this script for the changes to take effect.

Command: **inetdconfig**
Ninja reply: Interactive script is started.

inhibitoutputsmode

This command shows the current inhibit mode of the timing signal outputs (if any) prior to initial system lock to the GPS receiver. If it is set to ON, then all timing signals will be inhibited (not present) until the system is locked to the timing receiver. If it is set to OFF, then the timing signals will be present at all times.

```
Command:        inhibitoutputsmode
Ninja reply:    OFF
```

ionostat (Optional)

This command is only available if the RTIC option has been installed. Refer to ***Chapter 12 - Real-Time Ionospheric Corrections*** for more information.

kernelversion

This command prints the current Linux operating system kernel firmware version.

Command: **`kernelversion`**
Ninja reply:

```
6010-0087-000_v1.00 Linux Kernel 4.14.88-Ninja Wed Sep 11 23:40:28 2019
```

logrinex

This command is used to control logging of raw GPS pseudorange and carrier phase observations in the Rinex version 2.11 format to the */home/rinex/rinex.log* file. It requires one argument, which may be either ON or OFF.

If ON is asserted and logging is not currently in progress, then a new log of 2880 observations taken every 30 seconds will begin. Assuming full constellation visibility, this will take 24 hours to complete. If a previously-initiated log is in progress, then an ON assertion will be ignored.

> IMPORTANT!!! The data is always appended to the */home/rinex/rinex.log* file, so if you are performing multiple logs, it is up to you to manage the file and remove it before it grows too large and fills the limited storage available.

If OFF is asserted, then any currently active log will be terminated. Data logged prior to the OFF assertion is not discarded.

Once a complete log is available, you will need to prepend the Rinex version 2.11 header to it. An example header is contained in a file in the same directory with the log file for reference:

/home/rinex/rinex_header_2.11

You will need to edit various fields in it for your requirements. When you have a complete log with header, you may submit it to the Natural Resources Canada website for Precise Point Position (PPP) processing using the CSRS-PPP tool. This is the only freely-accessible PPP tool that will perform single-frequency PPP processing that we are aware of. The web URL is :

http://www.nrcan.gc.ca/earth-sciences/geomatics/geodetic-reference-systems/tools-applications/10925#ppp

Command: **`logrinex ON`**
Ninja reply: Rinex Log Control Command Successful

netconfig

This command starts an interactive script that allows you to configure the IP network subsystem of Ninja. By default, the unit is configured to configure itself using the Dynamic Host Configuration Protocol (DHCP). If you need to set up static IP configuration, you must run this script as *root* from the RS-232 serial I/O port during the installation process. Refer to ***Chapter 2 - Basic Installation, Using netconfig to Set Up Your IP*** for details on the use of the command.

This script creates or modifies these files: */etc/HOSTNAME*, */etc/hosts*, */etc/networks, /etc/resolv.conf* and */etc/rc.d/rc.inet1.conf*. All of these are non-volatilely stored in the FLASH disk */boot/etc* directory. You must reboot Ninja after running this script for the changes to take effect.

Command: `netconfig`
Ninja reply: Interactive script is started.

ntpconfig

This command starts an interactive script that allows you to configure the NTP Subsystem of Ninja. By default, the unit is configured to authenticate its replies to clients using its default MD5 keys in the */etc/ntp.keys* file. If you need to create your own MD5 keys (recommended) or set up broadcast/ multicast operation, you must run this script as root. Refer to ***Chapter 7 - Configuring the NTP Server*** for details on the use of this command.

The two files that are modified are */etc/ntp.keys* and */etc/ntp.conf*. Both of these are non-volatilely stored in the FLASH disk */boot/etc* directory. You must reboot Ninja after running this script for the changes to take effect.

Command: `ntpconfig`
Ninja reply: Interactive script is started.

ntpstat

This command provides some key information regarding the operation of the NTP daemon. It shows the current offset between the NTP-steered system clock and the GPS Subsystem, the counts of received, sent and dropped packets, and the sent packet rate. The format of the response is:

```
YYYYMMDD.HH:MM:SS +S.sssssssss RCVDCNT SENTCNT SENT/sec DROPCNT
```

Where:

YYYY is the year of the UTC timestamp of the most recent update received from the GPS Sub system.

MMDD is the month and day-of-month of the UTC timestamp of the most recent update received from the GPS Subsystem.

HH:MM:SS is the hour, minute and second of the UTC timestamp of the most recent update received from the GPS Subsystem.

+S.sssssssss is the offset in seconds between the NTP system clock and the GPS Subsystem clock. Positive implies that the system clock is ahead of the GPS Subsystem clock.

RCVDCNT is a count of the number of NTP packets received since startup of the NTP daemon. This is a 32-bit counter so it will wrap back to zero after 4,294,967,295 packets.

SENTCNT is a count of the number of NTP packets sent since startup of the NTP daemon. This is a 32-bit counter so it will wrap back to zero after 4,294,967,295 packets.

SENT/sec is the current rate of NTP packets being sent per second.

DROPCNT is a count of the number of NTP packets dropped since startup of the NTP daemon. This is a 32-bit counter so it will wrap back to zero after 4,294,967,295 packets.

Below is an example of a typical response to this command:

```
Command:      ntpstat
Ninja reply:
20130117.00:02:40 -0.000000051 129127988 129015079  1594.4/sec      15
```

oscctrlstat

This command displays the current values of the system oscillator control parameters. These parameters are related to the disciplined system oscillator. The command formats the data and prints this fixed-length string having these fields:

```
YYYYMMDD.HH:MM:SS LKSTAT COAST ESTERR MEASERR TIMEDEV AGERATE TAU EFCDAC TEMP
```

Where:

YYYY is the year of the UTC timestamp of the most recent update received from the GPS Subsystem.

MMDD is the month and day-of-month of the UTC timestamp of the most recent update received from the GPS Subsystem.

HH:MM:SS is the hour, minute and second of the UTC timestamp of the most recent update received from the GPS Subsystem.

LKSTAT is the GPS Subsystem oscillator control status, either INIT (waiting for initial data), LKG (locking), LKD (locked), STEP (rejecting outlier data) or COAST (no new data).

COAST is the number of seconds the GPS Subsystem has been in coast mode (unlocked to GPS).

ESTERR is the estimated time error of the GPS Subsystem when in coast mode, in seconds.

MEASERR is the last measured time offset of the GPS Subsystem to GPS while locked, in seconds.

TIMEDEV is the time deviation (TDEV) of the offset measurements in seconds. The tau associated with this measurement is three seconds, which is the update interval of the position fixes received from the GPS Receiver.

AGERATE is the regression-computed system oscillator ageing rate per day (several-hour delay before the first measurements are displayed).

TAU is the system oscillator control loop averaging time constant, in seconds. It's value is automatically adjusted to maintain optimum offset and stability.

EFCDAC is the system oscillator Electronic Frequency Control 20-bit DAC value. The system automatically sets this value to remove frequency errors. Values may range from 0 to 1048575. Values close to the maximum or minimum will set the DAC fault flag that will appear in the fault status display. The Time/Status display will also indicate a fault condition.

TEMP is the chassis internal temperature in °C.

Below is an example of a typical response to this command:

Command: **`oscctrlstat`**
Ninja reply:

```
20130117.00:23:10 LKD     0 6.26e-09 -6.26000e-09 1.25e-09 -6.93e-13 1955.3 524281
+50.750
```

passwd

This command is used to change the password for the user that you are logged in as. It affects the serial port, SSH, Telnet and HTTPS. **`passwd`** is a Linux command that is also described in ***Appendix C - Helpful Linux Information***.

Command: **`passwd`**
Ninja reply: Interactive script is started.

rcvrserialnumber

This command shows the serial number of the GPS Receiver in the Ninja.

Command: **`rcvrserialnumber`**
Ninja reply: **`15080056`**

rcvrstat

This command shows three critical status parameters of the GPS Receiver: the number of satellites currently being tracked, the automatic gain control DAC value for the receiver front end, and the average Carrier-to-Noise ratio of the tracked satellites.

Command: **`rcvrstat`**
Ninja reply: **`20150622.23:35:50 8 125 45.0`**

resetlastgpswn

This command is for use with a GPS simulator Refer to ***Appendix I - Operation with a GPS Simulator*** for more information.

resetleaphistory

This command is for use with a GPS simulator   Refer to ***Appendix I - Operation with a GPS Simulator*** for more information.

rticmode (Optional)

This command is only available if the RTIC option has been installed. Refer to ***Chapter 12 - Real-Time Ionospheric Corrections*** for more information.

setantfltmask

This command allows you to enable or mask the GPS antenna fault. Parameter for this command is either MASKED or ENABLED. Setting this command to MASKED will prevent the antenna fault from creating an alarm condition. Some installations may need to mask this fault due to special antenna situations like splitters or DC blocks that confuse the antenna detection circuit. The factory default setting is ENABLED.

```
Command:      setantfltmask MASKED
Ninja reply:  Antenna Fault Mask set to MASKED
```

setcaldelay

This command starts an interactive utility that allows you to change the clock calibration delay. This setting is used to advance or retard the clock in order to compensate for antenna cable length or other external hardware or cabling.   See ***Appendix E - Installing the GPS Antenna, Calibrate Your Receiver*** for more details.

```
Command:      setcaldelay
Ninja reply:  Interactive utility is started.
```

setgpsdynmode

This command accepts a single argument: ON or OFF, to allow you to set the dynamic mode of operation of the GPS Subsystem. By default, the unit is configured for static operation, so this setting is OFF. It is important that the dynamic mode be set OFF when the instrument is in a static installation.

If Ninja will be mounted on a moving platform then this setting must be changed to ON. The change takes place immediately and is stored non-volatilely. Dynamic mode is intended for shipboard applications only.

```
Command:      setgpsdynmode ON
Ninja reply:  GPS Dynamic Mode is ON
```

setgpsrefpos

This command starts an interactive utility that allows you to set the accurate reference position of Ninja. This utility must be run as the root user. By default, the unit is configured to locate itself using the GPS satellites. In some situations, visibility of the sky is limited and Ninja will not be able to determine its position. In this case, you must determine an accurate WGS-84 position by other means and input it using this command. Changes you make to the position take place immediately. Refer to ***Appendix E - Installing the GPS Antenna, GPS Reference Position*** for details. *If the GPS dynamic mode setting is ON (see* **`gpsdynmode`**/**`setgpsdynmode`** *commands), then running this utility will have no effect.*

In addition to setting a new reference position, you can also invalidate an existing one. We recommend you do this if Ninja has an established position and then you move your GPS antenna. You can invalidate an old position by setting the position mode to UNKNOWN. This will speed up the time it takes for Ninja to acquire a new position and relock to the GPS signal. A cold start in unknown position mode should take about 20 minutes to lock, assuming a decent antenna installation.

Command: **`setgpsrefpos`**
Ninja reply: Interactive utility is started.

setinhibitoutputsmode

This command allows you to set the inhibit mode of the timing outputs. Timing outputs are pulse rates and time codes. If it is set to ON, then all timing outputs will be inhibited (not present) until the system is locked to the GPS signal. If it is set to OFF, then all timing outputs will be present at all times. This command requires one argument: ON or OFF.

Command: **`setinhibitoutputsmode ON`**
Ninja reply: **`Inhibit Outputs Mode is ON`**

setrticmode (Optional)

This command is only available if the RTIC option has been installed. Refer to ***Chapter 12 - Real-Time Ionospheric Corrections*** for more information.

setsigfltmask

This command allows you to enable or mask the Signal Loss Fault. Parameter for this command is either MASKED or ENABLED. Setting this command to MASKED will prevent a signal loss fault from creating an alarm condition. Some installations may need to mask this fault when operating the NTP server as a Stratum 2 server. The factory default setting is ENABLED.

Command: **`setsigfltmask MASKED`**
Ninja reply: **`Signal Loss Fault Mask set to MASKED`**

settfomfltlvl

This command allows you to change the TFOM Fault Level. This is the threshold at which a signal loss fault will be asserted. See ***Appendix A - Time Figure of Merit*** for more information. By changing the TFOM Fault Level you control the point at which the time error will produce a signal loss fault, which then creates an alarm condition. The factory default setting is 9, which is the maximum TFOM value.

Command: **`settfomfltlvl 6`**
Ninja reply: **`TFOM Fault Level set to 6`**

sigfltmask

This command displays the current setting for the Signal Loss Fault Mask.

```
Command:      sigfltmask
Ninja reply:  Signal Loss Fault is ENABLED
```

syskernel

This command returns the currently booted linux kernel, either 0 or 1, where 0 is the factory-installed kernel and 1 is the upgraded kernel.

```
Command:      syskernel
Ninja reply:  BOOTED KERNEL IMAGE = 1 (Upgrade)
```

sysosctype

This command displays the installed system oscillator type. It is one of HP-TCXO, MS-OCXO, HS-OCXO or US-OCXO.

```
Command:      sysosctype
Ninja reply:  Installed Oscillator is MS-OCXO.
```

sysrootfs

This command returns the currently loaded linux root file system, either 0 or 1, where 0 is the factory-installed root file system and 1 is the upgraded root file system.

```
Command:      sysrootfs
Ninja reply:  BOOTED ROOT FILE SYSTEM IMAGE = 1 (Upgrade)
```

sysstat

This command allows you to query the status of the NTP Subsystem. It retrieves information from the NTP daemon to determine the current synchronization status of the NTP Subsystem. It then retrieves the last line in the logfile */var/log/praecis0.monitor* controlled by the NTP daemon reference clock driver that communicates with the GPS Subsystem. This logfile is updated every 16 seconds under normal operation. It parses and formats the data contained therein and prints this fixed-length (generally, since grossly unsynchronized states could cause the floating offset field to overflow momentarily) string having these fields:

```
LKSTAT TO GPS, Offset = +S.ssssssss, LI = ??, TFOM = ? @ YEAR DOY HH:MM:SS LS
```

Where:

LKSTAT is the system peer status of the NTP daemon relative to the GPS Subsystem, either LOCKED or NOTLKD. NOTLKD can imply several things: the system has just started, there is a fault in the GPS Subsystem which has caused NTP to either be unable to obtain timing information from the GPS Subsystem or to reject the timing information that it is obtaining from it.

+S.sssssssss is the offset in seconds between the NTP system clock and the GPS Subsystem clock. Positive implies that the system clock is ahead of the GPS Subsystem.

LI = ?? is the NTP daemon leap indicator bits. Leap seconds typically occur every 1 - 3 years. Possible indicator values are:

00:	Normal, locked operation.
01:	Leap second insertion event will occur after 23:59:59 UTC.
10:	Leap second deletion event will occur after 23:59:58 UTC.
11:	Fault. Unsynchronized state.

TFOM = ? is a value between 3 and 9 and indicates clock accuracy.
A detailed explanation of TFOM is in ***Appendix A - TFOM.***.

YEAR is the year of the UTC timestamp of the most recent update received from the GPS Sub system.

DOY is the day-of-year of the UTC timestamp of the most recent update received from the GPS Subsystem.

HH:MM:SS is the hour, minute and second ot the UTC timestamp of the most recent update received from the GPS Subsystem.

LS is the current number of leap seconds difference between the UTC and GPS timescales (18 at the time of this writing).

Below is an example of a typical response to this command:

Command: **sysstat**
Ninja reply:

```
LOCKED TO GPS, Offset = +0.000000024, LI = 00, TFOM = 4 @ 2013 012 06:03:10 18
```

systemio

This command returns the current settings for all installed, system-wide I/O signals that may be routed to the SMA connectors. These signals are the 1 PPS and Time Code outputs. See ***Chapter 9 - Inputs/Outputs*** for information on the various options.

Command: **systemio**
Ninja reply:

```
System I/O Signal 1 PPS OUTPUT is Installed --
   Current Setting = 1 Milliseconds Pulse Width

System I/O Signal TIME CODE OUTPUT is Installed --
   Current Setting = IRIG-B122/B002 Format
```

systemioconfig

This command is an interactive utility that allows the root user to modify the settings for all installed, system-wide I/O signals. See ***Chapter 9 - Inputs/Outputs*** for information on the various options and modules.

Command: `systemioconfig`
Ninja reply: Interactive shell script is started.

systimemode

This command displays the current time mode for the optional Time Code Output. Time modes are UTC or GPS.

Command: `systimemode`
Ninja reply: `Time Mode = UTC`

systimemodeconfig

This command starts an interactive utility that allows you to configure the time mode of the optional Time Code Output. *These settings have no effect on the operation of the NTP daemon or the underlying Linux operating system time. These ALWAYS operate in UTC.*

By default, the unit is configured to operate in UTC mode. If you need to modify the setting, you must run this utility as root. Settings made using this command are non-volatile.

Command: `systimemodeconfig`
Ninja reply: Interactive utility is started.

sysversion

This command displays the firmware version and build date of the Linux root file system.

Command: `sysversion`
Ninja reply: `NinjaPTM 6010-0086-000 v 1.00 - Mar 6 02:16:53 2020`

tfomfltlvl

This command displays the current setting for the TFOM fault level.

Command: `tfomfltlvl`
Ninja reply: `TFOM fault level set to 6`

triggerppo (Optional)

This command is only available if one or more Programmable Pulse Output (PPO) options have been installed. Refer to ***Chapter 9 - Inputs/Outputs (I/O), Programmable Pulse Output (PPO) Option*** for more information.

updatekernelflag

This command allows you to update the configuration of the Linux bootloader after a new kernel image has been written to the UPGRADE kernel partition of Ninja FLASH disk. You may also use it to reset the default back to the FACTORY kernel partition. Refer to ***Appendix B - Upgrading the Firm-***

ware, Performing the Linux Kernel Upgrade for detailed instructions for performing the upgrade procedure. One argument is accepted, whose value is either 0 or 1, which causes a flag to be set that indicates to the bootloader which kernel image should be loaded by default. If an argument value of 2 is given, then the currently configured default kernel is shown.

```
Command:       updatekernelflag 1
Ninja reply:   Default Kernel now set to: UPGRADE
Command:       updatekernelflag 2
Ninja reply:   Default Kernel = UPGRADE
```

updaterootflag

This command allows you to update the configuration of the Linux bootloader after a new root file system image has been written to the UPGRADE root file system partition of Ninja FLASH disk. You may also use it to reset the default back to the FACTORY root file system partition. Refer to ***Appendix B - Upgrading the Firmware*** for detailed instructions for performing the upgrade procedure. One argument is accepted, whose value is either 0 or 1, which causes a flag to be set that indicates to the bootloader which root file system image should be loaded by default. If an argument value of 2 is given, then the currently configured default root file system is shown.

```
Command:       updaterootflag 1
Ninja reply:   Default Root File System now set to: UPGRADE
Command:       updaterootflag 2
Ninja reply:   Default Root File System = UPGRADE
```

upgradekernel

This utility allows you to upgrade the Linux Kernel. It is run after the *kernel.gz* file has been copied to the */tmp* directory on the system. It performs an erase of the upgrade kernel partition and then writes the */tmp/kernel.gz* file to it. Refer to ***Appendix B - Upgrading the Firmware, Performing the Linux Kernel Upgrade*** for detailed information.

```
Command:       upgradekernel
Ninja reply:   Shows progress indicator.
```

upgradercvr

This utility allows you to upgrade the GPS Receiver firmware. Prior to executing this command, you must copy the new binary firmware file to */tmp/rcvr.bin*.

The utility starts the X-modem file transfer, and then displays progress to the console. See ***Appendix B - Upgrading the Firmware, Performing the GPS Receiver Upgrade*** for more information.

```
Command:       upgradercvr
Ninja reply:   Upgrade progress is shown.
```

upgradercvrfpga

This utility allows you to upgrade the Field-Programmable Gate Array (FPGA) resident on the GPS Receiver. Prior to executing this command, you must copy the new binary file to */tmp/rcvrfpga.rbf*.

The utility starts the X-modem file transfer, and then displays progress to the console. See ***Appendix B - Upgrading the Firmware, Performing the GPS Receiver FPGA Upgrade*** for more information.

Command:	`upgradercvrfpga`
Ninja reply:	Upgrade progress is shown.

upgraderootfs

This utility allows you to upgrade the Linux Root File System. It is run after the *rootfs.gz* file has been copied to the */home* directory on the system. It performs an erase of the upgrade root file system partition and then writes the */home/rootfs.gz* file to it. Refer to ***Appendix B - Upgrading the Firmware, Performing the Linux RFS Upgrade*** for detailed information..

Command:	`upgraderootfs`
Ninja reply:	Shows progress indicator.

This page intentionally left blank.

Chapter *Four*

Hyper Text Transport Protocol (HTTP/HTTPS)

This chapter briefly describes the web interface that resides on the Ninja. This interface is a fast and easy-to-use graphical interface that is compatible with your standard web browser. Simply point your browser to the IP address of Ninja and log in securely with HTTPS. Security-conscious customers may disable this interface entirely (see Chapter 5 for instructions).

NOTE

When Ninja is shipped from the factory, HTTP/HTTPS is disabled. If you want to enable, see ***Chapter 5 - Security, Enable/Disable Protocols***. For security reasons, we recommend that you configure for HTTPS only. See ***Configure HTTPS only for IPv4*** and ***Configure HTTPS only for IPv6*** in this chapter. We also recommend that you restrict access to specific IP addresses. See ***Restrict Access*** in this chapter.

The HTTPS implementation uses HTTP encrypted via Transport Layer Security (TLS). HTTPS enhances security because it encrypts and decrypts the requested and returned pages from the server, including any passwords which are transmitted. The HTTPS implementation is built from the standard Hiawatha distribution:

hiawatha-webserver.org

It uses HTTPS (HTTP via TLS) with MbedTLS (formerly known as PolarSSL). For more information about this protocol, refer to:

tls.mbed.org

See later in this chapter for information on changing the default HTTP/HTTPS configuration and TLS certificate and key. To disable the HTTP/HTTPS protocol, see ***Chapter 5 - Security, Disable SNMP, SSH and HTTPS***.

Interface Description

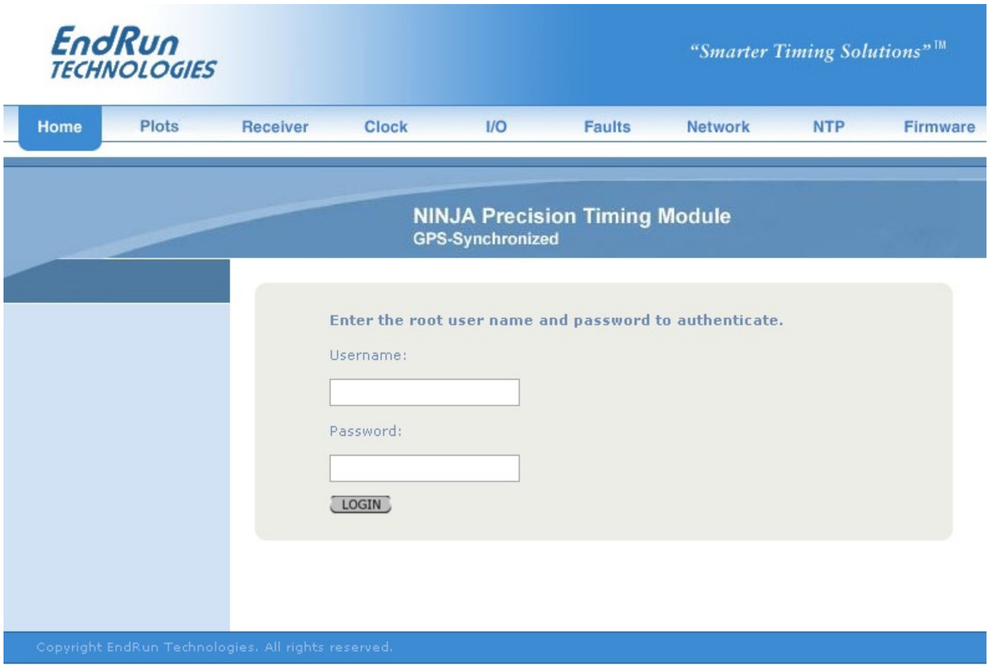
For security reasons the web pages on the Ninja show status and configuration information only. You cannot change any operational settings. To make changes to the Ninja you will need to use the command line interface via either a network or serial port.

To get started with the web interface simply point your browser to the IP address of the Ninja and log in. By default, IPv4 will accept a request for HTTP or HTTPS. The server needs additional configuration for HTTPS only and IPv6. Following are example address formats for IPv4 and IPv6:

IPv4: 192.168.1.1
IPv6: [fe80:0:0:0:20e:f3ff:fe01:1f] Do not forget the brackets [].

With HTTPS, a warning dialog page will be presented for the certificate. Acknowledge the dialog page and the server will continue to load, protected by TLS. The browser URL should change from http: to https:, indicating that the page is protected by TLS. To maximize security you should replace the TLS Certificate. See ***Security, Configure Certificate and Key*** later in this chapter for details.

Below is a picture of the login page:



NINJA Precision Timing Module
GPS-Synchronized

Enter the root user name and password to authenticate.

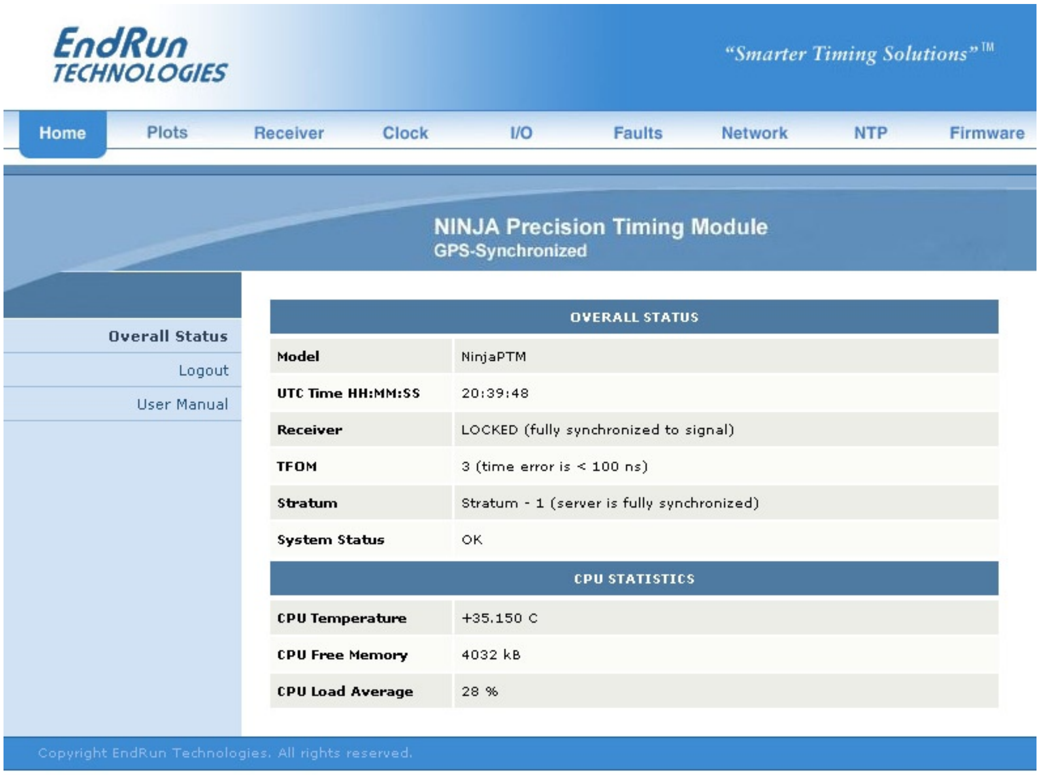
Username:

Password:

Copyright EndRun Technologies. All rights reserved.

Navigation

The main menu tabs across the top of each webpage allow you to navigate through the status information. These tabs are: Home, Plots, Receiver, Clock, I/O, Faults, Network, NTP and Firmware.



NINJA Precision Timing Module
GPS-Synchronized

OVERALL STATUS	
Model	NinjaPTM
UTC Time HH:MM:SS	20:39:48
Receiver	LOCKED (fully synchronized to signal)
TFOM	3 (time error is < 100 ns)
Stratum	Stratum - 1 (server is fully synchronized)
System Status	OK

CPU STATISTICS	
CPU Temperature	+35.150 C
CPU Free Memory	4032 kB
CPU Load Average	28 %

Copyright EndRun Technologies. All rights reserved.

Configure HTTPS

HTTP/HTTPS use files for the default configuration located in */etc/hiawatha*. Of these, you will typically only need to modify *hiawatha.conf*. Advanced users who need to modify the default configuration will need to edit the file and copy it to the */boot/etc/hiawatha* directory. Do not attempt to change the directives unless you have a real need to do so. (See ***Appendix C - Helpful Linux Information, Text Editors***.)

To configure HTTPS (HTTP encrypted via TLS), you will need to modify *hiawatha.conf*. When configured, a HTTP request will be redirected to HTTPS. You must edit the */etc/hiawatha/hiawatha.conf* file and configure as shown below.

Configure HTTPS only for IPv4

You must edit the */etc/hiawatha/hiawatha.conf* file and set the Hostname value in the VirtualHost configuration block to the IPv4 (global) address of the server.

After making and saving your changes, you must copy the edited file to the non-volatile FLASH area and reboot the unit:

```
cp -p /etc/hiawatha/hiawatha.conf /boot/etc/hiawatha
reboot
```

Configure HTTP for IPv6

You must edit the */etc/hiawatha/hiawatha.conf* file and set the Interface value in the Binding configuration block for Port 80. (A commented example is present in the default file.)

After making and saving your changes, you must copy the edited file to the non-volatile FLASH area and reboot the unit:

```
cp -p /etc/hiawatha/hiawatha.conf /boot/etc/hiawatha
reboot
```

Configure HTTPS only for IPv6

You must edit the */etc/hiawatha/hiawatha.conf* file and:

1. Set the Interface value in the Binding configuration block for Port 80.
2. Set the Interface value in the Binding configuration block for Port 443.
3. Set the Hostname value in the VirtualHost configuration block to the IPv6 (global) address of the server.

After making and saving your changes, you must copy the edited file to the non-volatile FLASH area and reboot the unit:

```
cp -p /etc/hiawatha/hiawatha.conf /boot/etc/hiawatha
reboot
```

Security

Restricting access to specific hosts and configuring the certificate and key are described below. For information on disabling the HTTP/HTTPS protocol see ***Chapter 5 - Security, Disable SNMP, SSH and HTTPS***.

Restrict Access

To control access via HTTP/HTTPS, you must edit the */etc/hiawatha/hiawatha.conf* file and set the AccessList values in the VirtualHost configuration block to define which IPs have access to the VirtualHost. '**allow**' gives access. '**deny**' denies access.

The default file contains this example that must be edited and the '**#**" character removed:

```
#Access List allow 192.168.1.1, deny all
```

After making and saving your changes, you must copy the edited file to the non-volatile FLASH area and reboot the unit:

```
cp -p /etc/hiawatha/hiawatha.conf /boot/etc/hiawatha
reboot
```

Configure Certificate and Key

For TLS it is recommended, but not required, that new certificates and keys are generated and installed on the Hiawatha web server. The factory-configured, self-signed certificate is located in */etc/hiawatha/tls*. After creating new certificates, they will need to be saved in */boot/etc/hiawatha/tls/hiawatha.pem*. To generate a new certificate and key, issue these commands:

```
cd /boot/etc/hiawatha/tls
openssl req -new > cert.csr
openssl rsa -in privkey.pem -out key.pem
openssl x509 -in cert.csr -out cert.pem -req -signkey key.pem -days 1001
cat key.pem cert.pem > hiawatha.pem
```

The file will be created in the */boot/etc/hiawatha/tls* directory. You must reboot the Ninja for changes to take effect.

NOTE

If you request your X.509 SSL/TLS certificate from a Certificate-Signing Authority (CA) you <u>must</u> have the following order in the ***hiawatha.pem*** file as shown below:

-----BEGIN RSA PRIVATE KEY-------
[webserver private key]
-----END RSA PRIVATE KEY-----

-----BEGIN CERTIFICATE-----
[webserver certificate received from CA]
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
[optional intermediate CA certificate]
-----END CERTIFICATE-----

This page intentionally left blank.

Chapter *Five*

Security

Your Ninja incorporates several important security features to prevent unauthorized tampering with its operation. Many of these are standard multiple-user access control features of the underlying Linux operating system which controls the Ninja. Others are provided by the additional protocol servers selected for inclusion in your Ninja, and the way that they are configured.

Secure user authentication and session privacy while performing routine monitoring and maintenance tasks are provided by the OpenSSH implementations of the “secure shell” daemon, **`sshd`** and its companion “secure copy” utility, **`scp`**. The Hiawatha implementation of the Hyper Text Transfer Protocol (HTTPS) with Secure Sockets Layer (SSL) daemon (**`hiawatha`**) provides for a secure, encrypted session with a digital certificate. The NET-SNMP implementation of the Simple Network Management Protocol (SNMP) daemon, **`snmpd`** conforms to the latest Internet standard, known as SNMPv3, which also supports secure user authentication and session privacy. In addition, the Network Time Protocol daemon, **`ntpd`** supports client-server authentication security measures to deter spoofing of NTP clients by rogue NTP servers. This chapter describes these security measures and gives the advanced network administrator information that will allow custom configuration to fit specific security needs.

DEFAULTS

As shipped from the factory, SSH and SNMP are both enabled with default passwords and all hosts allowed. To ensure security, change the passwords or disable the protocols. To change the password for SSH use the **`passwd`** command. To change the passwords/community strings for SNMP see ***Chapter 6 - SNMP, Change Default Community Strings (Passwords)***. To restrict access to specific hosts see ***Restrict Access - Telnet, SSH and SNMP*** in this chapter.

For security-conscious users, risky protocols such as HTTP/HTTPS, Telnet, Time and Daytime are disabled by default. To enable, see ***Enable/Disable Protocols*** in this chapter. If you enable Telnet or HTTP/HTTPS then be sure to change the default passwords by using the **`passwd`** command.

By default all hosts are allowed access to Telnet. To restrict access to specific hosts, see ***Restrict Access - Telnet, SSH and SNMP*** in this chapter. All hosts are allowed access via HTTPS as well. To restrict access via HTTPS, see ***Chapter 4 - HTTP/HTTPS, Security, Restrict Access***.

To completely disable any or all of these protocols see ***Enable/Disable Protocols*** below.

Linux Operating System

The Linux operating system versions are shown in ***Appendix J - Specifications***. Linux supports a complete set of security provisions:

- System passwords are kept in an encrypted file, */etc/shadow* which is not accessible by users other than *root*.

- Direct *root* logins are only permitted on the local RS-232 console or via SSH.

- The secure copy utility, **`scp`**, eliminates the need to use the insecure FTP protocol for transferring program updates to Ninja.

- Access via HTTP/HTTPS may be restricted or completely disabled. See ***Disable SNMP, SSH and HTTPS*** in this chapter or ***Chapter 4 - HTTP/HTTPS, Restrict Access***.

- SNMP access for system monitoring only, is configurable to provide the security of the latest version 3 Internet standard which supports both view-based access control and user-based security using modern encryption techniques. Previous versions v1 and v2c supported access control essentially via passwords transmitted over the network in plain text. Refer to ***Chapter 6 - SNMP*** and ***Restrict Access - Telnet, SSH and SNMP*** (below) for details. SNMP may also be completely disabled. See ***Disable SNMP, SSH and HTTP/HTTPS*** below.

- Individual host access to protocol server daemons **`in.telnetd,`** **`snmpd`** or **`sshd`** are controlled by directives contained in the files */etc/hosts.allow* and */etc/hosts.deny*, which are configured using the interactive script **`accessconfig.`** See ***Restrict Access - Telnet, SSH and SNMP*** below.

- Insecure protocols like Time, Daytime and Telnet are disabled by default. They may be enabled by configuration of the **`inetd`** super-server daemon using the interactive script **`inetdconfig`**. See ***Enable/Disable Telnet, Time and Daytime*** below.

Restrict Access

The following paragraphs describe how to restrict SNMP, SSH, Telnet and HTTPS access to specific hosts. To restrict access to HTTPS, see ***Chapter 4 - HTTP/HTTPS, Security, Restrict Access.*** Also, to restrict NTP query access, see ***Chapter 7 - NTP, Restrict NTP Query Access***.

Restrict Access - Telnet, SSH and SNMP

By default, Ninja is configured to allow access by all users via Telnet, SSH and SNMP. To ensure security and to protect against denial-of-service attacks, you should restrict access by using the **`accessconfig`** command.

`accessconfig` modifies two files, */etc/hosts.allow* and */etc/hosts.deny*, which are used by **`tcpd`** and the standalone daemons, **`snmpd`** and **`sshd`**, to determine whether or not to grant access to a requesting host. These two files may contain configuration information for a number of protocol servers, but in Ninja only access control to the protocol server daemons **`in.telnetd, sshd`** and **`snmpd`** is configured.

As shipped from the factory, these two files are empty. When you run **accessconfig**, these lines are added to the */etc/hosts.deny* file:

```
in.telnetd:  ALL
sshd:  ALL
snmpd:  ALL
```

This tells **tcpd** to deny access to **in.telnetd, sshd** and **snmpd** to all hosts not listed in the */etc/hosts.allow* file. The **snmpd** and **sshd** daemons also parse this file directly prior to granting access to a requesting host.

Next you will be prompted to enter a list of hosts that will be granted access to **in.telnetd, sshd** and **snmpd**. These appear in the */etc/hosts.allow* as lines like this:

```
in.telnetd:  192.168.1.2, 192.168.1.3
sshd:  192.168.1.2, 192.168.1.3
snmpd:  192.168.1.2, 192.168.1.3
```

This simple shell script handles the needs of most users, however the syntax of these two files supports elaborate configuration possibilities which are beyond the capabilites of this simple shell script. Advanced users who need these capabilities will need to edit these two files directly and then copy them to the */boot/etc* directory. (See ***Appendix C - Helpful Linux Information, Using Editors***.) Be careful to maintain the proper ownership and access permissions by using **cp -p** when copying the files.

Enable/Disable Protocols

As shipped from the factory, SSH and SNMP are enabled. Telnet, Time, Daytime and HTTP/HTTPS are all disabled. See below for instructions on how to enable and/or disable these protocols. The Network Time Protocol (NTP) cannot be disabled.

Enable/Disable Telnet, Time and Daytime

To enable or disable Telnet, Time and Daytime use the **inetdconfig** command to start an interactive script that will ask you which protocols to enable. Then it will modify the */etc/inetd.conf* file, which is read by the super-server daemon, **inetd**. Requests from remote hosts for protocols not configured in */etc/inetd.conf* will be refused. Currently, three servers are configurable via **inetdconfig**: Time and Daytime (whose protocol servers are contained within the **inetd** daemon itself), and **in.telnetd**. Any one or all of these may be enabled or disabled for start-up.

Disable SNMP, SSH and HTTP/HTTPS

To disable SNMP, SSH or HTTP/HTTPS, you only have to modify the file mode of the scripts that control their execution. These are located in the */etc/rc.d* directory. To disable any of these daemons, issue one or more of these commands:

```
chmod -x /etc/rc.d/rc.snmpd
chmod -x /etc/rc.d/rc.sshd
chmod -x /etc/rc.d/rc.hiawatha
```

After issuing these commands, you must copy the modified file(s) to the non-volatile FLASH area using one or more of these commands:

```
cp -p /etc/rc.d/rc.snmpd /boot/etc/rc.d
cp -p /etc/rc.d/rc.sshd /boot/etc/rc.d
cp -p /etc/rc.d/rc.hiawatha /boot/etc/rc.d
```

Reboot Ninja when done for the changes to take effect.

IMPORTANT

After modifying ***/etc/rc.d/rc.snmpd, rc.sshd or rc.hiawatha***, you must copy them to the ***/boot/etc/rc.d*** directory and reboot the system. It is very important to use the **`-p`** when performing the copy. During the boot process, the files contained in the ***/boot/etc/rc.d*** directory are copied to the working ***/etc/rc.d*** directory on the system RAM disk. In this way the factory defaults are overwritten.

Enable SNMP, SSH and HTTP/HTTPS

Since the factory default is for SNMP and SSH to be enabled, if you want to re-enable SNMP or SSH after previously disabling one or both of them, then all you need to do is remove the previously copied *rc* file from the */boot/etc/rc.d* directory using one or more of these commands:

```
rm /boot/etc/rc.d/rc.snmpd
rm /boot/etc/rc.d/rc.sshd
```

Since the factory default is for HTTP/HTTPS to be disabled, you will need to execute these commands to enable it:

```
chmod +x /etc/rc.d/rc.hiawatha
cp -p /etc/rc.d/rc.hiawatha /boot/etc/rc.d
```

Reboot Ninja when done for the changes to take effect.

Is the Protocol Disabled?

Telnet, TIME and DAYTIME: To determine if one of these protocols is disabled, use the **`inetdconfig`** command.

SNMP, SSH and HTTP/HTTPS: To determine if one of these protocols is disabled, issue the following command:

```
  ls -l /boot/etc/rc.d
```

If you see one of the following files listed, and there is NOT an '*' after the file name, then the corresponding protocol is disabled. Also, if you do not see the *rc.hiawatha* file, it is disabled.:

```
-rw-r--r-- 1 root root 1144 Feb 19 01:52 rc.hiawatha
-rw-r--r-- 1 root root 1168 Oct 26 2012  rc.snmpd
-rw-r--r-- 1 root root 2684 Feb 18 02:16 rc.sshd
```

If *rc.snmpd* or *rc.sshd* is not listed, or it is listed and there is an '*' after the file name, then the protocol is enabled. For HTTP/HTTPS to be enabled, *rc.hiawatha* must be present with an '*'. Here is an example:

```
-rwxr-xr-x 1 root root 1168 Oct 26 2012  rc.hiawatha*
```

OpenSSH

The secure shell protocol server running in Ninja is based on the portable OpenSSH for Linux. As such it supports both SSH1 and SSH2 protocol versions. By default, only SSH2 is enabled in Ninja due to security issues with SSH1. For more information about OpenSSH, and to obtain client software, refer to the OpenSSH website:

openssh.com

An excellent book which describes operation and configuration of the various SSH implementations, including OpenSSH is:

SSH, The Secure Shell, Barrett & Silverman, O'Reilley & Associates, 2001.

NOTE: To disable the SSH protocol see ***Disable SNMP, SSH and HTTP/HTTPS*** above. To restrict access see ***Restrict Access - Telnet, SSH and SNMP*** above.

Configure Keys

On initial boot-up from out-of-the-box, the SSH start-up script, */etc/rc.d/rc.sshd*, will detect that no keys are present in the */etc/ssh* directory. It will call **ssh-keygen** to generate a set of host keys and then it will copy them to the */boot/etc/ssh* directory. These will be copied to */etc/ssh* during each boot up. A complete set of security keys for both SSH1 and SSH2 versions of the protocol are generated. RSA keys are supported by both versions, and DSA keys are supported when using the SSH2 version. Should you need to replace your keys at any time, you can just remove the keys from the */boot/etc/ssh* directory and then reboot Ninja. A new set of host keys will automatically be generated.

To configure root logins to your Ninja via passwordless, public key authentication, you must generate a public/private pair of SSH2 keys using your own ssh key generating utility, or you can use the **ssh-keygen** that is resident on the Ninja file system. You must then append the public key to the */boot/root/.ssh/authorized_keys2* file in the non-volatile FLASH area on your Ninja. At boot time, Ninja will copy these to the actual working */root/.ssh* directory of the system ramdisk. To use this capability, the corresponding private key must reside in the */root/.ssh* directory of your remote computer as *id_rsa* or *id_dsa*. If you are unfamilar with this process, refer to the man page for the **ssh-keygen** utility for details (issue **man ssh-keygen** at the prompt). (Be careful to maintain the proper ownership and access permissions of the private key by using **cp -p** when copying the file. It MUST be readable only by *root*.)

Advanced users wishing to modify the overall configuration of the **sshd** daemon should edit the */etc/ssh/sshd_config* file and then copy it to the */boot/etc/ssh* directory of Ninja. Be careful to maintain the proper ownership and access permissions by using `cp -p` when copying the file. At boot time, it will be copied to the */etc/ssh* directory of the system ramdisk, thereby replacing the factory default configuration file.

HTTP/HTTPS

For information on how to configure a certificate and key, see ***Chapter 4 - HTTP/HTTPS, Security, Configure Certificate and Key***. To restrict access to HTTP/HTTPS, see ***Chapter 4 - HTTP/HTTPS, Security, Restrict Access.***

NTP

You can configure your NTP clients for secure MD5 authentication. See ***Chapter 7 - NTP, Unix-like Platforms: MD5 Authenticated NTP Client Setup*** or ***Chapter 7 - NTP, Windows: MD5 Authenticated NTP Client Setup***. You can also restrict query access. See ***Restrict NTP Query Access*** in ***Chapter 7***.

Network Security Vulnerabilities

EndRun addresses major network security vulnerabilities that affect Ninja on this webpage:

endruntechnologies.com/support/product-bulletins

This Application Note describes best practices to secure your time server and mitigate many network security vulnerabilities:

endruntechnologies.com/pdf/Time-Server-Security-Best-Practices.pdf

Chapter *Six*

Simple Network Management Protocol (SNMP)

Your Ninja includes the NET-SNMP version 5.5.1 implementation of an SNMP agent, **`snmpd`**, *and a SNMP notification/trap generation utility,* **`snmptrap`**. *It supports all versions of the protocol in use today: SNMPv1 (the original Internet standard), SNMPv2c (never reached standard status, often called "community SNMP") and SNMPv3 (the latest Internet standard).*

The NET-SNMP project has its roots in the Carnegie-Mellon University SNMP implementation. For more detailed information about the NET-SNMP project and to obtain management software and detailed configuration information, you can visit this website:

net-snmp.org

An excellent book which describes operation and configuration of various SNMP managers and agents, including the NET-SNMP implementations, is available from O'Reilley & Associates:

Essential SNMP, Mauro & Schmidt, O'Reilley & Associates, 2001

If you are planning to operate with SNMPv3, it is highly recommended that you make use of both of these resources to familiarize yourself with the agent configuration concepts.

SNMPv3 Security

Prior to SNMPv3, SNMP had definite security inadequacies due to using two community names in a manner analogous to passwords that were transmitted over the network as clear text. In addition, since no mechanism existed for authenticating or encrypting session data, any number of man-in-the-middle data corruption/replacement exploits were possible in addition to plain old snooping to learn the community names. SNMPv3 implements the User-based Security Model (USM) defined in RFC-2274 which employs modern cryptographic technologies to both authenticate multiple users and to encrypt their session data for privacy, much in the same way that SSH does for remote login shell users.

In addition, it implements the View-based Access Control Model (VACM) defined in RFC-2275. This RFC defines mechanisms for limiting the access of multiple users having various security levels (no authentication, authentication or authentication plus privacy) to specific "views" of the Structure of Management Information (SMI) object tree.

Enterprise Management Information Base (MIB)

In addition to providing the SNMP variables contained in MIB-II as described in RFC-1213, EndRun Technologies has implemented an enterprise MIB using the syntax of the SMI version 2 (SMIv2) as described in RFC-2578:

NINJA-MIB

Which is located on your Ninja in this ASCII file:

/usr/local/share/snmp/mibs/NINJA-MIB.txt

In addition to a complete set of NTP and Receiver (GPS) status objects, the MIB defines four SMIv2 notification objects:

- NTP Leap Indicator Bits status change
- NTP Stratum change
- Receiver Fault Status change
- Receiver Time Figure of Merit change

Invocation of the SNMP daemon

The SNMP daemon, **`snmpd`** is started from the */etc/rc.d/rc.snmpd* system start-up script. By default, it will listen on port 161 for SNMP queries from the network management system. If you would like to have it listen on another port, you could edit the file and change the port number in the argument list being passed to **`snmpd`** when it is started.

IMPORTANT

After modifying ***/etc/rc.d/rc.snmpd***, you must copy it to the ***/boot/etc/rc.d*** directory and reboot the system. It is very important to retain the access mode for the file, so be sure to use **`cp -p`** when performing the copy. During the boot process, the files contained in the ***/boot/etc/rc.d*** directory are copied to the working ***/etc/rc.d*** directory on the system RAM disk. In this way the factory defaults are overwritten.

Quick Start Configuration -- SNMPv1/v2c

You should be able to compile the NINJA-MIB file on your SNMP management system and access the variables defined therein. The factory default community names are “NINJAGPS_0” for the read-only community and “endrun_1” for the read-write community. This is all that is required for operation under v1 and v2c of SNMP.

Change Default Community Strings (Passwords)

You can, and should, change the default community names by editing */etc/snmpd.conf* and modifying these two lines:

```
rwcommunity  endrun_1
rocommunity  NINJAGPS_0
```

Configuring SNMPv1 Trap Generation

To have your Ninja send SNMPv1 traps (RFC-1215) you must configure the community and destination for SNMPv1 traps by uncommenting and editing this line in */etc/snmpd.conf*:

```
trapsink    xxx.xxx.xxx.xxx trapcommunity trapport
```

where **trapcommunity** should be replaced by your community, and **xxx.xxx.xxx.xxx** is the IP address or hostname of the destination host for receiving the traps generated by Ninja. By default, the trap will be sent to port 162. You may optionally add another parameter, **trapport** to the end of the above line to override the default port setting. Otherwise leave it blank.

Note: Though the agent will recognize multiple **trapsink** lines within */etc/snmpd.conf* and send the generic SNMP coldStart or authenticationFailure traps to multiple destinations, the Ninja enterprise MIB trap generation mechanism will only send a trap to the last three declared **trapsink** in the file.

Configuring SNMPv2c Notifications and Informs

To have your Ninja send SNMPv2c notifications (SMIv2, RFC-2578) or informs, you must configure the communities and destinations by uncommenting and editing one or both of these lines in */etc/snmpd.conf*:

```
trap2sink    xxx.xxx.xxx.xxx trap2community trap2port
informsink   xxx.xxx.xxx.xxx informcommunity informport
```

where **trap2community** and **informcommunity** should be replaced by your communities, and **xxx.xxx.xxx.xxx** is the IP address or hostname of the destination host for receiving the notifications or informs generated by Ninja. By default, the v2c trap or inform will be sent to port 162. You may optionally add another parameter, **trap2port** or **informport** to the ends of the above lines to override the default port setting. Otherwise leave it blank.

Note: Though the **snmpd** agent will recognize multiple **trap2sink** or **informsink** lines within */etc/snmpd.conf* and send the generic SNMP coldStart or authenticationFailure notifications and informs to multiple destinations, the Ninja enterprise MIB notification/inform generation mechanism will only send a notification to the last three declared **trap2sink,** and an inform to the last three declared **informsink** in the file.

IMPORTANT

After editing ***/etc/snmpd.conf***, you must copy it to the ***/boot/etc*** directory and reboot the system. It is very important to retain the access mode for the file (readable only by ***root***), so be sure to use **`cp -p`** when performing the copy. During the boot process, the files contained in the ***/boot/etc*** directory are copied to the working ***/etc*** directory on the system RAM disk. In this way the factory defaults are overwritten.

Configuration of SNMPv3

IMPORTANT

You must kill the **`snmpd`** daemon prior to editing, ***/boot/net-snmp/snmpd.conf***. Otherwise, the secret key creation may not complete properly. Issue the command **`/etc/rc.d/rc.snmpd stop`** to kill the **`snmpd`** daemon. You can verify that the **`snmpd`** daemon has been killed by issuing the **`ps -e`** command and verifying that it is not present.

If you are planning to use SNMPv3, you should definitely make use of the two resources mentioned previously (NET-SNMP website and *Essential SNMP*) and study them carefully. There are rather elaborate configuration options available when you are using v3. The instruction presented here will give you the flavor of the configuration but definitely not the full scope of possibilities. To access your Ninja via v3 of SNMP, you will have to configure two files:

/etc/snmpd.conf
/boot/net-snmp/snmpd.conf

The first file contains static configuration parameters that the agent uses to control access and to determine where to send notifications/traps. Other aspects of the agent's operation are also configurable in this file, but you should not need to modify those. To use the SNMPv3 capabilities of Ninja, you must first set up user information and access limits for those users in */etc/snmpd.conf.* Uncomment and edit these two lines to define your v3 users and their access parameters:

```
rwuser root    priv .1
rouser Ninja auth .1.3.6.1.4.1.13827
```

The first line defines a SNMPv3 read-write user *root* whose minimum security level will be authenticated and encrypted for privacy (choices are noauth, auth and priv), and who will have read-write access to the entire *iso(1)* branch of the SMI object tree. The second line defines a SNMPv3 read-only user *Ninja* whose minimum security level will be authenticated but not encrypted, and who will have read-only access to the entire *iso(1).org(3).dod(6).internet(1).private(4).enterprises(1).endRunTechnologiesMIB(13827)* branch of the SMI object tree. After adding the user lines to */etc/snmpd.conf*, copy it to the */boot/etc* directory using **`cp -p`**.

The second file is located on the non-volatile FLASH disk and is used by the SNMP agent to store "persistent data" that may be dynamic in nature. This may include the values of the MIB-II variables sysLocation, sysContact and sysName as well as any configured SNMPv3 user crypto keys. In order to use SNMPv3, you must configure user keys in this file for each SNMPv3 user that you have set up in */etc/snmpd.conf*. To do this, you must add lines to */boot/net-snmp/snmpd.conf* like these for each user:

```
createUser root MD5 endrun_1 DES endrun_1
createUser Ninja SHA NINJAGPS_0
```

The first line will cause the agent, **`snmpd`** to create a user *root* who may be authenticated via Message Digest Algorithm 5 (MD5) with password *endrun_1* and may use the Data Encryption Standard (DES) to encrypt the session data with passphrase *endrun_1*. The second line will cause a user *Ninja* to be created who may be authenticated using the Secure Hash Algorithm (SHA) with password *NINJAGPS_0*. Passwords and passphrases must have a *minimum* of 8 characters, or you will not be able to be authenticated.

After rebooting, the agent will read the */boot/net-snmp/snmpd.conf* configuration file and compute secret key(s) for each of the users and delete the **`createUser`** lines from the file. It will then write the secret key(s) to the file. These lines begin with the string, **`usmUser`**. In this way, un-encrypted passwords are not stored on the system.

IMPORTANT

To generate new keys, stop the **`snmpd`** process, delete the existing **`usmUser`** key lines from the file ***/boot/net-snmp/snmpd.conf*** and then add new **`createUser`** lines. Then reboot the system.

This example gives the simplest configuration to begin using SNMPv3 but doesn't make use of the full capabilities of the VACM in defining groups and views for fine-grained access control. The factory default */etc/snmpd.conf* file contains commented blocks of lines that can be uncommented to give you a basic configuration that uses the User-based Security Model (USM) described in RFC-2274 and the View-based Access Control Model (VACM) described in RFC-2275. The comments included in the file should help you in modifying it for your specific requirements.

Configuring SNMPv3 Notifications and Informs

If you have followed the steps in ***Configuration of SNMPv3*** (above), then you are almost ready to use SNMPv3 notifications and informs.

SNMPv3 uses the same **`trap2sink`** and **`informsink`** directives in */etc/snmpd.conf* as SNMPv2c. The difference being that **`snmptrap`** requires authorization and authentication information be provided to it when sending SNMPv3 notifications and/or informs. This additional information comes from the usmUser records in */boot/net-snmp/snmp.conf*. A usmUser record is a space delimited record on one line with the following fields:

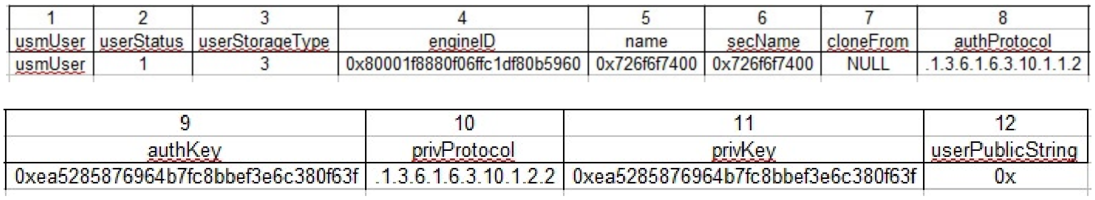
Field#	Field Name	Field Value
1	usmUser	usmUser
2	usmStatus	[a number (most likely 1)]
3	userStorageType	[a number (most likely 3)]
4	engineID	[0x followed by a number string/hash]
5	name	[0x followed by a number string/hash]
6	secName	[0x followed by a number string/hash]
7	cloneFrom	[NULL, unless user was created from a clone]
8	authProtocol	[a dotted number representing: MD5, SHA, ""]
9	authKey	[0x followed by a number string/hash]
10	privProtocol	[a dotted number representing: DES, AES, " "]
11	privKey	[0x followed by a number string/hash]
12	userPublicString	[0x followed by a number string/hash]

`snmptrap` requires (depending on the level of security) the use of fields 4, 5, 8, 9, 10, and 11. Your SNMP management station(s) will need to be configured to handle these hashed values.

Additionally, in */etc/snmptraps.conf* you will need to change the setting for V3 to ON. Copy *snmptraps.conf* to */boot/etc/* after you have made those changes so settings will be saved through reboots with *cp -a snmptraps.conf /boot/etc/snmptraps.conf*.

You should leave V1V2C set to ON until you verify that you can receive SNMPv3 notifications/ informs. Then you can change the value to OFF.

Example of usmUser Record



1	2	3	4	5	6	7	8
usmUser	userStatus	userStorageType	engineID	name	secName	cloneFrom	authProtocol
usmUser	1	3	0x80001f8880f06ffc1df80b5960	0x726f6f7400	0x726f6f7400	NULL	.1.3.6.1.6.3.10.1.1.2

9	10	11	12
authKey	privProtocol	privKey	userPublicString
0xea5285876964b7fc8bbef3e6c380f63f	.1.3.6.1.6.3.10.1.2.2	0xea5285876964b7fc8bbef3e6c380f63f	0x

The image above shows an example of a usmUser record, where the fields are:

Field#	Example Field Value
5 & 6	hashed from the value *root*
8	MD5
9	hashed from the value *endrun_1*
10	DES
11	hashed from the value *endrun_1*

Disable or Restrict Access

To disable SNMP, see ***Chapter 5 - Security, Disable SNMP, SSH and HTTPS***. To restrict access to specific hosts see ***Chapter 5 - Security, Restrict Access - Telnet, SSH and SNMP***.

Chapter *Seven*

Network Time Protocol (NTP)

This chapter describes how to configure the Ninja NTP Server. It also includes brief instruction for setting up NTP Clients on your Unix-like or Windows platform. This manual is not a 'How-To' on installing and using NTP. Only basic approaches to NTP client configuration for operation with Ninja will be described. If you have never used NTP, then you should spend some time reading the on-line documents, especially the Distribution Notes, FAQ and Configuration subject matter, which are available at:

For Linux:
nwtime.org/documentationandlinks/
or for Windows:
docs.microsoft.com/en-us/windows-server/networking/windows-time-service/windows-time-service-top

A simple introduction to NTP is here:
endruntechnologies.com/pdf/NTP-Intro.pdf

Configuring the NTP Server

Configuring the Ninja as a Stratum 1 Server

To configure your Ninja as a Stratum 1 NTP Server you must have successfully completed the Basic Installation procedures in Chapter 2. By default, Ninja is configured to respond to NTP requests from clients that may or may not be using MD5 authentication. If the clients are using MD5 authentication, they must be configured properly with the same MD5 authentication keys as Ninja. If you need to modify the factory default Ninja MD5 keys (recommended) or set up broadcast/multicast operation, then you will need to reconfigure the NTP subsystem. You may perform the configuration from either a **`telnet`** or **`ssh`** session or the local RS-232 console.

NOTE

If you would like to configure your server for multicast operation, configure it as you would for broadcast operation, with the exception that you must enter this specific NTP IPV4 multicast address: 224.0.1.1, or this specific IPV6 multicast address ff05::101, when you are prompted to enter the broadcast address.

Configuring NTP Using the Network Interface or Serial Port

The following shows the question and answer configuration utility called **`ntpconfig`**. The user-entered responses are shown in a larger font size.

```
Ninja(root@Ninja:~)-> ntpconfig

*******************************************************************************
*********************Network Time Protocol Configuration***********************
*******************************************************************************
*                                                                             *
*   This script will allow you to configure the ntp.conf and ntp.keys files   *
*   that control Ninja NTP daemon operation.                                  *
*                                                                             *
*   You will be able to create new MD5 authentication keys which are stored   *
*   in the ntp.keys file.                                                     *
*                                                                             *
*   You will be able to update the authentication related commands in the    *
*   ntp.conf file.                                                            *
*                                                                             *
*   You will be able to configure the “broadcast” mode of operation, with     *
*   or without authentication.  If you supply the multicast address instead   *
*   of your network broadcast address, then you will be able to configure     *
*   the time-to-live of the multicast packets.                                *
*                                                                             *
*   The changes you make now will not take effect until you re-boot the       *
*   Ninja.  If you make a mistake, just re-run ntpconfig prior to             *
*   re-booting.                                                               *
*                                                                             *
*   You will now be prompted for the necessary set up parameters.             *
*                                                                             *
*******************************************************************************
*******************************************************************************

---MD5 Keyfile Configuration


Would you like to create a new ntp.keys file? ([y]es, [n]o)  y


You will be prompted for a key number (1 - 65534), then the actual key.
When you have entered all of the keys that you need, enter zero at the next
prompt for a key number.

MD5 keys may contain from 1 to 31 ASCII characters.  They may not contain
contain SPACE, TAB, LF, NULL, or # characters!  If the key is longer than
20 characters, then only the valid hexadecimal characters
(0 - 9, a, b, c, d, e, f) may be used.


Enter a key number (1-65534) or 0 to quit:  1

Enter the key (1-31 ASCII characters):  EndRun_Technologies

Writing key number: 1 and Key: EndRun_Technologies to ntp.keys

Enter a key number (1-65534) or 0 to quit:  2

Enter the key (1-31 ASCII characters):  Ninja

Writing key number: 2 and Key: Ninja to ntp.keys

Enter a key number (1-65534) or 0 to quit:  0

---NTP Authentication Configuration
```

```
Do you want authentication enabled using some or all of the keys in
the ntp.keys file? ([y]es, [n]o)  y

You will be prompted for the key numbers (1 - 65534), that you want NTP to
“trust”.  The key numbers you enter must exist in your ntp.keys file.  If you
do not want to use some of the keys in your ntp.keys file, do not enter them
here.  NTP will treat those keys as “untrusted”.

Clients that use any of the “trusted” keys in their NTP polling packets will
receive authenticated replies from the Ninja.  When you have entered
all of the “trusted keys” that you need, enter zero at the next prompt for a
key number.

Enter a trusted key number (1-65534) or 0 to quit:  1

Enter a trusted key number (1-65534) or 0 to quit:  2

Enter a trusted key number (1-65534) or 0 to quit:  0

---NTP Broadcast/Multicast Configuration


Would you like to enable broadcast/multicast server operation? ([y]es, [n]o)  y

Set the network broadcast/multicast address for the Ninja
to use.  For broadcast mode on IPV4 networks, this address is
the all 1’s address on the sub-net.

Example: 111.112.113.255

On IPV6 networks, there is more than one way to
define a range of multicast addresses:

Example: ff05::1 (all nodes on the local site)
Example: ff02::1 (all nodes on the local link)

There are specific multicast addresses assigned for NTP Operation:

For IPV4 multicast operation, it is this specific address-> 224.0.1.1
For IPV6 multicast operation, it is this specific site scope address-> ff05::101

Enter IP address for NTP broadcast/multicast operation
(aaa.bbb.ccc.ddd or aaaa::bbbb ):  224.0.1.1

You have selected multicast operation.  Enter the TTL value that is
needed for multicast packets on your network (1, 32, 64, 96, 128, 160, 192, 224):  32

It is highly recommended that authentication be used if you are using NTP
in broadcast/multicast mode.  Otherwise clients may easily be “spoofed” by
a fake NTP server.  You can specify an MD5 key number that the Ninja
will use in its broadcast/multicast packets.  The clients on your network must
be configured to use the same key.

Would you like to specify an MD5 key number to use with
broadcast/multicast mode? ([y]es, [n]o)  y

Enter the MD5 key number to use (1-65534):  2
```

```
*******************************************************************************
*******************************************************************************
*                                                                             *
*  The Ninja Network Time Protocol configuration has been updated.            *
*                                                                             *
*             Please re-boot now for the changes to take effect.              *
*                                                                             *
*******************************************************************************
*******************************************************************************
*******************************************************************************
```

Configuring the Ninja as a Stratum 2 Server

Operating Ninja as a Stratum 1 Server is the recommended mode. However, there are times when Stratum 2 operation is a good strategy:

1. When you want a backup source of time. In this case, Ninja will operate as a Stratum 1 Server as long as it is locked to the GPS signal. If it loses the signal, then Ninja will start to drift away from "perfect" time. Eventually, when it has drifted 10 milliseconds, it reach the unlocked condition and stop serving time on your network. If you have Ninja configured for Stratum 2 operation, then it will continue serving time, using another Time Server as its reference. If Ninja is later able to acquire lock on the GPS signal again, it will switch back to Stratum 1 operation.

2. When you want your Ninja to serve accurate time, but you don't want to use the antenna (for some reason). In this case, Ninja can operate solely as a Stratum 2 server, with no antenna connected.

Since there are innumerable ways to configure your network with Stratum 2 servers, specific insructions for how to do that are beyond the scope of this manual. General instructions on how to edit the *ntp.conf* file are below.

Edit ntp.conf File

You must edit the ntp.conf file in order to point your Stratum 2 server at a Stratum 1 server. Edit */etc/ntp.conf* and add your server line(s). (See ***Appendix C - Helpful Linux Information*** for information on a simple editor.) Here is an example:

```
server 192.168.1.1
```

Or, if you have set up a domain name server via **`netconfig`**, here is another example:

```
server your.timeserver.com
```

IMPORTANT

Do not remove the server lines for the refclock. Even if your Time Server is not connected to an antenna, the refclock server lines must remain.

Now save the edited file and copy it to the non-volatile flash partition with this command:

```
cp -p /etc/ntp.conf /boot/etc
```

Mask Alarm

In Stratum 1 operation an alarm will be indicated when there is a loss of signal or if the antenna is not connected. For Stratum 2 operation you may not want to see these alarms. You can mask them (prevent them from showing) by using the console port (serial/network) commands **`setsigfltmask`** and **`setantfltmask`**.

Setting Up NTP Clients on Unix-like Platforms

To configure your Unix-like computer to use your Ninja, you must have successfully completed the NTP Server basic installation procedure described above. It is expected that you are, or have access to, a capable Unix/Linux system administrator and know more than a little about installing distributions from source code. Installation must be performed by a user with root priviledges on the system.

If you have access to a usenet news server, many problems may be solved by the helpful people who participate in the Internet news group devoted to NTP at *comp.protocols.time.ntp*.

Three methods of using Ninja with NTP clients on Unix-like platforms will be described:

Basic: This is the simplest, and will operate without MD5 authentication. **NTP beginners should always perform this setup first**.

MD5: This method is trickier only because MD5 keys must be set up and distributed accurately to the NTP clients in a secure way. Ninja is factory configured to authenticate its replies to NTP MD5 clients using its default set of keys.

Broadcast/Multicast: This method simplifies configuration of the clients on large networks since specific server addresses need not be configured in each client's */etc/ntp.conf* file. It can be configured either with or without MD5 authentication. However, it is highly recommended that authentication be configured when using broadcast/multicast mode due to the relative ease with which a fake NTP server can take over the clock setting of the broadcast/multicast clients on the network.

Unix-like Platforms: Basic NTP Client Setup

Basic setup is relatively simple, if:

- You have been able to successfully communicate with Ninja on your network.
- You have installed NTP on your client computer.

Configure NTP

You must edit the *ntp.conf* file which **`ntpd`**, the NTP daemon, looks for by default in the the */etc* directory. Add this line to the ntp.conf file:

```
server 192.168.1.120
```

This line tells **`ntpd`** to use the NTP server at address 192.168.1.120 in addition to any other servers which might also be configured in the client's *ntp.conf* file.

Restart **ntpd** to have it begin using Ninja. Use the NTP utility **ntpq** to check that **ntpd** is able to communicate with Ninja. After issuing the command

```
ntpq
```

you will see the **ntpq** command prompt:

```
ntpq>
```

Use the command

```
peers
```

to display the NTP peers which your computer is using. One of them should be the Ninja server which you have just configured. You should verify that it is being 'reached'. (You may have to continue issuing the peers command for a minute or two before you will see the 'reach' count increment.) If you have other peers configured, verify that the offset information for the Ninja server peer and your other peers is in agreement to within a few milliseconds, assuming that the other peers are synchronized to that level of accuracy.

It may also be useful to start the NTP daemon in 'debug' mode (**ntpd -d**) to confirm successful configuration. Refer to the NTP documentation for detailed usage of these debug utilities.

Unix-like Platforms: MD5 Authenticated NTP Client Setup

MD5 authenticated setup is relatively simple, if:

- You have been able to successfully communicate with Ninja on your network.

- Your Ninja has been configured to perform authentication either by factory default, or by running the **ntpconfig** shell script. The example Ninja authentication configuration shown in ***Configuring NTP Using the Network Interface or Serial Port*** above, will be assumed in the example configuration commands shown here.

- You have installed NTP on your client computer.

- You have successfully performed the ***Unix-like Platforms: Basic NTP Client Setup*** on your client computer.

Create the ntp.keys File

You must create a file named *ntp.keys* in the */etc* directory. It must be a copy of the one residing in the */etc* directory of your Ninja. You can **telnet** into your Ninja and start an **ftp** session with your client computer to send the Ninja's */etc/ntp.keys* file to your client computer, use the secure copy utility **scp**, or you can just use a text editor on your client computer to create an equivalent file.

IMPORTANT

Handling of the ***/etc/ntp.keys*** file is the weak link in the MD5 authentication scheme. It is very important that it is owned by ***root*** and not readable by anyone other than ***root***.

After transferring the file by **`ftp`**, and placing it in the */etc* directory on the client computer, issue these two commands at the shell prompt:

```
chown root.root /etc/ntp.keys
chmod 600 /etc/ntp.keys
```

Configure NTP

You must edit the *ntp.conf* file which **`ntpd`**, the NTP daemon, looks for by default in the */etc* directory. Assuming that you have created two trusted keys as shown in ***Configuring the NTP Server Using the Network Interface or Serial Port*** above, add these lines to the end of the *ntp.conf* file:

```
keys /etc/ntp.keys
trustedkey 1 2
```

Modify the line added previously in ***Unix-like Platforms: Basic NTP Client Setup*** so that authentication will be used with the Ninja server using one of the trusted keys, in this example, key # 1:

```
server 192.168.1.120 key 1
```

Restart **`ntpd`** to have it begin using the Ninja server with MD5 authentication. Use the NTP utility **`ntpq`** to check that **`ntpd`** is able to communicate with Ninja. After issuing the command

```
ntpq
```

you will see the **`ntpq`** command prompt:

```
ntpq>
```

Use the command

```
peers
```

to display the NTP peers which your computer is using. One of them should be the Ninja server which you have just configured. You should verify that it is being ‘reached’. (You may have to continue issuing the peers command for a minute or two before you will see the ‘reach’ count increment.) You can verify that authentication is being used by issuing the command

```
associations
```

to display the characteristics of the client server associations. In the “auth” column of the display, you should see “OK” for the row corresponding to the Ninja server. If you see “bad”, you should wait a few minutes to be sure that there is a problem since “bad” is the initial state of this setting. If the “bad” indication persists then you must check your configuration for errors. Typically this is due to a typing error in creating the */etc/ntp.keys* file on the client that causes a mismatch between the keys being used by the server and client. (If you transfer the file by **`ftp`** or **`scp`**, this shouldn't be a problem.) It is also possible to have a typing error in the */etc/ntp.conf* file that causes the needed key to not be included in the "trustedkey" list.

Unix-like Platforms: Broadcast/Multicast NTP Client Setup

Broadcast/multicast client setup is relatively simple, if:

- You have been able to successfully communicate with Ninja on your network.

- Your Ninja has been configured to perform broadcasts or multicasts by running the **`ntpconfig`** shell script. (This is not the factory default configuration, so be sure to run **`ntpconfig`**.) If you are going to use MD5 authentication, your Ninja must have been configured to operate with authentication in the broadcast/multicast mode, and you must know which of the trusted keys it is using for broadcast/multicast operation. The example Ninja configuration shown in ***Configuring the NTP Server*** above will be assumed in the example configuration commands shown here.

- You have installed NTP on your client computer.

- You have successfully performed the ***Unix-like Platforms: MD5 Authenticated NTP Client Setup*** on your client computer, if you plan to use MD5 authentication.

Configure NTP Client for Broadcast

You must edit the *ntp.conf* file which **`ntpd`**, the NTP daemon, looks for by default in the */etc* directory. Assuming that your Ninja server has been configured to use key 2 for broadcast authentication as shown in the example in ***Configuring the NTP Server*** above, make sure that key 2 is included in the **`trustedkey`** line, and add this line to the end of the *ntp.conf* file:

```
broadcastclient
```

If you are not using MD5 authentication, you would add these lines:

```
disable auth
broadcastclient
```

You may remove the line added previously in ***Unix-like Platforms: Basic NTP Client Setup***:

```
server 192.168.1.120
```

or the authenticated version added in ***Unix-like Platforms: MD5 Authenticated NTP Client Setup***:

```
server 192.168.1.120 key 1
```

Configure NTP Client for Multicast

You must edit the *ntp.conf* file which **`ntpd`**, the NTP daemon, looks for by default in the the */etc* directory. And add these lines for multicast:

```
multicastclient 224.0.1.1
```

or for IPv6:

```
multicastclient ff05::101
```

If you are not using MD5 authentication, you would add these lines:

```
disable auth
multicastclient 224.0.1.1
```

or for IPv6:

```
disable auth
multicastclient ff05::101
```

You may remove the line added previously in ***Unix-like Platforms: Basic NTP Client Setup***:

```
server 192.168.1.120
```

or the authenticated version added in ***Unix-like Platforms: MD5 Authenticated NTP Client Setup***:

```
server 192.168.1.120 key 1
```

Test Broadcast/Multicast

Restart **ntpd** to have it begin using Ninja as a broadcast or multicast server. Use the NTP utility **ntpq** to check that **ntpd** is able to communicate with Ninja. After issuing the command

```
ntpq
```

you will see the **ntpq** command prompt:

```
ntpq>
```

Use the command

```
peers
```

to display the NTP peers which your computer is using. One of them should be the Ninja server which you have just configured. You should verify that it is being ‘reached’. (You may have to continue issuing the peers command for a minute or two before you will see the ‘reach’ count increment.)

If you are using authentication, you can verify that authentication is being used by issuing the command

```
associations
```

to display the characteristics of the client server associations. In the “auth” column of the display, you should see “OK” for the row corresponding to the Ninja server. If you see “bad”, you should wait a few minutes to be sure that there is a problem since “bad” is the initial state of this setting. If the “bad” indication persists then you must check your configuration for errors. Typically this is due to a typing error in creating the */etc/ntp.keys* file on the client that causes a mismatch between the keys being used by the server and client. (If you transfer the file by **ftp** or **scp**, this shouldn’t be a problem.) It is also possible to have a typing error in the */etc/ntp.conf* file that causes the needed key to not be included in the “trustedkey” list.

Setting Up NTP Clients on Windows

To configure your Windows computer to use your Ninja, you must have successfully completed the procedures in ***Configuring the NTP Server*** above. Client installation must be performed by a user with administrative priviledges.

If you have access to a usenet news server, many problems may be solved by the helpful people who participate in the Internet news group devoted to NTP at *comp.protocols.time.ntp*.

The most common NTP client on Windows platforms is described below. Information on other NTP Client software is available at:

endruntechnologies.com/products/ntp-time-servers/ntp-client-software

Windows: W32Time

Windows uses a time service called W32Time which is automatically enabled by default during Windows installation. **`w32tm.exe`** synchronizes time in different ways, depending on the network implementation used. When peer-to-peer networking is used, then each individual workstation synchronizes to the NTP Server. For details, copy and paste this into your browser:

docs.microsoft.com/en-us/windows-server/networking/windows-time-service/windows-time-service-tools-and-settings

However, the most common method is with Windows Domain Networking. In this case, you must configure the Primary Domain Controller (PDC) to synchronize to the NTP Server. All other servers and workstations in the domain synchronize to the PDC. The default Windows installation procedure automatically configures workstations and servers to synchronize to the controlling PDC. So, only the PDC needs to be configured to synchronize to the NTP Server.

Security

For Unix-like platforms you can configure your NTP clients for secure MD5 authentication. See ***Unix-like Platforms: MD5 Authenticated NTP Client Setup***. You can also restrict NTP query access - see below.

Restrict NTP Query Access

The Network Time Protocol (NTP) implementation in Ninja is built from the reference distribution at:

nwtime.org/downloads/

By factory default, remote control and query of the NTP daemon **ntpd** is disabled. Query-only operation is supported only from processes running on Ninja itself, i.e. from the *localhost*. This restricts access to **ntpd** from remote hosts using either of the two NTP companion utilities **ntpq** and **ntpdc**.

Control via these two utilities is disabled in the */etc/ntp.conf* file in two ways. First, MD5 authentication keys are not defined for control operation via a *requestkey* or *controlkey* declaration. Second, this default address restriction line is present in the file:

```
restrict default nomodify noquery nopeer
restrict 127.0.0.1 nomodify
restrict 0::1 nomodify
```

The first line eliminates control and query access from ALL hosts. The second and third lines disable the localhost from making any modifications to the **ntpd** daemon, but query access is not affected by this restriction. These lines must not be removed, as they are necessary for various monitoring processes running on Ninja to function properly.

Knowledgable NTP users who would like to customize the security aspects of the configuration of the NTP daemon in Ninja should edit the */etc/ntp.conf* file directly and then copy it to the */boot/etc* directory. Be sure to retain the ownership and permissions of the original file by using **cp -p** when performing the copy.

CAUTION

If you are planning to make changes to the ***/etc/ntp.conf*** file, you must NOT restrict query access from the local host to the NTP daemon. Various system monitoring processes running on the system require this access.

An example follows which shows how to allow query access from a specific remote host with IP address 192.168.1.10 while also allowing processes running on Ninja to have query access as well:

```
restrict default noquery nomodify nopeer
restrict 127.0.0.1 nomodify
restrict 0::1 nomodify
restrict 192.168.1.10 nomodify
```

This page intentionally left blank.

Chapter *Eight*

IPv6 Information

The Ninja supports IPv6 out-of-the-box with a modern version 4.14.88 Linux kernel. During network configuration, you have the option to disable IPv6 on the Ethernet port. The IPv6 addressing scheme will see expanding deployment in the near future due to the fact that there are no longer any IPV4 addresses to be allocated in many regions of the world.

IPv6 Capabilities

The presence of an IPv6-capable kernel will automatically enable most of the IPv6 capabilities. By default, autoconfiguration of the Ethernet interfaces via IPv6 Router Advertisements is enabled. To disable acceptance of Router Advertisements, or to configure a static IPv6 address and default IPv6 gateway, and to configure IPv6 domain name servers, you must run the interactive **`netconfig`** script. This will allow you to configure your Ethernet interface for both IPv4 and IPv6 operation. You can also configure the hostname and domainname for the unit.

OpenSSH

By default, **`sshd`** is factory-configured to listen on both IPv4 and IPv6 addresses. It may be forced to listen on either IPv4 only, or IPv6 only by editing the */etc/ssh/sshd_config* file and modifying the **`AddressFamily`** directive, and then copying it to */boot/etc/ssh*. Refer to the *sshd_config* man page for detailed information (**`man sshd_config`**).

Hiawatha HTTPS

By default, **`hiawatha`** is factory-configured to listen on IPv4 only. It may be configured to listen on IPv6 only. Refer to ***Chapter 4 - HTTPS Interface, Configure HTTPS for IPv6***.

Net-SNMP

By default, **`snmpd`** is factory-configured to listen on both IPv4 and IPv6 addresses. This may be changed by editing */etc/rc.d/rc.snmpd* and modifying the agent address argument passed to **`snmpd`** at start-up, and then copying it to */boot/etc/rc.d*.

NTP

By default, **`ntpd`** is factory-configured to listen on both IPv4 and IPv6 addresses on all interfaces. This may be changed by editing */etc/ntp.conf* and adding the desired **`interface`** directives to achieve the desired behavior, and then copying it to */boot/etc*. For example, adding this line:

```
interface ignore ipv6
```

will cause ntpd to not bind to any IPv6 addresses. Refer to the NTP documentation for details on the `interface` directive.

IPv4-Only Protocols

There are several protocols running on Ninja which are not IPv6-capable: `telnet` (client and server), `ftp` and `dhcpcd`. Due to their intrinsic insecurity, `telnet` and `ftp` are rapidly being deprecated, and probably have little business running over an IPv6 network. The address autoconfiguration capabilities of IPv6 along with the Neighbor Discovery Protocol (NDP) make the DHCP protocol less important in IPv6 networks.

Chapter *Nine*

Inputs/Outputs (I/O)

Standard configuration for Ninja is one Ethernet port, an antenna input, an RS-232 serial port, and a DC power connector. Up to nine optional time and frequency outputs can be configured.

*Status and user settings for the output signals can be easily viewed and modified via the console port. Methods to do this are described in this chapter. Refer to **Appendix J - Specifications** for details on signals, connector types, pinouts, etc.*

Standard I/O

Antenna Jack

This SMA connector mates with the SMA-to-TNC adapter to connect with the downlead cable from the external antenna.

Ethernet Port

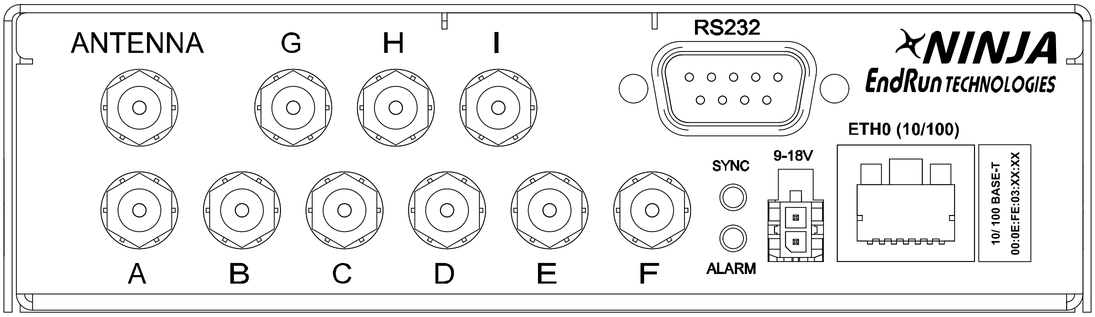
This RJ-45 connector mates with the Ethernet twisted pair cable from the network. It is labeled with a MAC address and “ETH0”. The upper-right green LED indicates link activity. Units with the IEEE-1588 PTP option indicate the link speed via the upper-left green LED. See ***Chapter 3 - Console Port Control and Status*** for more information.

Serial I/O Port

This DB9M connector provides the RS-232 serial I/O console interface. See ***Chapter 3 - Console Port Control and Status*** for details and ***Appendix J - Specifications, Serial Port I/O*** for pinout.

DC Power Input Jack

This 2-position jack provides connection to the DC power source. See ***Chapter 2 - Basic Installation, Performing a Site Survey*** for installation instructions.

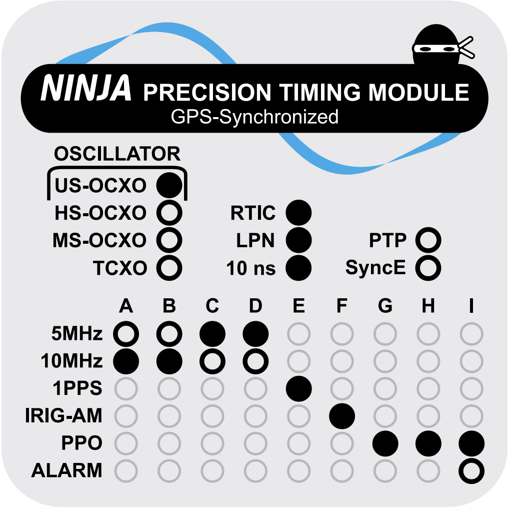


Output Options

In addition to the standard I/O described above, the Ninja can be configured with up to nine optional outputs. These outputs are labeled A through I. These optional outputs are described below.

Configuration Label

The label on top of each unit shows which signals are on which outputs (A through I). The label below indicates 10-Nanosecond Calibration with an Ultra-Stable OCXO and the RTIC Option, two 10-MHz and two 5-MHz low-phase-noise (LPN) outputs, a 1PPS Output, an IRIG-B AM Output and three Programmable Pulse Outputs (PPO).



NINJA PRECISION TIMING MODULE
GPS-Synchronized

OSCILLATOR

US-OCXO	●				
HS-OCXO	○	RTIC	●		
MS-OCXO	○	LPN	●	PTP	○
TCXO	○	10 ns	●	SyncE	○

	A	B	C	D	E	F	G	H	I
5MHz	○	○	●	●	○	○	○	○	○
10MHz	●	●	○	○	○	○	○	○	○
1PPS	○	○	○	○	●	○	○	○	○
IRIG-AM	○	○	○	○	○	●	○	○	○
PPO	○	○	○	○	○	○	●	●	●
ALARM	○	○	○	○	○	○	○	○	○

5MHz and10 MHz Sine Wave Options

The Ninja can provide up to four 5 MHz or up to four 10 MHz or a combination of both. These are sine wave outputs with the capability of being low-phase-noise, depending on the oscillator. For details see ***Appendix J - Specifications, Optional 5MHz and 10 MHz Outputs***.

View the Sine Wave Connectors

The **`cpuio`** command will list any connector on the Ninja that has an I/O signal. The Sine Wave connectors are identified as A, B, C and D.

Command: **`cpuio`**

Ninja reply:

```
CPU I/O A - 10MHz SINEWAVE OUTPUT is Installed
CPU I/O B - 10MHz SINEWAVE OUTPUT is Installed
```

1 PPS Option

The 1 PPS Option is a standard TTL output. The pulse width is normally 1 millisecond wide when shipped from the factory but can be changed (see below). For details on the 1 PPS signal definition see ***Appendix J - Specifications, Optional 1 PPS Output***.

View the 1 PPS Connector

The **cpuio** command will list any connector on the Ninja that has an I/O signal. The connector available for the 1 PPS signal is E.

```
Command:        cpuio
Ninja reply:    CPU I/O E - 1 PPS OUTPUT is Installed
                Current Setting = (See systemio command)
```

Change the 1 PPS Pulse Width

Use the **systemio** command to view the 1PPS pulse width. Use the **systemioconfig** command to change the pulse width. You will be able to choose from these pulse widths: 20 microseconds, 1 millisecond, 100 milliseconds and 500 milliseconds.

```
Command:        systemio
Ninja reply:    System I/O Signal 1 PPS OUTPUT is Installed
                Current Setting = 1 Milliseconds Pulse Width

Command:        systemioconfig
Ninja reply:    Interactive script is started so you can change the pulse width.
```

The 1 PPS is a “system signal”. This means that there is one 1 PPS signal that affects the whole system. In other words, if your Ninja has multiple 1 PPS outputs and you change the pulse width, then all 1 PPS outputs will be affected. This includes any 1PPS signal on a PPO connector.

IRIG-AM Option

The IRIG-AM Option is an amplitude-modulated (AM) time code output. The time code format is normally IRIG-B122 when shipped from the factory but can be changed (see below). For details on signal definition see ***Appendix J - Specifications, Optional IRIG-AM Output.***

View the IRIG-AM Connector

The **cpuio** command will list any connector on the Ninja that has an I/O signal. The connector available for IRIG-AM option is F.

```
Command:        cpuio
Ninja reply:    CPU I/O F - AM TIME CODE OUTPUT is Installed
                Current Setting = (See systemio command)
Command:        cpuioconfig
Ninja reply:    Interactive script is started so you can change the ulse rate setting.
```

Change the AM Time Code Format

Use the **systemio** command to view the time code format. Use the **systemioconfig** command to change the format. Changing the time code format will change it at the analog time code output and also on the digital time code output (via PPO connector, if any).

```
Command:        systemio
Ninja reply:    System I/O Signal TIME CODE OUTPUT is Installed
                Current Setting = IRIG-B122/B002 Format

Command:        systemioconfig
Ninja reply:    Interactive script is started so you can change the Time Code format.
```

Programmable Pulse Output (PPO) Option

The PPO Option provides user-selectable, on-time pulse rates from 1 PPS to 10 MPPS, or a digital time code. Other selections are 1PP60S (pulse per 60 seconds, on the minute), 1PP2S (pulse per 2 seconds, on the even second), and Trigger PPO (see below). For details on signal definition see ***Appendix J - Specifications, Optional Programmable Pulse Output***.

View and Change the PPO

The **`cpuio`** command will list any connector on the Ninja that has an I/O signal. Connectors available for the PPO are G, H and I. The **`cpuioconfig`** command will allow you to change the pulse rate selection.

```
Command:        cpuio
Ninja reply:    CPU I/O G - PROGRAMMABLE PULSE OUTPUT is Installed
                Current Setting = OFF

Command:        cpuioconfig
Ninja reply:    Interactive script is started so you can change the pulse rate setting.
```

View and Change the Digital Time Code

Use the **`systemio`** command to view the time code format. Use the **`systemioconfig`** command to change the format. Changing the time code format will change it at the analog time code output (IRIG-AM, if any) and the digital time code output.

```
Command:        systemio
Ninja reply:    System I/O Signal TIME CODE OUTPUT is Installed
                Current Setting = IRIG-B122/B002 Format
```

Trigger PPO Function

When the PPO option is installed on the CPU Module, then the **`triggerppo`** command is available via the console port. The **`triggerppo`** is used to generate an on-time pulse via the PPO connector. Multiple instances of this command can be running for different pulse output times, allowing scheduling of multiple triggers.

The format of the command is:

```
triggerppo = X hh:mm:ss<CR>
```

Where:

X is the CPU I/O port of the PPO Option BNC (G, H or I).

hh:mm:ss is the UTC time-of-day for the trigger, with 1 second resolution.

<CR> is the ASCII carriage return character (0x0D).

An example is shown here:

Command:	**`triggerppo = G 13:50:00`**	This will cause a pulse on BNC A at
Ninja reply:	**`OK`**	UTC time-of-day 13:50:00.

Trigger PPO Specifications

Rising edge on time, aligned with the desired second.
Accuracy < 25 nanoseconds RMS to UTC(USNO) when locked. (10 ns with Calibration Option.)
Pulse width mimics the standard 1PPS Output pulse width (20 us, 1 ms, 100 ms, or 500 ms).
Multiple trigger times must be at least 2 seconds apart.
Other PPO specifications are in ***Appendix J - Specifications, Optional Programmable Pulse Output***.

Trigger PPO Operational Details

The **`triggerppo`** monitors the system time and compares it to the trigger time passed to it via the console port interface. When the system time is about 1⁄2 second before the trigger time, then the PPO is armed so that a single pulse occurs at the next second mark. When the system time is about a 1⁄2 second after the trigger time, then the PPO is disarmed. For other details, at the console port type:

```
help triggerppo
```

Alarm Option

The Alarm Option provides an open-collector output that indicates when the GPS Subsystem has lost lock, or when serious hardware faults are detected. For a detailed description of the faults see ***Appendix G - System Faults***.

Care should be taken not to directly connect this open-collector output to a voltage source. A series current-limiting resistor of at least 1k ohms in value should be used. The pull-up voltage must not exceed 40V. For more details see ***Appendix J - Specifications, Optional Alarm Output***.

View the Alarm Output Connector

The **`cpuio`** command will list any connector on the Ninja that has an I/O signal. The connector available for the Alarm Output is identified as I.

Command:	**`cpuio`**
Ninja reply:	**`CPU I/O I - OPEN COLLECTOR ALARM OUTPUT is Installed`**

This page intentionally left blank.

Chapter *Ten*

Optional Precision Time Protocol (PTP/IEEE-1588)

This chapter contains the configuration and status information for the optional Precision Time Protocol. PTP version 2 is supported. Both the default profile and the IEEE 802.1AS profile are available. The PTP protocol running on Ninja is a full Grandmaster Clock implementation of the IEEE-1588-2008 standard.

Option

The PTP/IEEE-1588 protocol is an optional feature in the Ninja. It is not a software option that you can install after purchase. You must have it installed at the factory. If you are unsure whether your device has PTP installed then you check the label on top of the unit. There will be a PTP indicator. If you are remote then you can use the **`ptpstat`** command. If the reply is “command not found” then the PTP Option is not installed. For example:

Command:	**`ptpstat`**	
Ninja reply:	**`command not found`**	(PTP Option is not installed.)

About PTP

The PTP implementation in Ninja is based on the distribution at the linuxptp website:

linuxptp.sourceforge.net/

For more information about the **`ptpd`** daemon and to obtain PTP Slave software, refer to the Linux PTP website above.

An excellent book which describes the PTP Master and Slave operation is:

Measurement, Control, and Communication using IEEE 1588,
John C. Eidson, Springer, November 2006.

More information on IEEE-1588 PTP can be found at the NIST National Institute of Standards and Technology IEEE 1588 website:

nist.gov/el/isd/ieee/ieee1588.cfm

PTP Configuration and Status

The PTP daemon status and configuration is supported from two PTP companion utilities **`ptpstat`** and **`ptpconfig`**. The following table shows Ninja utilities that pertain to PTP:

	Daemon	Status	Configuration
PTP	**`ptpd`**	**`ptpstat`**	**`ptpconfig`**

The default PTP configuration settings in the Ninja are shown below. If you need to modify these settings then you will need to reconfigure the PTP Subsystem. You may perform the configuration from either a telnet or ssh session, or the local RS-232 console. Default PTP settings are:

Default PTP Configuration Settings

Profile	Default
Sync Interval	1 second
Announce Inverval	2 seconds
Priority 1	128
Priority 2	128
Delay Mechanism	E2E
Domain	0
PTP TTL	1
Transmission Mode	Multicast

PTP Configuration

The **`ptpconfig`** command starts an interactive shell script that will allow you to configure the PTP Subsystem of Ninja. You will be prompted to set PTP parameters as follows:

Profile	Default	802.1AS
Sync Interval (per second)	1, 2, 3, 4, 16, 32, 64, 128	N/A
Announce Interval (seconds)	1, 2, 4, 8 or 16	N/A
Priority1	0-255	0-255
Priority2	0-255	0-255
Delay Mechanism	E2E or P2P	N/A
Domain	0-255	0-255
PTP TTL	1-255	1-255
Transmission Mode	Multicast or Hybrid	N/A

PTP Config for Default Profile

The following is a transcript of the question and answer configuration utility provided by **`ptpconfig`**. The utility modifies this non-volatile file: */boot/etc/ptp.conf*. The user-entered parameters for the default profile are underlined:

```
Ninja (root@NinjaPTP:~)-> ptpconfig
****************************************************************************
*********Precision Time Protocol Configuration******************************
****************************************************************************
*
*   This interactive utility will guide you in configuring the ptp daemon
*   configuration file that controls its operation.
*
*   You will be able to configure the PTP profile.
*
*   For the default profile you will be able to configure: sync interval,
*   announce interval, priority1, priority2, delay mechanism , ptp domain,
*   time-to-live (TTL), and transmission mode.
*
*   For the 802.1AS profile you will be able to configure: priority1,
*   priority2,and time-to-live (TTL).
*
*   The changes you make now will not take effect until you re-boot.
*
*   If you make a mistake, just re-run ptpconfig prior to re-booting.
*
*   You will now be prompted for the necessary set up parameters.
****************************************************************************
****************************************************************************

Set the PTP Profile (Default or 802.1AS) Default

Set the PTP Sync Interval in packets per second (1,2,4,8,16,32,64,128) 1

Set the PTP Announce Interval in seconds (1,2,4,8,16) 2

Set the PTP Priority1 value (0 - 255) 128

Set the PTP Priority2 value (0 - 255) 128

Set the PTP Delay Mechanism (E2E or P2P) E2E

Set the PTP Domain value (0 - 255) 0

Set the PTP TTL value (1 - 255) 1

Set the PTP Transmission Mode (Multicast or Hybrid) Multicast

********************************************************************************
********************************************************************************
*                                                                              *
*  The Precision Time Protocol IEEE-1588 configuration has been updated.   *
*                                                                              *
*             Please re-boot now for the changes to take effect.              *
*                                                                              *
********************************************************************************
********************************************************************************
********************************************************************************
```

PTP Config for 802.1AS Profile

The following is a transcript of the question and answer configuration utility provided by **`ptpconfig`**. The utility modifies this non-volatile file: */boot/etc/ptp.conf*. The user-entered parameters for the 802.1AS profile are underlined:

```
Ninja (root@NinjaPTP:~)-> ptpconfig
********************************************************************************
*************Precision Time Protocol Configuration******************************
********************************************************************************
*                                                                              *
*   This interactive utility will guide you in configuring the ptp daemon      *
*   configuration file that controls its operation.                            *
*                                                                              *
*   You will be able to configure the PTP profile.                             *
*                                                                              *
*   Default profile you will be able to configure, sync interval,              *
*   announce interval, priority1, priority2, delay mechanism , ptp domain,     *
*   time-to-live (TTL), and transmission mode.                                 *
*                                                                              *
*   802.1AS profile you will be able to configure, priority1, priority2,       *
*   and time-to-live (TTL).                                                    *
*                                                                              *
*   The changes you make now will not take effect until you re-boot.           *
*   If you make a mistake, just re-run ptpconfig prior to                      *
*   re-booting.                                                                *
*                                                                              *
*   You will now be prompted for the necessary set up parameters.              *
*                                                                              *
********************************************************************************
********************************************************************************

Set the PTP Profile (Default or 802.1AS) 802.1AS

Set the PTP Priority1 value (0 - 255) 128

Set the PTP Priority2 value (0 - 255) 128

Set the PTP Domain value (0 - 255) 0

Set the PTP TTL value (1 - 255) 1


********************************************************************************
********************************************************************************
*                                                                              *
*  The Precision Time Protocol IEEE 802.1AS configuration has been updated.    *
*                                                                              *
*             Please re-boot now for the changes to take effect.               *
*                                                                              *
********************************************************************************
********************************************************************************
********************************************************************************
```

PTP Status

The **`ptpstat`** command allows you to query the status of the PTP Subsystem. Following is the response to this command:

```
V  SI  AI  P1  P2  DM  DOM  TTL  CLASS  STATE  CLKID  UTC UTCV  CA  L59  L61  TT  FT  TM  TR  PR
```

Where:

V	is the IEEE-1588 version 2 for the 2008 standard.

SI	is the PTP sync interval either 1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, or 1/128 seconds.

AI	is the PTP announce interval, either 1, 2, 4, 8, or 16 seconds.

P1	is the PTP priority 1 in a range from 0 to 255.

P2	is the PTP priority 2 in a range from 0 to 255.

DM	is the PTP delay mechanism , either E2E or P2P.

DOM	is the PTP domain, in a range from 0 to 255.

TTL	is the PTP multicast time-to-live in a range from 1 to 255.

CLASS	is the PTP clock class one of SYNCHRONIZED, HOLDOVER, or UNLOCKED.

STATE	is the PTP port state one of MASTER, PASSIVE, LISTENING or INITIALIZING.

CLKID	is the PTP clock source either GPS or OSC.

UTC	is the PTP utc offset in seconds from TAI.

UTCV	is the PTP utc offset valid, either TRUE or FALSE.

CA	is the PTP clock accuracy one of 25ns, 100ns, 250ns, 1us, 2.5us, 10us, 25us, 100us, 250us, 1ms, 2.5ms, 10ms, or Unknown.

L59	is the PTP leap 59 second indicator, either TRUE or FALSE.

L61	is the PTP leap 61 second indicator, either TRUE or FALSE.

TT	is the PTP time traceable indicator, either TRUE or FALSE.

FT	is the PTP frequency traceable indicator, either TRUE or FALSE.

TM	is the PTP transmission mode, either MULTICAST or HYBRID.

TR	is the PTP transport, either UDPv4 or L2.

PR	is the PTP Profile, either Default or 802.1AS

PTP Operation

The Ninja is configured as an IEEE-1588 Grandmaster Clock using the default profile or the 802.1AS profile. Verify that the network settings have been configured and tested using **`netconfig`**. Once the network has been configured, the Ninja will begin to transmit PTP Sync messages after it is locked.

The PTP Sync Interval is user configured. 1, 2, 4, 8, 16, 32, 64, or 128 packets per second are transmitted as a multicast. The packets are only transmitted when the clock is fully synchronized or in holdover with a known clock accuracy.

The PTP Announce Interval is user configured. Packets are transmitted every 1, 2, 4, 8, or 16 seconds as a multicast. The packets are only transmitted when the clock is fully synchronized or in holdover with a known clock accuracy.

The PTP Priority 1 is user configured in a range from 0 to 255.

The PTP Priority 2 is user configured in a range from 0 to 255.

NOTE

If using a single Grandmaster, keep the default setting of 128 for Priority 1 and Priority 2. If using two redundant Grandmasters, then you can configure the preferred clock by setting Priority 1 to 127 and Priority 2 to 128.

The Delay Request Interval is not user-configurable. It is set to 32 seconds.

The PTP Delay Mechanism is user configured to either E2E or P2P. E2E uses the delay request-response mechanism and P2P uses the peer delay mechanism.

The PTP Domain is user-configured in a range from 0 to 255.

The PTP Multicast TTL is user configured in a range from 1 to 255. For a local area network the TTL should be configured to 1.

PTP Clock Class one of SYNCHRONIZED, HOLDOVER, or UNLOCKED. The Clock Class is SYNCHRONIZED when the GPS Subsystem TFOM level is less than or equal to 4 (see ***Appendix A - TFOM***). The Clock Class is HOLDOVER when the GPS Subsystem TFOM level is greater than 4 and less than 9. The Clock Class is UNLOCKED when the GPS Subsystem TFOM level is 9.

The PTP State is the state of the port. When the Ninja is the Master PTP Clock on the network, the State will report MASTER. Otherwise, the State will report PASSIVE, LISTENING or INITIALIZING.

The PTP Clock Source is either GPS or OSC. The Clock Source is GPS if the Clock Class is Synchronized, otherwise it is OSC based on the system oscillator.

The PTP UTC Offset is the offset between TAI and UTC in units of seconds.

The PTP UTC Offset Valid is either TRUE or FALSE. The UTC Offset Valid is TRUE if the current UTC Offset is known to be correct, otherwise it is FALSE.

The PTP Clock Accuracy is transmitted when the time is accurate to within the the following:

25ns	Clock is synchronized or in holdover, PTP clock < 25 nanoseconds
100ns	Clock is synchronized or in holdover, PTP clock < 100 nanoseconds
250ns	Clock is synchronized or in holdover, PTP clock < 250 nanoseconds
1us	Clock is synchronized or in holdover, PTP clock < 1 microsecond
2.5us	Clock is synchronized or in holdover, PTP clock < 2.5 microseconds
10us	Clock is synchronized or in holdover, PTP clock < 10 microseconds
25us	Clock is synchronized or in holdover, PTP clock < 25 microseconds
100us	Clock is synchronized or in holdover, PTP clock < 100 microseconds
250us	Clock is synchronized or in holdover, PTP clock < 250 microseconds
1ms	Clock is synchronized or in holdover, PTP clock < 1 millisecond
2.5ms	Clock is synchronized or in holdover, PTP clock < 2.5 milliseconds
10ms	Clock is synchronized or in holdover, PTP clock < 10 milliseconds
Unknown	Clock is unsynchronized, TFOM = 9

The PTP Leap 59 Second Indicator is either TRUE or FALSE. The Leap 59 is TRUE if the PTP Timescale is PTP and the last minute of the current UTC day contains 59 seconds, otherwise it is FALSE.

The PTP Leap 61 Second Indicator is either TRUE or FALSE. The Leap 61 is TRUE if the PTP Timescale is PTP and the last minute of the current UTC day contains 61 seconds, otherwise it is FALSE.

The PTP Time Traceable Indicator is either TRUE or FALSE. The Time Traceable is TRUE if the Time Scale is PTP and the Clock Class is Synchronized or Holdover, otherwise it is FALSE.

The PTP Frequency Traceable indicator is either TRUE or FALSE. The Frequency Traceable is TRUE if the Time Traceable is TRUE, otherwise it is FALSE.

The PTP Transmission Mode is either Multicast or Hybrid. Multicast Mode is the default and is defined in the IEEE-1588 standard. All packets sent from the Grandmaster are Multicast. Hybrid Mode uses Multicast and Unicast. In this mode, delay response messages are sent Unicast in response to the slave delay request. NOTE: Unicast messages are only sent when the Delay Mechanism is configured to E2E.

The PTP Transport is either UDPv4 Layer 3 in the Default profile or Layer 2 in the 802.1AS profile.

The PTP Profile is either Default as defined in the IEEE-1588 standard or 803.1AS as defined in the IEEE-802.1AS standard.

About PTP Seconds

The IEEE standard defines the PTP epoch beginning at 0 hours on 1 January 1970. The time measured since this epoch is designated in the standard as PTP seconds. The PTP second is monotonic so does not include leap seconds.

Disable the PTP Protocol

The instructions below assume that the PTP Option has been installed on your Ninja. To check, see the section titled ***Option*** at the beginning of this chapter.

To disable the Precision Time Protocol issue the following command:

```
chmod -x /etc/rc.d/rc.ptpd
```

Copy the *rc.ptpd* file to the non-volatile FLASH area like this:

```
cp -p /etc/rc.d/rc.ptpd /boot/etc/rc.d
```

Then:

```
reboot
```

Once PTP has been disabled, it will no longer execute. However, the web interface will still show PTP and the PTP commands will still exist on the system.

Re-Enable PTP

To re-enable PTP, remove the *rc.ptpd* file from the */boot/etc/rc.d* directory as shown below:

```
rm /boot/etc/rc.d/rc.ptpd
```

Then:

```
reboot
```

Chapter *Eleven*

Optional Synchronous Ethernet (SyncE)

When locked to GPS, the Ninja can operate as a Synchronous Ethernet Primary Reference Clock (PRC). This chapter contains the information for the optional Synchronous Ethernet protocol (SyncE). SyncE is most often used in telecom applications.

Option

The Synchronous Ethernet protocol is an optional feature in the Ninja. It is not a software option that you can install after purchase. You must have it installed at the factory. If you are unsure whether your device has SyncE installed then you can check the label on top of the unit. There will be a SyncE indicator. If you are remote then you can use the **`syncestat`** command. If the reply is "command not found" then the SyncE Option is not installed. For example:

```
Command:       syncestat
Ninja reply:   command not found          (SyncE Option is not installed.)
```

About SyncE

Synchronous Ethernet, also referred as SyncE, is an ITU-T standard for computer networking that facilitates the transference of clock signals over the Ethernet physical layer (Layer 1). In addition to the physical clock signals, the Synchronization Status Messaging (SSM) is transmitted on the data link layer (Layer 2). The Ethernet Synchronization Messaging Channel (ESMC) contains the clock Quality Level (QL) with the embedded SSM code.

The Ninja is configured as a Synchronous Ethernet Primary Reference Clock (PRS) when tightly locked to GPS (TFOM <= 4. See ***Appendix A - TFOM***). There is no SyncE user configuration required.

Synchronous Ethernet is defined in the ITU-T:
G.8261 Architecture and wander performance
G.8262 Timing characteristics
G.8264 Ethernet Synchronization Message Channel (ESMC)

More information on SyncE can be found at the ITU-T Telecommunications Standardization Sector website:

itu.int/en/ITU-T/Pages/default.aspx

Operation

As stated above, the clock signal is transmitted on Layer 1 (physical) and the Synchronization Status Messaging (SSM) is transmitted on layer 2 (data link). If SyncE is enabled, you would use **`syncestat`** for the SSM code.

Verify that the network settings have been configured and tested using **`netconfig`**. Once the network has been configured, the Ninja will synchronize the Ethernet and begin to transmit the SSM code.

The Ethernet Synchronization Messaging Channel (ESMC) contains the clock Quality Level (QL) with the embedded SSM code. The ESMC data will be sent at a rate of once per second.

Status

The **`syncestat`** command allows you to query the status of SyncE. When the clock is locked, the ESMC and **`syncestat`** will report the quality level of QL-PRC and set the SSM code. When the clock is not locked (TFOM > 4), syncestat will report the quality level of QL-DNU and set the SSM code.

QL-PRC: Primary Reference Clock that is defined in ITU-T G.811.

QL-DNU: Do Not Use this signal for synchronization.

For example:

```
Command:        syncestat
Ninja reply:    SSM Code: QL-PRC
```

or

```
Command:        syncestat
Ninja reply:    SSM Code: QL-DNU
```

Disable the SyncE Protocol

The instructions below assume that the SyncE Option has been installed on your Ninja. To check, see the section titled ***Option*** at the beginning of this chapter.

To disable SyncE issue the following command:

```
chmod -x /etc/rc.d/rc.synced
```

Copy the *rc.synced* file to the non-volatile FLASH area like this:

```
cp -p /etc/rc.d/rc.synced /boot/etc/rc.d
```

Then:

```
reboot
```

Once SyncE has been disabled, it will no longer execute. However, the web interface will still show SyncE and the **`syncestat`** command will still exist on the system.

Re-Enable SyncE

To re-enable SyncE, remove the *rc.synced* file from the */boot/etc/rc.d* directory as shown below:

```
rm /boot/etc/rc.d/rc.synced
```

Then:

```
reboot
```

This page intentionally left blank.

Chapter *Twelve*

Real-Time Ionospheric Corrections (RTIC) Option

This chapter describes the RTIC Option, its operation, and the time and frequency performance benefits to Ninja. Tests described in this chapter were done with the Meridian II Precision TimeBase. Meridian II timing components are identical to those in the Ninja.

Option

The RTIC Option is an optional feature in the Ninja Precision Timing Module. It is not a software option that you can install after purchase. You must have it installed at the factory. If you are unsure whether your device has RTIC installed then you can check the label on top of the unit. There will be an RTIC indicator. If you are remote then you can use the **`ionostat`** command. If the reply is "command not found" then the RTIC Option is not installed. For example:

```
Command:        ionostat
Ninja reply:    command not found          (RTIC Option is not installed.)
```

About RTIC

The Real-Time Ionospheric Corrections (RTIC) software option for the Ninja Precision Timing Module optimizes time and frequency stability and accuracy. The RTIC option uses proprietary algorithms within EndRun's L1 GPS timing receiver to directly measure and remove ionospheric delays in real-time. This unprecedented real-time capability was previously only available with expensive dual frequency L1/L2 GPS receivers.

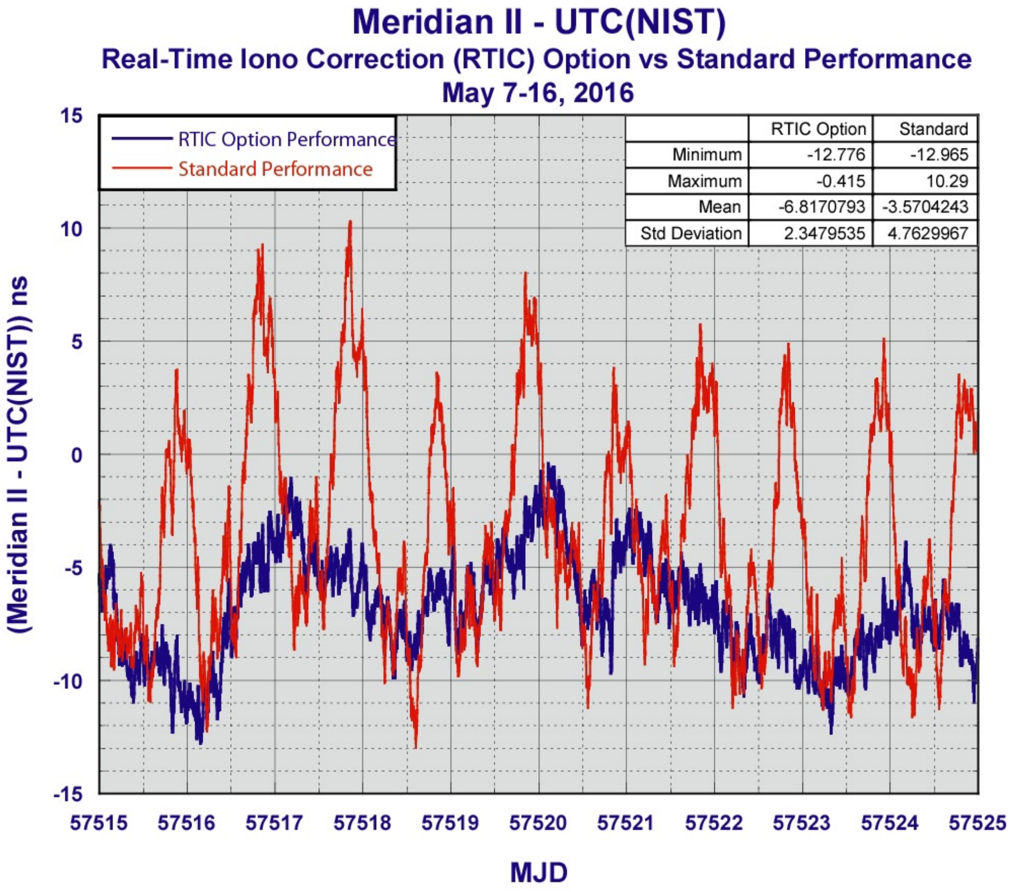
Ionospheric Delay and Impact To GPS Time-Transfer

The largest contributor to GPS time-transfer error is the variable delay of the satellite signals as they pass through the ionosphere, a layer of ionized particles a few hundred kilometers above the Earth's surface. The ionization is caused by solar radiation phenomena, and is maximum a little after local noon and minimum a little after local midnight. The GPS signal delay through the ionosphere is proportional to the ionization level, expressed as Total Electron Content (TEC). Data transmitted from the satellites contain a model (Klobuchar) that receivers may use to partially compensate for this delay. This model, however, provides only a coarse compensation for night-to-day variations in the ionosphere. It is unable to compensate for the much shorter-term variations in the ionospheric delay caused by various types of solar "storms", and it cannot keep up with the day-to-day variations. As such it was never intended to achieve more than about a 50% improvement over not compensating for the ionospheric delay at all.

Our Proprietary Solution

The GPS code modulation and carrier phase delays are affected differently as they propagate through the ionosphere. By recognizing a few other aspects of this behavior, EndRun developed the RTIC algorithm to directly quantify the delay through the ionosphere, and resolve the code phase and carrier phase bias. The bias information enables real-time measured delay compensation in EndRun's proprietary, single-frequency, L1 GPS receiver. *To our knowledge, this capability is unique to EndRun Technologies*.

The algorithm is effective when the receiver is operating at a stationary location with continuous tracking of at least one satellite at all times. After a 24-hour initialization period it begins producing real-time ionospheric delay measurements, with full accuracy of those measurements achieved after several days. When ionospheric conditions are quiet or relatively normal, this option improves the accuracy and stability of the timing outputs by as much as a factor of three at observation intervals between 3,000 and 100,000 seconds. During a major ionospheric storm event, the improvement may be much greater. It is very difficult to achieve this stability over these observation intervals by another method, as only a high-performance cesium frequency standard is stable enough to act as a filter for the timing errors caused by the GPS broadcast model. The following phase plot shows the Meridian II / Ninja with the RTIC option active (blue) compared to what the performance would have been using the GPS broadcast model (red). Throughout the 10-day test period the RTIC option demonstrated substantial improvement to frequency and time-domain stability through real-time ionospheric delay measurements and compensation.

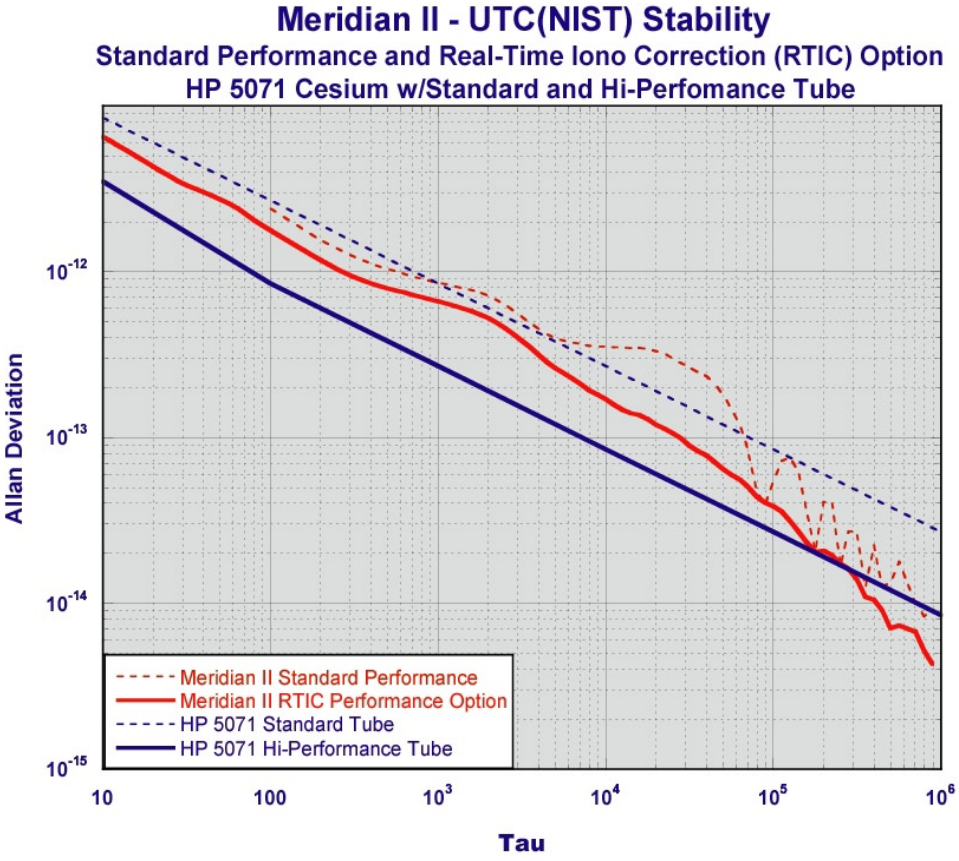


	RTIC Option	Standard
Minimum	-12.776	-12.965
Maximum	-0.415	10.29
Mean	-6.8170793	-3.5704243
Std Deviation	2.3479535	4.7629967

Performance Verified at National Institute of Standards and Technology (NIST)

To characterize the performance gain of the RTIC Option, a Meridian II was sent to NIST in May of 2016, where its 1PPS output was monitored continuously relative to UTC(NIST) for over 30 days. The RTIC Option was active in the Meridian II during this time. The ionospheric delay corrections calculated using the broadcast ionospheric model, along with the real-time corrections computed using the RTIC Option, were logged. With these logs, the NIST timing measurements were post-processed to determine what the performance would have been if the RTIC Option had not been active, and the broadcast model had instead been used.

The following chart shows the difference in stability between the two delay compensation methods of the Meridian II / Ninja, along with the stability of the two performance levels of HP 5071 Cesium standards for comparison. The Meridian II / Ninja stability performance with the RTIC option exceeds the standard HP 5071 Cesium at all measurement intervals and exceeds the High Performance HP 5071 after 300,000 seconds.



RTIC Configuration and Status

RTIC Configuration and Status Using the Network or Serial Port

The commands listed below allow you to get the current RTIC status and turn RTIC on or off.

ionostat

This command shows parameters associated with the RTIC Option. The string contains several parameters in the format shown below:

YYYYMMDD.HH:MM:SS K.KKKe-KK V +S.SSSe-SS

where:

YYYYMMDD.HH:MM:SS is the timestamp of the data.
K.KKKe-KK is the standard broadcast Klobuchar model ensemble ionospheric delay in seconds.
V is the validity of the Real-Time ionospheric delay calculations, T(rue) or F(alse).
+S.SSSe-SS is the Real-Time ensemble ionospheric delay calculation in seconds.

Here is an example:

```
Command:       ionostat
Ninja reply:   20160916.18:13:10 1.093e-08 F +1.093e-08
```

rticmode

This command shows the operating state of the Real-Time Ionospheric Correction Option. It is either OFF or ON. When ON, Real-Time ionospheric delays are calculated and used to compensate for the GPS signal delay through the ionosphere. When OFF, the Real-Time delays continue to be calculated but the standard broadcast Klobuchar ionospheric model delays are used to compensate for the GPS signal delay through the ionosphere.

```
Command:       rticmode
Ninja reply:   ON
```

setrticmode

This command allows the root user to manually set the Real-Time Ionospheric Correction Option mode to either OFF or ON. For the highest accuracy and stability at a static location with good satellite visibility, this should be set to ON. Even if this setting is ON, it will be ignored if the GPS Dynamic mode is also ON, and real-time ionospheric delays will not be calculated. Factory default mode is ON.

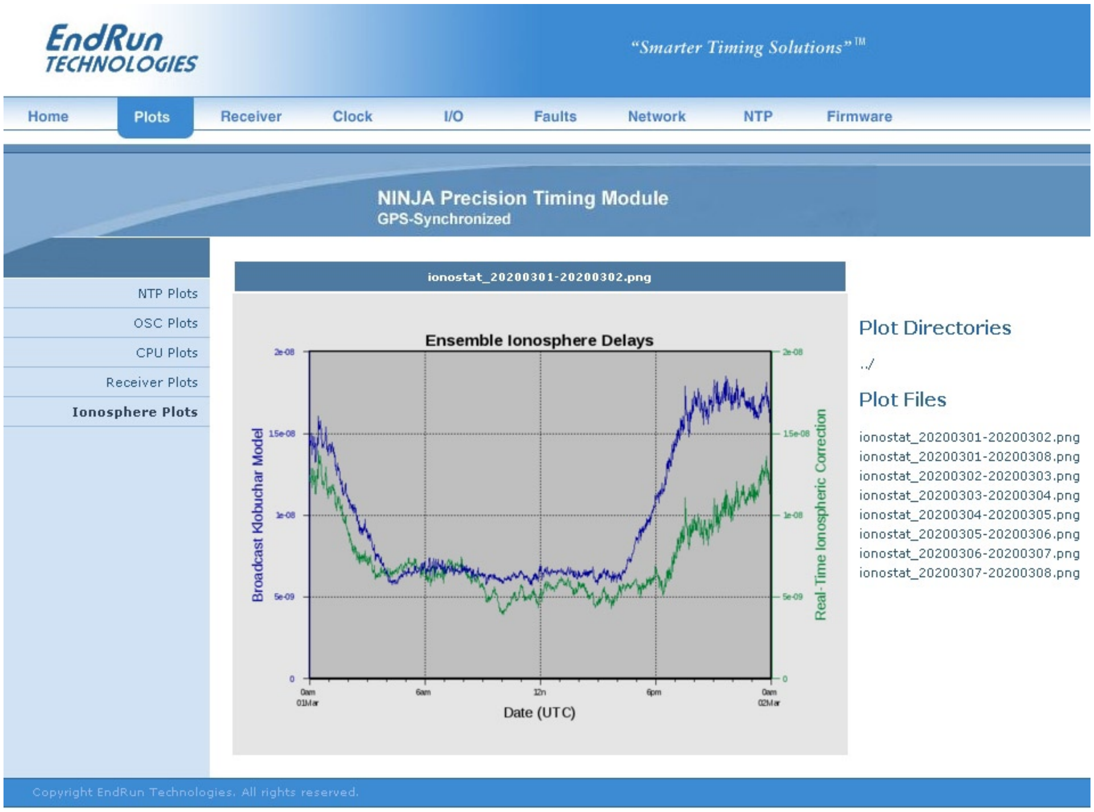
NOTE: The Real-Time Ionospheric Correction Option requires a minimum of 24 hours of operation after lock to GPS to begin producing enhanced accuracy ionospheric delay calculations, and these will reach their full accuracy after several days.

It accepts one command line argument: either ON or OFF. Use the **rticmode** command to confirm any change in setting.

```
Command:      setrticmode ON
Ninja reply:  Real-Time Ionospheric Corrections Mode is ON
```

RTIC Performance Plots Using the HTTP Interface

Plots showing the Ensemble Ionosphere Delays are available on the Plots Page as shown below:



ionostat_20200301-20200302.png
ionostat_20200301-20200308.png
ionostat_20200302-20200303.png
ionostat_20200303-20200304.png
ionostat_20200304-20200305.png
ionostat_20200305-20200306.png
ionostat_20200306-20200307.png
ionostat_20200307-20200308.png

Copyright EndRun Technologies. All rights reserved.

RTIC Stability Specifications

The specifications shown here supersede the 1PPS Stability specifications in ***Appendix J - Specifications***.

1PPS Stability (RTIC Option): TDEV < 2 ns @ $\tau < 10^5$ seconds, $\sigma_y(\tau) < 4.0\text{x}10^{-14}$ @ $\tau=10^5$ secs.

System Oscillator Stability (Allan Deviation) Table:

Tau in Seconds	**MS-OCXO**	**HS-OCXO**	**US-OCXO**
1	$3.0\text{x}10^{-12}$	$1.0\text{x}10^{-12}$	$6.0\text{x}10^{-13}$
10	$3.9\text{x}10^{-12}$	$1.3\text{x}10^{-12}$	$6.0\text{x}10^{-13}$
100	$3.0\text{x}10^{-12}$	$1.7\text{x}10^{-12}$	$8.5\text{x}10^{-13}$
1000	$2.0\text{x}10^{-12}$	$1.3\text{x}10^{-12}$	$7.0\text{x}10^{-13}$
10000	$2.0\text{x}10^{-13}$	$2.0\text{x}10^{-13}$	$2.0\text{x}10^{-13}$
100000	$4.0\text{x}10^{-14}$	$4.0\text{x}10^{-14}$	$4.0\text{x}10^{-14}$

Appendix *A*

Time Figure of Merit (TFOM)

This appendix describes the Time Figure of Merit number. The TFOM number indicates the level of accuracy that should be included in the interpretation of the time-of-day and ranges from 3 to 9:

3	time error is < 100 nanoseconds
4	time error is < 1 microseconds
5	time error is < 10 microseconds
6	time error is < 100 microseconds
7	time error is < 1 milliseconds
8	time error is < 10 milliseconds
9	time error is > 10 ms, unsynchronized state if never locked to GPS

In all cases, the Ninja reports this value as accurately as possible, even during periods of GPS signal outage where the Ninja is unable to directly measure the relationship of its timing outputs to UTC. During these GPS outage periods, assuming that the Ninja had been synchronized prior to the outage, the Ninja extrapolates the expected drift of the Ninja timing signals based on its knowledge of the characteristics of the system oscillator. The extrapolated TFOM is based on a conservative estimate of the performance of the oscillator and should be considered ‘worst case’ for a typical benign ambient temperature environment.

Due to this extrapolation behavior, after initial synchronization, brief periods without GPS satellite visibility will not induce an immediate alarm condition. (Removal of the antenna to simulate this will induce an immediate alarm, however.) If the condition persists for long enough periods, you should see the TFOM character change to indicate a gradually deteriorating accuracy of the timing outputs. If the signal loss condition persists longer, then the final, unsynchronized state will eventually be reached. If the Ninja is unable to re-synchronize within one hour after reaching this state, the Alarm LED will light and the **`faultstat`** command will show a No Signal Time-Out fault.

Once the Ninja reaches the unsynchronized TFOM state, then the Network Time Protocol (NTP) daemon will report that it is running at stratum 16 and the leap indicator bits will be set to the fault state. NTP clients will recognize this and cease to use the unsynchronized server.

This page intentionally left blank.

Appendix *B*

Upgrading the Firmware

Periodically, EndRun Technologies will make bug fixes and enhancements to our products available for download from our website. All such downloads are freely available to our customers, without charge. You may securely upgrade your Ninja firmware via the network interface. Software upgrades for the Ninja are available at this link:

endruntechnologies.com/support/software-upgrades/Ninja

NOTE

The Ninja firmware consists of several different binary files. You may only need one or two of them. The revision history on our website will tell you which files need to be upgraded. The firmware image files are for the Linux RFS (root file system), the Linux Kernel and the GPS Receiver. (The GPS Receiver FPGA is rarely, if ever, upgraded.)

Upgrade via the Console Port

In order to upgrade via the network interface you will need to first download the appropriate firmware image from our website. The Ninja firmware consists of four different binary files. You may only need one or two of them. The revision history on our website will tell you which files need to be upgraded. The firmware image files are for the Linux RFS (root file system), the Linux Kernel and the GPS Receiver. The website link is shown above.

Performing the Linux RFS Upgrade

NOTE TO LINUX GEEKS

There are two FLASH disk partitions which hold the compressed Linux root file system images. These partitions are raw FLASH blocks, have no file system and may not be mounted. They are accessed through low-level device drivers. To protect the factory root file system from accidental erasure or over-writing, the upgrade utilties you will be using will only access the upgrade root file system partiton. When performing an upgrade, you will be erasing and then copying the new image to this device.

First you need to download the Linux RFS firmware from the EndRun website to a place on your network which is accessible to the Ninja. The link to the Ninja upgrade page is shown above.

CAUTION

Some browsers will automatically unzip the file when downloading from the website. Please make sure that the downloaded file size matches what the website says it should be. Upgrading the partition with a too-large file size will cause problems.

Transfer File to Ninja

You may transfer the file to your Ninja using either **`ftp`** or **`scp`**. If you are using **`ftp`**, transfer the previously downloaded file using *binary* transfer mode from the remote host to this specific file on your Ninja: */home/rootfs.gz*. The root file system image will be named with the software part number and version like: *6010-0086-000_3.00.gz*. When following the instructions below, substitute the name of the actual root file system image that you are installing for *6010-0086-000_3.00.gz*. Issue these commands from the console of your Ninja:

```
ftp remote_host                      {perform ftp login on remote host}
bin                                  {set transfer mode to binary}
get 6010-0086-000_3.00.gz /home/rootfs.gz  {transfer the file}
quit                                 {close the ftp session after transfer }
```

If you are using **`scp`**, you may open a command window on the remote computer and securely transfer the root file system image from the remote computer to your Ninja. A command like this should be used:

```
scp -p 6010-0086-000_3.00.gz root@host.your.domain:/home/rootfs.gz
```

Now issue the following command to the Ninja console to initiate the upload:

```
upgraderootfs
```

Next, update the default file system partition by issuing this command to your Ninja console:

```
updaterootflag 1
```

You should see this line displayed:

```
Default Root File System now set to: UPGRADE
```

Finally, reboot the system by issuing this command at the shell prompt:

```
reboot
```

Wait about 90 seconds for the system to shutdown and reboot. Then log in to the Ninja using **`telnet`** or **`ssh`**. If all has gone well, you should be able to log in the usual way. After you have entered your password, the system version message will be displayed. You should notice that it now indicates the software version and date of the upgrade that you previously downloaded. You can also check this at any time by issuing

```
sysversion
```

which will cause the system version message to be re-displayed.

You can also check to see which root file system image the system is currently booted under by issuing this command at the shell prompt:

```
sysrootfs
```

Which should cause this to be printed to the console:

```
BOOTED ROOT FILE SYSTEM IMAGE = 1 (Upgrade)
```

If so, and your unit seems to be operating normally, you have successfully completed the root file system upgrade. If your unit does not boot up successfully, and you are not able to **`telnet`** or **`ssh`** into the system after 90 seconds, then there has been some kind of problem with the root file system upgrade. It is possible that the file downloaded was corrupt or that you forgot to set your FTP download file mode to binary when downloading the file--either from the EndRun Technologies website or when transferring it to the Ninja.

Recovering from a Failed RFS Upgrade

To restore your Ninja to a bootable state using the factory root file system, you must use the serial I/O port and reboot the Ninja by cycling the power. Refer to ***Chapter 2 – Basic Installation, Connect the Serial I/O Port and Test the Serial I/O Port*** for setup details. When you have connected your terminal to the serial I/O port, apply power to the Ninja.

Pay close attention to the terminal window while the unit is rebooting. After the Linux bootloader displays the message

```
You can:

  Override the default kernel and/or root file system boot configuration,
  and/or
  Reset the root password to the factory password,

By typing these commands:

  bootcfg=*#     (* = 0 or 1 to select FACTORY or UPGRADE kernel,
                  # = 0 or 1 to select FACTORY or UPGRADE root file system)

  pwrst=xxxxxxxx (xxxxxxxx is reset code obtained from EndRun Tech Support)

Begin typing within 5 seconds to extend the boot timeout.
```

you must begin typing “bootcfg=00” (if you are running the FACTORY kernel) or “bootcfg=10” (if you are running the UPGRADE kernel) within five seconds to let the bootloader know that you are going to override the default root file system. Watch the rest of the boot process to make sure that you have successfully recovered. If the system boots normally, then you should resolve the problems with the previous root file system upgrade and re-perform it.

Performing the Linux Kernel Upgrade

First you need to download the Linux Kernel firmware from the EndRun website to a place on your network which is accessible to the Ninja. The link to the Ninja upgrade page is shown above.

Transfer File to Ninja

You may transfer the file to your Ninja using either **ftp** or **scp**. If you are using **ftp**, transfer the previously downloaded file using *binary* transfer mode from the remote host to this specific file on your Ninja: */tmp/kernel.gz*. The kernel image will be named with a software part number like: *6010-0087-000_2.00.gz*. When following the instructions below, substitute the name of the actual kernel image that you are installing for *6010-0087-000_2.00.gz.* Issue these commands from the console of your Ninja:

```
ftp remote_host                        {perform ftp login on remote host}
bin                                    {set transfer mode to binary}
get 6010-0087-000_2.00.gz /tmp/kernel.gz   {transfer the file}
quit                                   {close the ftp session after transfer }
```

If you are using **scp**, you may open a command window on the remote computer and securely transfer the kernel image from the remote computer to your Ninja. A command like this should be used:

```
scp -p 6010-0087-000_2.00.gz root@host.your.domain:/tmp/kernel.gz
```

Now issue the following command to the Ninja console to initiate the upload:

```
upgradekernel
```

Next, update the default file system partition by issuing this command to your Ninja console:

```
updatekernelflag 1
```

You should see this line displayed:

```
Default Kernel now set to: UPGRADE
```

Finally, reboot the system by issuing this command at the shell prompt:

```
reboot
```

Wait about 90 seconds for the system to shutdown and reboot. Then log in to the Ninja using **telnet** or **ssh**. If all has gone well, you should be able to log in the usual way. You can check the running kernel version at any time by issuing

```
kernelversion
```

which will cause the kernel version message to be displayed.

You can also check to see which kernel image the system is currently booted under by issuing this command at the shell prompt:

```
syskernel
```

Which should cause this to be printed to the console:

```
BOOTED KERNEL IMAGE = 1 (Upgrade)
```

If so, and your unit seems to be operating normally, you have successfully completed the kernel upgrade. If your unit does not boot up successfully, and you are not able to **telnet** or **ssh** into the system after 90 seconds, then there has been some kind of problem with the kernel upgrade. It is possible that the file downloaded was corrupt or that you forgot to set your **ftp** download file mode to binary when downloading the file--either from the EndRun Technologies website or when transferring it to the Ninja.

Recovering from a Failed Kernel Upgrade

To restore your Ninja to a bootable state using the factory kernel, you must use the serial I/O port and reboot the Ninja by cycling the power. Refer to ***Chapter 2 – Basic Installation, Connect the Serial I/O Port and Test the Serial I/O Port*** for setup details. When you have connected your terminal to the serial I/O port, apply power to the Ninja.

Pay close attention to the terminal window while the unit is rebooting. After the Linux bootloader displays the message

```
You can:

  Override the default kernel and/or root file system boot configuration,
  and/or
  Reset the root password to the factory password,

By typing these commands:

  bootcfg=*#     (* = 0 or 1 to select FACTORY or UPGRADE kernel,
                  # = 0 or 1 to select FACTORY or UPGRADE root file system)

  pwrst=xxxxxxxx (xxxxxxxx is reset code obtained from EndRun Tech Support)

Begin typing within 5 seconds to extend the boot timeout.
```

you must begin typing “bootcfg=00” (if you are running the FACTORY root file system) or “bootcfg=01” (if you are running the UPGRADE root file system) within five seconds to let the bootloader know that you are going to override the default kernel. Watch the rest of the boot process to make sure that you have successfully recovered. If the system boots normally, then you should resolve the problems with the previous kernel upgrade and re-perform it.

Performing the GPS Receiver Upgrade

This section has instructions for upgrading the GPS Receiver. First you need to download the GPS Receiver firmware from the EndRun website to a place on your network which is accessible to the Ninja. The link to the Ninja upgrade page is shown above.

You may transfer the file to your Ninja using either **ftp** or **scp**. If you are using **ftp** to perform the upgrade, transfer the previously downloaded file using *binary* transfer mode from the remote host to this specific file on your Ninja: */tmp/rcvr.bin*. The GPS Receiver image will be named with the software part number and version like: *6010-0088-000_1.04.bin*. When following the instructions below, substitute the name of the actual GPS Receiver image that you are installing for *6010-0088-000_1.04.bin*. You will be transferring the file to a temporary file, */tmp/rcvr.bin* on your Ninja.

```
ftp remote_host                {perform ftp login on remote host}
bin                            {set transfer mode to binary}
get 6010-0088-000_1.04.bin /tmp/rcvr.bin   {transfer the file}
quit                           {close the ftp session after the transfer }
```

If you are using SSH to perform the GPS Receiver upgrade, you may open another command window on the remote computer and securely transfer the GPS Receiver image to */tmp/rcvr.bin* using **scp** from the remote computer. A command like this could be used:

```
scp -p 6010-0088-000_1.04.bin root@host.your.domain:/tmp/rcvr.bin
```

Now you must kill two processes running on the Ninja that are using the same communications port that we need for performing the firmware upgrade to the GPS Receiver. To do that, issue these two commands exactly as shown (the ` character is on the same keyboard key as the ~ character):

```
kill `pidof g_keeper`
kill `pidof ntpd`
```

Now issue the following command to the Ninja console to initiate the upload:

```
upgradercvr
```

This command performs the file transfer to the GPS Receiver. You will see a file transfer progress message while it is performing the transfer. After it completes, wait about 10 seconds and issue these two commands to restart the processes we killed previously:

```
/etc/rc.d/rc.ntpd start
g_keeper
```

Wait about 10 seconds and issue this command to check the GPS Receiver version:

```
gpsversion
```

You should see a message like this:

```
F/W 6010-0088-000 Ver 1.00 - FPGA 6020-0018-000 Ver 02 - Feb 28 16:06:57 2020
```

The firmware version should match that of the binary file that you uploaded.

Problems with the GPS Receiver Upgrade

Should you have difficulties with the upgrade due to a corrupt file, power failure during upload, or other accident, do not be alarmed. Even though you may have lost the existing application program, the GPS Receiver bootloader program will remain intact. Correct any problem with the binary file and retry the upload procedure. If you are still unable to successfully perform the GPS Receiver upgrade, you should contact Customer Support at EndRun Technologies.

Performing the GPS Receiver FPGA Upgrade

This section has instructions for upgrading the Field-Programmable Gate Array (FPGA) resident on the GPS Receiver. This is rarely, if ever, upgraded.

First you need to download the FPGA image from the EndRun website to a place on your network which is accessible to the Ninja. The link to the Ninja upgrade page is shown above.

You may transfer the file to your Ninja using either **ftp** or **scp**. If you are using **ftp** to perform the upgrade, transfer the previously downloaded file using *binary* transfer mode from the remote host to this specific file on your Ninja: */tmp/rcvrfpga.rbf*. The FPGA image will be named with the image part number and version like: *6020-0018-000_02.rbf*. When following the instructions below, substitute the name of the actual FPGA image that you are installing for *6020-0018-000_02.rbf*. You will be transferring the file to a temporary file, */tmp/rcvrfpga.rbf* on your Ninja.

```
ftp remote_host                {perform ftp login on remote host}
bin                            {set transfer mode to binary}
get 6020-0018-000_02.rbf /tmp/rcvrfpga.rbf       {transfer the file}
quit                           {close the ftp session after the transfer }
```

If you are using SSH to perform the GPS Receiver upgrade, you may open another command window on the remote computer and securely transfer the FPGA image to */tmp/rcvrfpga.rbf* using **scp** from the remote computer. A command like this could be used:

```
scp -p 6020-0018-000_02.rbf root@host.your.domain:/tmp/rcvrfpga.rbf
```

Now you must kill two processes running on the Ninja that are using the same communications port that we need for performing the FPGA firmware upgrade to the GPS Receiver. To do that, issue these two commands exactly as shown (the ` character is on the same keyboard key as the ~ character):

```
kill `pidof g_keeper`
kill `pidof ntpd`
```

Now issue the following command to the Ninja console to initiate the upload:

```
upgradercvrfpga
```

This command performs the file transfer to the FPGA on the GPS Receiver. You will see a file transfer progress message while it is performing the transfer. After it completes, wait about 10 seconds and issue these two commands to restart the processes we killed previously:

```
/etc/rc.d/rc.ntpd start
g_keeper
```

Wait about 10 seconds and issue this command to check the FPGA version on the GPS Receiver:

```
gpsversion
```

You should see a message like this:

```
F/W 6010-0088-000 Ver 1.00 - FPGA 6020-0018-000 Ver 02 - Feb 28 16:06:57 2020
```

The FPGA version should match that of the binary file that you uploaded.

Appendix C

Helpful Linux Information

You do not need knowledge of Linux commands in order to operate Ninja. All commands necessary for proper operation are described in ***Chapter 3 - Console Port Control and Status****. However, the Ninja does support a subset of the standard Linux commands and utilities and it uses the* **`bash`** *shell, which is the Linux standard, full-featured shell. Very brief descriptions of some of the most useful Linux information is described in this appendix.*

Linux Users

Ninja is shipped from the factory with two users enabled. The first is the “root” user with password “endrun_1”. The root user has access to everything on the system, including the ability to perform system setup procedures.

The other user is “sysuser” with password “Praecis”. When logged in as sysuser you may check status information and view log files but you will not be able to modify any system settings or view secure files.

For security reasons, we recommend you change the default passwords using the Linux **`passwd`** command (see ***Change Password*** below).

Linux Commands

Detailed Information Is Available

A very brief description of the most helpful Linux commands and utilities is listed in this appendix. On Linux systems, the system commands are located in the directories with “bin” in their name, e.g. */usr/bin* or */sbin*. You can list the contents of those directories using the **`ls`** command to see what is installed on your Ninja. Then you can find out about those commands using the **`man`** command, which stands for “manual”. For example, to read details on the **`ps`** command type this:

```
man ps
```

A very detailed description, called a “man page”, of the **`ps`** command will be shown. To navigate in the document, press `d’ to scroll down, `b’ to scroll up, and `q’ to quit and return to the command prompt.

To search the database of man pages, use either **`apropos`** or **`whatis`**. **`apropos`** will do partial word searches, while **`whatis`** will only find matching whole words. For example to find all man pages dealing with ntp:

```
apropos ntp
```

The relevant available man pages are shown:

```
ntp []              (1)  - keygen - Create a NTP host key
ntpd []             (1)  - NTP daemon program
ntpdc []            (1)  - vendor-specific NTP query program
ntpq []             (1)  - standard NTP query program
ntpsnmpd []         (1)  - NTP SNMP MIB agent
sntp []             (1)  - standard SNTP program
```

Now you can issue **man** commands on each of these man pages to find what you are looking for.

Change Password

This command is used to change the password for the user that you are logged in as. It affects the serial port, SSH, Telnet and HTTPS.

```
passwd
```

List Active Processes

This command displays all active processes running in the system.

```
ps -e
```

NTP Monitoring and Troubleshooting

The following command displays which NTP clients are reaching the NTP daemon running on the Ninja. It will not try to look up host names.

```
ntpq -n -c mrulist
```

A useful command for querying NTP status is the following.

```
ntpq -peers
```

To query a remote time server (if the remote timeserver will accept the query) type:

```
ntpq -peers <hostname>
```

A table of information will be displayed. For details on what each of the table columns means type:

```
man ntpq
```

To see what version of the NTP daemon, ntpd, is operating type:

```
ntpd --version
```

Text Editors

There are two text editors resident on the Ninja file system: **`edit`** and **`joe`**. Each of these may be useful when needing to edit system configuration files or to view and search within system log files.

`joe` is the recommended editor for all purpose use in configuring and monitoring the Ninja. It is a full-featured editor with syntax highlighting and is also based on the Wordstar commands. It is user friendly with easy to find help for its key commands, and complete man page documentation. It is started by simply issuing the command **`joe [file-to-edit]`**, optionally with a file name to edit. It is the modern replacement for **`edit`** (see below).

`edit` is a very simple editor with Wordstar key commands that was originally developed for extremely memory-limited environments, such as floppy boot disks and embedded Linux appliances. When EndRun Technologies' first generation Linux-based embedded network time servers were introduced, they fell into this category and the **`edit`** text editor was appropriate. Now it is included on the Ninja file system for legacy reasons, since it has been the default editor for all first and second generation EndRun Technologies products. A man page for **`edit`** is resident on the system. When it is first started, and you did not give it a file name to edit on the command line, it shows a start-up screen with its command syntax, But once you have opened a file to edit, online help is not available. It is started by issuing the command **`edit [file-to-edit]`**, optionally with a file name to edit.

Change Log-In Banners

There are three different log-in banners in the Ninja - the serial port banner, the Telnet banner, and the SSH banner. You must be logged in as the "root" user in order to edit the *rc.local* file and change the log-in banners. Perform the following:

```
edit /etc/rc.d/rc.local
```

Change the banners as appropriate. After saving the file, copy it to */boot/etc* like this:

```
cp -p /etc/rc.d/rc.local /boot/etc/rc.d
```

Then reboot for your changes to take effect.

Query and Change Ethernet Port

`ethtool` is a Linux utility that allows you to query or change the settings for Port 0 (**`eth0`**). For example, to view current settings issue the following command:

```
ethtool eth0
```

Here is an example of one way to set the speed on Port 0 to 100Base-T:

```
ethtool -s eth0 speed 100 duplex full autoneg off
```

The command above will immediately change the port speed to 100Base-T, but it will revert to its factory (10/100Base-T) at a system reset. If you want to retain the setting after a system reset, then you need to edit the *rc.M* configuration file. Follow this sequence:

1. Edit */etc/rc.d/rc.M* using one of the editors on the previous page.
Insert the desired **ethtool** line (see example above) after the Gatekeeper Daemon is started and before the Precision Time Protocol is started. Exit and save the *rc.M* file.

2. Now you need to copy the *rc.M* file into a location that will ensure your changes persist through a system reset. Copy */etc/rc.d/rc.M* to */boot/etc/rc.d* as shown:

```
cp /etc/rc.d/rc.M /boot/etc/rc.d
```

For more details on **ethtool** and how to use it type:

```
man ethtool
```

Redirect Syslog Files to Remote Host

You can redirect syslog files to a remote host (syslog server) by adding the standard Linux redirect commands to the Ninja's *syslog.conf* file. Follow this sequence:

1. Edit */etc/syslog.conf* using one of the editors on the previous page. Insert this line:

```
*.* @remote_host
```

Substitute the actual name or IP address of your remote syslog server for "remote_host". The most common log file to be directed to the Syslog Server is the *messages.log* file which contains authenticated user login activity. If you would like to only redirect this log info to the remote host, insert this line instead of the one above:

```
messages.log @remote_host
```

Exit and save the *syslog.conf* file.

2. Now you need to copy the *syslog.conf* file into a location that will ensure your changes persist through a system reset. Copy */etc/syslog.conf* to */boot/etc/syslog.conf* as shown:

```
cp /etc/syslog.conf /boot/etc/syslog.conf
```

Appendix *D*

Third-Party Software

Your Ninja is running several different software products created and/or maintained by open source projects. Open source software comes with its own license. These are printed out for your information below.

GNU General Public License

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

Copyright © 2007 Free Software Foundation, Inc. (fsf.org)

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

NTP Software License

Information about the NTP Project can be found at www.ntp.org. The distribution and usage of the NTP software is allowed, as long as the following copyright notice is included in our documentation. For more information see: opensource.org/licenses/ntp-license.php

```
***************************************************************************
*                                                                         *
* Copyright (c) University of Delaware 1992-2015                          *
*                                                                         *
* Permission to use, copy, modify, and distribute this software and       *
* its documentation for any purpose with or without fee is hereby         *
* granted, provided that the above copyright notice appears in all        *
* copies and that both the copyright notice and this permission           *
* notice appear in supporting documentation, and that the name            *
* University of Delaware not be used in advertising or publicity          *
* pertaining to distribution of the software without specific,            *
* written prior permission. The University of Delaware makes no           *
* representations about the suitability this software for any             *
* purpose. It is provided “as is” without express or implied              *
* warranty.                                                               *
*                                                                         *
***************************************************************************
```

```
************************************************************************
*                                                                      *
* Copyright (c) Network Time Foundation 2011-2015                      *
*                                                                      *
* All Rights Reserved                                                  *
*                                                                      *
* Redistribution and use in source and binary forms, with or without   *
* modification, are permitted provided that the following conditions   *
* are met:                                                             *
* 1. Redistributions of source code must retain the above copyright    *
*    notice, this list of conditions and the following disclaimer.     *
* 2. Redistributions in binary form must reproduce the above           *
*    copyright notice, this list of conditions and the following       *
*    disclaimer in the documentation and/or other materials provided   *
*    with the distribution.                                            *
*                                                                      *
* THIS SOFTWARE IS PROVIDED BY THE AUTHORS ``AS IS'' AND ANY EXPRESS   *
* OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED    *
* WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR           *
* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR             *
* CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,         *
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT         *
* NOT LIMITED TO, PROCUREMENT  OF SUBSTITUTE GOODS OR SERVICES;        *
* LOSS OF USE, DATA, OR PROFITS; OR  BUSINESS INTERRUPTION) HOWEVER    *
* CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,          *
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)        *
* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF          *
* ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.                           *
************************************************************************
```

Hiawatha Software License

The Hiawatha webserver as implemented in the Ninja is distributed under the GPL license version 2:

hiawatha-webserver.org/license

Linuxptp Software License

The Linuxptp as implemented in the Ninja is distributed under the GPL license version 2:

sourceforge.net/directory/license:gpl

This page intentionally left blank.

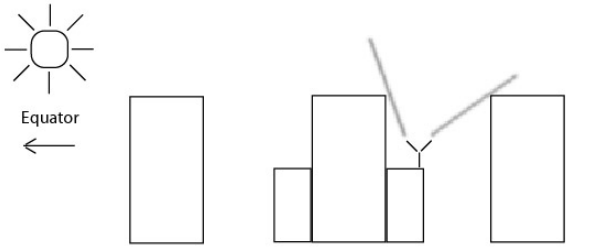
Appendix *E*

Installing the GPS Antenna

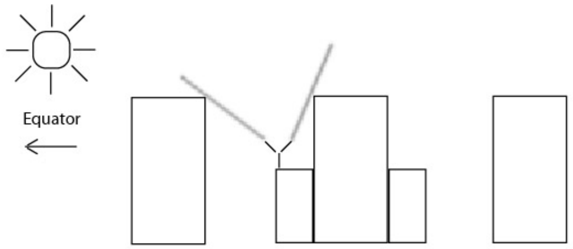
*This appendix contains guidelines for installing the GPS antenna. Mounting the antenna with a good view-of-the-sky is very important. A rooftop installation is usually best. The accuracy & stability specifications in **Appendix J - Specifications** assume an antenna installation with a very good view-of-the-sky.*

Antenna Location

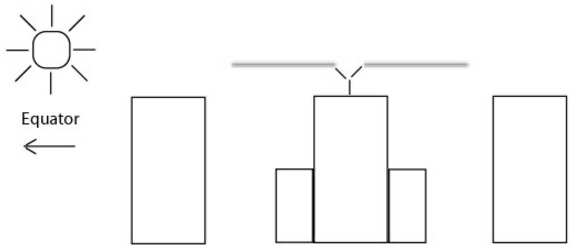
The location of the antenna must be chosen based on having as clear a view of the sky as possible. Any obstructions such as buildings, large metal objects or other antennas, and even trees, will limit the performance of the GPS antenna. The antenna should also be located away from overhead power lines or circuits, for safety reasons. The illustration below shows some examples of good and bad antenna mounting locations.



POOR ANTENNA LOCATION - blocked view of the sky plus it is on the side of the building facing away from the equator.



BETTER ANTENNA LOCATION - blocked view of the sky but it is on the side of the building facing the equator.



BEST ANTENNA LOCATION - full view of the sky.

GPS Antenna Kit

The Ninja is available with an optional GPS Antenna Kit which includes 50 feet (15 meters) of antenna cable. This amount of cable is sufficient for the majority of GPS antenna installations. Longer cable runs can also be accommodated. Figure 1 shows the GPS antenna rooftop mounting hardware. Below is a list of the items in a typical GPS Antenna Kit, part number 0610-0009-001:

- GPS Antenna (part #0502-0013-000)
- Antenna Mounting Adaptor (part #0602-0035-000)
- Aluminum Mounting Pipe (part #0602-0060-018)
- Hose Clamps (part #0100-0008-000)
- 50 feet of RG-59U (Belden 9104) Cable/TNC Male (part #0600-0013-050)
- A GPS Rooftop Installation Guide (part #5050-0017-000).

** NOTE: If you will not be using the EndRun-supplied cable, then please read the section below called ***Recommended Cable***.

About Coax Cable

The GPS signal frequency is considered to be in the microwave range and is highly affected by impedance mismatches and discontinuities in the transmission cables. All RF coax cables have a minimum bend radius. In order to prevent damage, cable should not be bent into tight curves. It is critically important during installation that kinks are not allowed to form in the cable. If RF coax cable is bent beyond its minimum bend radius. then damage to the inner construction of the cable may result. This can lead to much higher levels of loss and a non-functioning GPS receiver.

Similarly, care should be taken to ensure that the cable is not crushed, or likely to be crushed later. If the RF coax cable does suffer this kind of damage, then the dimensions of the cable will be changed and it will not maintain its characteristic impedance. Again, this can result in a non-functioning GPS receiver.

IMPORTANT

Care must be taken when installing the GPS cable. Do not treat it like a power extension cord or garden hose. Please:

Do not allow kinks to form in the cable at any time.
Ensure that the cable is not crushed at any time.
Do not bend the cable into tight curves.

Ignoring these precautions may damage the cable and cause a myriad of GPS reception problems.

Long Cable Runs

Most devices are installed with only 50 feet (15 meters) of antenna cable. However, there are many circumstances where 50 feet is inadequate. We can accommodate cable lengths up to 1000 feet using a combination of low-loss cable and preamplifiers.

Recommended Cable

The factory-supplied GPS cable is an RG-59 type. RG-59 is a broad classification, with wide variation in performance between cables from different manufacturers and for different applications. We supply two specific cables: Belden 9104 or Belden 1505A. Belden 9104 is constructed with a copper-plated steel center conductor and an aluminum outer braid. Belden 1505A is constructed of all solid copper conductors. Both cables are double shielded, low-loss cables designed for the cable TV industry, and they both have a signal loss of 10 dB/100 feet at GPS frequencies. The Belden 1505A also has very low DC resistance, which is important for long cable runs. For very long cables, if the DC resistance is too high, not enough voltage will be available at the end farthest from the Ninja where the antenna and preamplifiers are installed. For cable lengths less than 700 feet, Belden 9104 is acceptable.

IMPORTANT

If you are supplying the cable for your GPS installation, then you must make sure the cable you install is comparable to the Belden 9104 or Belden 1505A. Specifically, the cable must:

Be double-shielded
Have 10 dB or less of loss per 100 feet at 1.5 GHz.
Have very low DC resistance (for cable lengths > 700 feet)

<u>Choosing an inferior cable type can cause a myriad of GPS reception problems.</u>

Using GPS Low-Noise Amplifiers (LNAs)

For longer cable lengths, you will need one or more LNAs (see chart below). We produce the G-LNA2 which is a very high-performance, low-noise, low-power drain, inline amplifier for difficult GPS signal environments and long cable runs (greater than 250 feet of factory-supplied cable). The following table shows the number of LNAs we recommend for each GPS antenna installation using our factory-supplied cable. Installations using other cable types may have different preamplifer requirements.

Cable Length	Cable Type	Number of LNAs
Up to 250 feet (76 meters)	Belden 9104 or equivalent	0
251 to 500 feet (77 to 152 meters)	Belden 9104 or equivalent	1
501 to 700 feet (153 to 213 meters)	Belden 9104 or equivalent	2
701 to 750 feet (214 to 228 meters)	Belden 1505A or equivalent	2
751 to 1000 feet (229 to 305 meters)	Belden 1505A or equivalent	3

An Installation Guide for installing a rooftop-mounted antenna and a GPS preamplifer is shown at the end of this appendix in Figures 1 and 2.

Using Two or Three LNAs

Installation for one LNA is simple. But the physical layout of two or three LNAs is critical. An improper installation can cause feedback from the output of the last LNA, through the cable shield and back up to the antenna. This highlights the importance of properly constructed cable terminations and double shielded cable.

IMPORTANT

Proper installation of two or three LNAs is critical:

Use the appropriate cable as specified in the previous section ***Long Cable Runs***.
Ensure properly constructed cable terminations.
LNAs must be installed in a straight line down from the bottom of the antenna (no bends or loops).

An improper installation may create problems for your GPS receiver and any antennas nearby.

For installations using three preamps, we recommend that the last pre-amp be located as far as is practical from the antenna. This is because the antenna and three preamplifers will have more than 100 dB of gain, increasing the likelihood that enough leakage from the cable can cause “round-the-world” feedback to the antenna and set up oscillation. Here is the suggested configuration for an antenna installation with two or three preamplifiers:

Two LNAs

GPS Antenna
One-foot cable
LNA
One-foot cable
LNA
Up to 750 feet (228 meters) of cable
Ninja

Three LNAs

GPS Antenna
One-foot cable
LNA
One-foot cable
LNA
Up to 1000 feet (305 meters) of cable
LNA
One-foot cable
Ninja

Other Accessories

Lightning Arrestor

A lightning arrestor helps protect your GPS installation from damage due to lightning strikes. It is designed to pass the DC voltage that is needed to power the antenna and/or preamps without degrading the GPS signal. It is installed between the antenna and the receiver where the cable enters the building, near an earth-ground. You must bond the lightning arrestor to the earth-ground.

Signal Splitters

Signal splitters are used when two GPS receivers are sharing one antenna installation. The smart GPS Splitter supplied by EndRun is a one-input, two-output device. In the normal configuration, one of the splitter RF outputs (J1) passes DC from the connected GPS Receiver through the splitter to the antenna, allowing the GPS Receiver to power both the antenna and the splitter amplifier. The other RF output (J2) is DC loaded with a 200-ohm resistor to simulate the antenna current draw.

When selecting and installing a signal splitter keep these points in mind:

1. The splitter must be DC-blocked on one leg. The GPS Receiver in EndRun's products provide +5 VDC up the coax to power the GPS antenna's built-in preamp. The two GPS receivers connected to the splitter outputs must not have their power sources connected together.

2. It is desirable that the DC-blocked leg has a DC load resistor to simulate a GPS antenna load. This way you will not get a false alarm from the GPS Receiver's antenna load sensor. However, the Ninja allows you to mask an antenna fault alarm from causing a system fault by using the **`setantfltmask`** command. See details in ***Chapter 3 - Console Port Control and Status***.

3. The signal splitter supplied by EndRun has a built-in preamplifier to compensate for signal loss through the splitter. If using a splitter other than the one supplied by EndRun you may need to compensate for splitter signal loss by using a separate GPS preamplifier.

Calibrate Your Receiver

In order for the Ninja to synchronize with maximum accuracy to UTC, the delay for the cable and all devices between the antenna and the GPS receiver input (i.e. GPS preamplifiers, signal splitters, lightning arrestors, etc.) must be compensated for. You can do this via the console port **`caldelay`** and **`setcaldelay`** commands (see ***Chapter 3 - Console Port Control & Status)***.

Calibration is used to compensate for the propagation delay between the GPS antenna and the Ninja GPS receiver input connector. Positive values remove delay by advancing Ninja's 1 PPS on-time reference by the specified number of nanoseconds. Negative values add delay by retarding the 1 PPS and are used in special circumstances. The calibration value is determined by summing all the delays.

The calibration range is ±500,000 nanoseconds. The default value as shipped from the factory is 0.

The table below lists nominal propagation delays for the GPS cable and accessories supplied by EndRun Technologies. For the most demanding timing applications, it is recommended that the delay between the antenna and Ninja receiver input be precisely measured.

EndRun Part #	Description	Nominal Delay	Notes
	Antenna Cable		
0610-0009-001	Kit with 50' (15m) cable	62 nanoseconds	Belden 9104, 1.24 ns/foot
0600-0013-050	50' (15m) cable	62 nanoseconds	Belden 9104, 1.24 ns/foot
0600-0013-100	100' (30m) cable	124 nanoseconds	Belden 9104, 1.24 ns/foot
0600-0013-150	150' (46m) cable	186 nanoseconds	Belden 9104, 1.24 ns/foot
0600-0013-200	200' (61m) cable	248 nanoseconds	Belden 9104, 1.24 ns/foot
0600-0013-250	250' (76m) cable	310 nanoseconds	Belden 9104, 1.24 ns/foot
0600-0060-800	800' (244m) cable	992 nanoseconds	Belden 1505A, 1.24 ns/foot
0600-0060-A00	1000' (304m) cable	1240 nanoseconds	Belden 1505A, 1.24 ns/foot
	GPS Low-Noise Amplifier		
3509-0001-000	G-LNA2	20 nanoseconds	See device label for exact delay.
4011-0002-000	G-LNA2 Kit (with 1' cable)	21 nanoseconds	See device label for exact delay and add 1.24 ns for the 1' cable.
0502-0009-000	**Lightning Arrestor**	<1 nanosecond	
0502-0011-000	**GPS Signal Splitter**	<1 nanosecond	
	Fiber Optic Link		
3430-0003-000 3430-0004-000 3430-0005-000	Fiber Optic Receiver Fiber Optic Transmitter Fiber Optic Transmitter	17 nanoseconds per Receiver/Transmitter pair.	Add the delay of single mode fiber optic cable which is typically 1.4/1.5 ns/foot. See cable specification.

Mounting On A Rooftop

Mounting your GPS antenna with an unobstructed view of the sky (usually on a rooftop) is the recommended installation. Please follow these guidelines to eliminate exposure to electrical service wiring and to minimize the potential for lightning strikes.

⚠

Installations subject to lightning strikes should use a lightning arrestor installed at the building entrance. A lightning arrestor suited for this purpose is available through EndRun Technologies. The arrestor must be installed according to the manufacturer's instructions.

⚠

Do NOT route the antenna wiring near or with AC wiring (Class 1 circuits per the NEC). Do NOT mount the antenna wiring where it may become energized by nearby AC wiring or components should they fall.

Obtaining A Reference Position

Your Ninja is capable of operation from either an automatically determined GPS reference position or a manually entered GPS reference position. If you need to provide a reference position to your Ninja, it is best to use a previously determined position from the unit itself or a highly accurate surveyed position.

About WGS-84 Height

Internally, GPS receivers report latitude, longitude and height above the WGS-84 ellipsoid. However, for a lot of reasons, WGS-84 is not the way that mapmakers and surveyers report the height. That means, in order to use the height information as reported by Ninja, you need to do a conversion. One easy way to do the conversion is by going to this link:

unavco.org/software/geodetic-utilities/geoid-height-calculator/geoid-height-calculator.html

After entering your latitude and longitude, this website will give you a report showing the GPS ellipsoidal height, the Geoid height, and the Orthometric height. The Orthometric height is the one most people are familiar with, which is height above mean sea level. However, GPS receivers use the GPS ellipsoidal height. Below is a sample report:

GPS ellipsoidal height = 0 (meters)
Geoid height = -31.023 (meters)
Orthometric height (height above mean sea level) = 31.023 (meters)

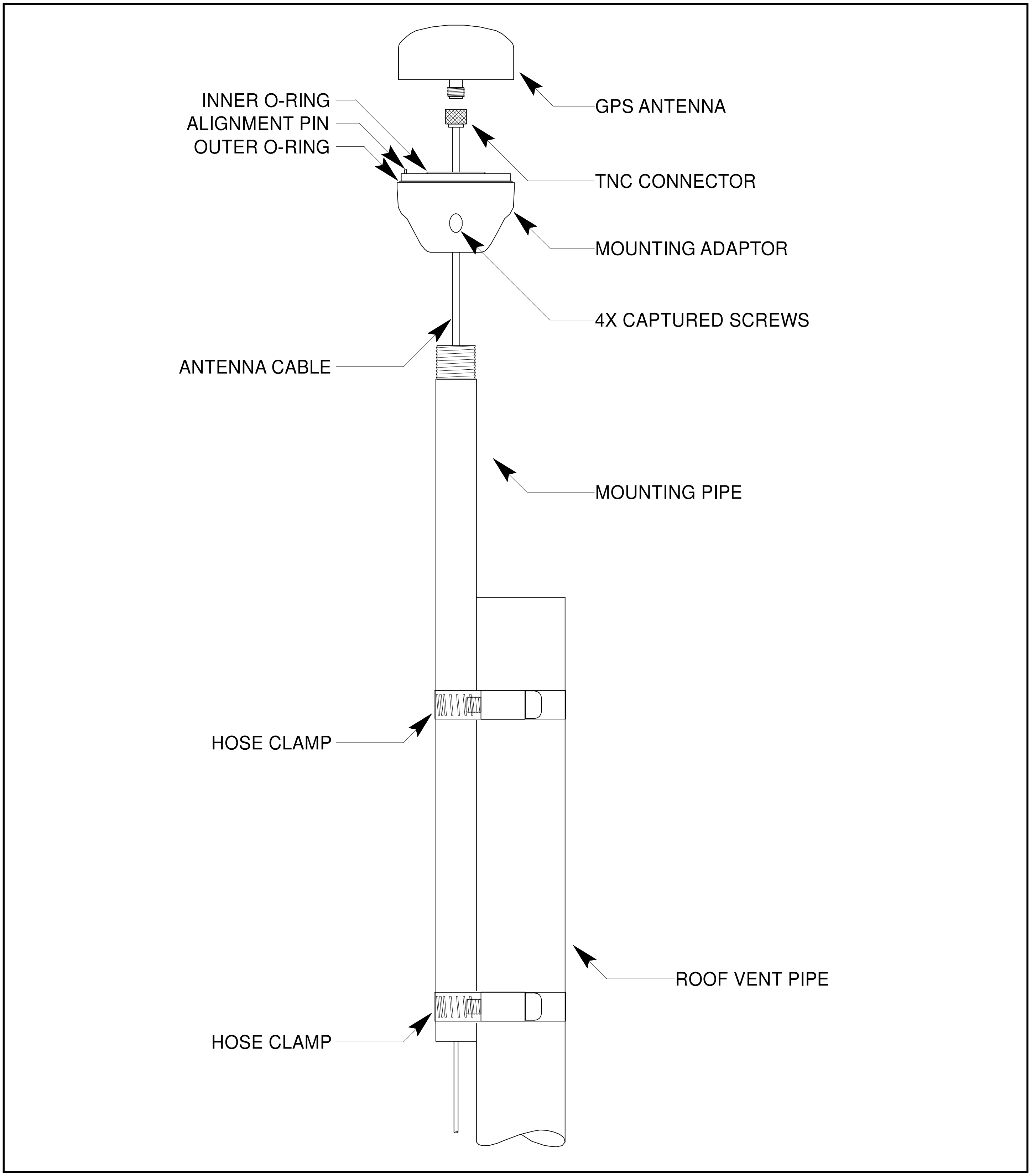


FIGURE 1 - GPS ANTENNA ROOFTOP MOUNTING HARDWARE

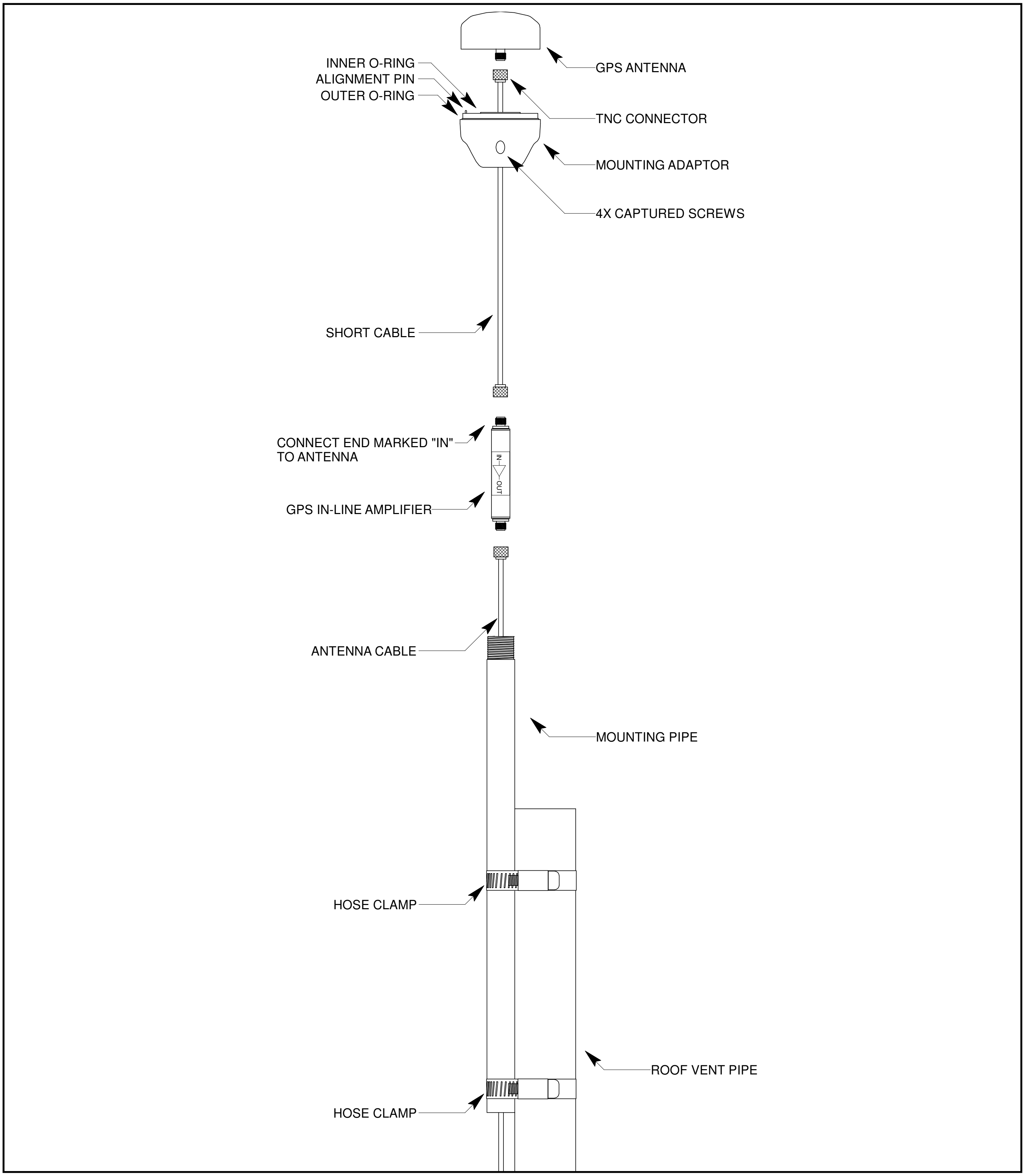


FIGURE 2 - GPS ANTENNA ROOFTOP MOUNTING HARDWARE WITH PREAMPLIFIER

This page intentionally left blank.

Appendix *F*

Leap Seconds

UTC stands for Coordinated Universal Time. UTC is the international time standard most commonly used in the world and by the Network Time Protocol (NTP). A leap second insertion is scheduled about every two to three years in order to keep UTC in alignment with the Earth's rotation. Possible leap second insertions can be scheduled after 23:59:59 UTC on June 30 or December 31.

Automatic Leap Second Insertion

Your GPS-synchronized Ninja precisely adjusts for leap seconds if and when they occur. There is nothing you need to do in order to keep your Ninja accurately synchronized to UTC.

You can see the current GPS-UTC parameters that are downloaded from the satellites by using the **`gpsutcinfo`** command. See ***Chapter 3 - Console Port Control and Status*** for details on this command or type **`help gpsutcinfo`** at the console port.

Background Information

Leap seconds are inserted from time-to-time in order to keep UTC, which is derived from atomic time (TAI), in agreement with the Earth's rotation rate. Relative to TAI, the Earth's rotation rate is slowing down. This means that UTC must be retarded periodically in order to maintain agreement between UTC and the apparent daylength. If this were not done, eventually UTC would drift out-of-sync with Earth's day and many astronomical and navigational problems would ensue.

The International Earth Rotation and Reference Systems Service (IERS) is the organization responsible for measuring the relationship between UTC and the rotation rate of the Earth. When the difference between UTC and apparent Earth time has exceeded a certain threshold, the IERS coordinates with the Bureau International of the Hour (BIH) to schedule the insertion of a leap second into the UTC time scale. The IERS publishes Bulletin C about 6 months in advance of each possible leap second insertion point. Bulletin C confirms whether a leap second will or will not be inserted at the next possible insertion point. The IERS website is:

iers.org

EndRun summarizes this information here:

endruntechnologies.com/support/leap-seconds

This page intentionally left blank.

Appendix *G*

System Faults

The status of the Ninja is constantly monitored and a fault will occur when any of several parameters is out of spec. When this happens the Alarm LED on the front panel will light. This appendix defines the various faults.

Overview

The Alarm LED will light when a fault has occurred. You can see which fault is the problem by using the **`faultstat`** command.

Masking Faults

Some faults can be masked. These are the ANT (GPS Antenna) and SIG (GPS Signal) faults. When masked, these faults will not cause an alarm. You may want to mask the ANT fault if you are using a GPS splitter. You may want to mask the SIG fault if you are operating your Ninja as a Stratum 2 NTP Server and are not using a GPS signal. For information on Stratum 2 see ***Chapter 7 - NTP, Configuring the Ninja as a Stratum 2 Server***.

To mask a fault you can use console port commands **`setantfltmask`** and **`setsigfltmask`**. For more information see ***Chapter 3 - Console Port Control and Status*** or type **`help setsigfltmask`** and **`help setantfltmask`** on the console.

Fault Definitions

System Oscillator DAC (DAC)

This fault indicates that the electronic frequency control DAC for the Ninja system oscillator has reached either the high or low alarm limit while locked to the GPS signal. Unless the unit is being subjected to out-of-specification environmental conditions, this would indicate that the oscillator frequency has drifted near to the end of life region. This should normally only occur after at least ten years of operation. The unit will continue to function until the oscillator frequency finally reaches one of the actual DAC endpoints. The unit should be returned to the factory for oscillator replacement at your convenience.

GPS Signal (SIG)

This fault indicates that the unit has not been able to acquire a GPS signal for one hour while the Time Figure of Merit has been 9, the unsynchronized condition. This could be due to a variety of reasons. If there are no other faults that could explain the inability to receive a signal, then there could be an antenna failure or blockage. If the condition persists indefinitely, and a problem with the antenna is not evident, then please contact EndRun Customer Support.

GPS Receiver FPGA Configuration (FPGA)
This fault indicates that the GPS Receiver is unable to configure its FPGA. This is a fatal fault. Please contact EndRun Customer Support.

GPS Receiver FLASH Writes (FLSH)
This fault indicates that the GPS Receiver is unable to verify a write to its FLASH non-volatile parameter storage area. This should not ever occur under normal operation.. Please contact EndRun Customer Support.

Synthesizer Limits (SYN1)
This fault indicates that the GPS Receiver front-end RF local oscillator synthesizer has reached the alarm limit. This condition should not normally occur unless the unit is subjected to out-of-specification environmental conditions. Otherwise, this is a fatal fault. Please contact EndRun Customer Support.

Synthesizer (SYN2)
This fault indicates that the GPS Receiver front-end RF local oscillator synthesizer has failed. This condition should not normally occur unless the unit is subjected to out-of-specification environmental conditions. Otherwise, this is a fatal fault. Please contact EndRun Customer Support.

GPS Reference Time (REF)
This fault indicates that the GPS Receiver received an erroneous time input from the Global Positioning System (GPS). If the condition persists please contact EndRun Customer Support.

GPS Receiver Oscillator (OSC)
This fault indicates that the GPS Receiver main oscillator has failed. This is a fatal fault. Please contact EndRun Customer Support.

Antenna Short (SHRT)
This fault indicates that the GPS antenna has an overcurrent condition (short).

Antenna Open (OPEN)
This fault indicates that the GPS antenna has an undercurrent condition (open).

GPS Receiver Oscillator Phase-Lock-Loop (PLL)
This fault indicates that there is an unlock condition between the GPS Receiver main oscillator and the Ninja system oscillator. This is a fatal fault. Please contact EndRun Customer Support.

NTP Polling (POLL)
This fault indicates that the GPS Receiver is not receiving polling requests from the NTP daemon. This could be due to a hardware or software failure. If the condition persists please contact EndRun Customer Support.

GPS Communication (COM)
This fault indicates that the Linux Subsystem is unable to establish communications with the GPS Receiver.

Appendix *H*

Time Code Formats

An optional feature of your Ninja is one amplitude-modulated time code output, or up to three DC-shift time code outputs (on PPO connectors). Time codes are commonly used to provide time information to external devices such as displays, magnetic tape devices, strip chart recorders and several types of embedded computer peripheral cards. The output code format is selectable via a console command. See ***Chapter 9 - Inputs/Outputs, IRIG-AM Option****. Each format is described below.*

IRIG-B122/002

This is the most widely used format and is normally the factory default. The IRIG-B122 format is a 100 bps code and is used to amplitude modulate a 1000 kHz sine wave carrier. The information contained in the time code is seconds through day-of-year coded in Binary Coded Decimal (BCD). Reference IRIG Document 104-60.

IRIG-B123/003

In addition to the time information identified in B122 above, this format also contains Straight Binary Seconds (SBS) of day. SBS is provided at the end of the frame, in the 17 bits starting in position 80.

IRIG-B120/000 (IEEE-Standard 1344-1995)

In addition to the time data and the Straight Binary Seconds data this format provides for time/status data in the control bit positions of IRIG-B. The information provided there is defined by IEEE standard 1344-1995: Unit and Tens of Years, Leap Second, DaylightSaving Time, Local Time Offset, Time Quality and Parity. The IEEE-1344 table provided below shows each bit position with detailed information.

NASA-36

NASA-36 bit time code is a 100-bit, pulse width modulated format used to amplitude modulate a 1000 kHz sine wave carrier. The information contained in the time code is seconds, minutes, hours and days. The format is used by several military ranges. Reference IRIG Document 104-59.

2137

The 2137 code is a 25-bit pulse width modulated format used to amplitude modulate a 1000 kHz sine wave carrier. The information contained in the time code is seconds, minutes and hours. The format is used by certain security organizations.

IEEE-1344 Bit Definition

Bit Position	Bit Definition	Explanation
P50	Year, BCD1	Unit years
P51	Year, BCD2	
P52	Year, BCD4	
P53	Year, BCD8	
P54	Not used	
P55	Year, BCD10	Tens years
P56	Year, BCD20	
P57	Year, BCD40	
P58	Year, BCD80	
P59	P6	Position identifier
P60	Leap second pending	Set to one, 59 seconds prior to leap insertion
P61	Leap second	0 = add second, 1 = delete second
P62	DaylightSaving Time pending	Set to one, 1 second prior to DST change
P63	DaylightSaving Time	1 = DST active
P64	Local offset sign	0 = +, 1 = -
P65	Local offset binary 1	Local offset from UTC time
P66	Local offset binary 2	
P67	Local offset binary 4	
P68	Local offset binary 8	
P69	P7	Position identifier
P70	Local offset half hour bit	0 = none, 1 = half hour time offset added
P71	Time quality binary 1	Time quality indicates clock precision.*
P72	Time quality binary 2	
P73	Time quality binary 4	
P74	Time quality binary 8	
P75	Parity	Odd parity for all preceding data bits
P76-P78	Not used	
P79	P8	Position identifier

*		
*	0	normal operation, clock locked (TFOM=3)
	4	time error is < 1 us
	5	time error is < 10 us
	6	time error is < 100 us
	7	time error is < 1 ms
	8	time error is < 10 ms
	9	time error is > 10 ms
	F	time not reliable, never locked to GPS

Refer to ***Appendix A - Time Figure-of-Merit*** for detailed information.

Appendix *I*

Operation with a GPS Simulator

This appendix describes several commands that are intended for use before and after operation with a GPS simulator. In addition, recommendations for the simulator setup are given so that the advanced integrity-checking and GPS week number ambiguity resolution algorithms operating in the GPS Receiver will not cause unexpected behaviors during and after running on the simulator.

Background

GPS-based timing systems supporting critical infrastructure functions could be vulnerable to malfunction due to weak signals, jamming, spoofing or accidental GPS control system errors. EndRun Technologies' GPS timing receiver technology has evolved to be highly robust against these threats. Because of this, operation with a GPS simulator requires careful attention when setting up the simulator. Careful consideration must also be given in configuring the GPS Receiver before and after simulator operation. Failure to understand and implement the information that follows could result in confusing simulation results and/or improper operation when the Ninja is subsequently reconnected to live GPS signals.

Console Port Commands

clearalmanac

The GPS Receiver maintains a record of the GPS almanac data received from the satellites in its non-volatile memory. These almanacs are used to calculate the elevations and doppler frequency offsets of GPS satellites so that they may be more quickly acquired and tracked by the signal processor. When operating on a GPS simulator, any previously stored almanacs gathered from the live GPS may not contain valid data, so they should be cleared prior to operation with a simulator. Likewise, following operation with a simulator, they should also be cleared before reconnecting to live GPS signals. The **`clearalmanac`** command provides this capability.

This command clears all GPS almanac data from both non-volatile and working memory in the GPS Receiver. (It does not clear the copy of the Yuma-formatted almanac that is retrieved with the **`dumpalmanac`** command. That data will be updated on reception of new almanac data.)

The **`clearalmanac`** command can be used at other times, but is typically used before and after operation with a GPS simulator.

SUMMARY

The **`clearalmanac`** command MUST be executed both before and after use with a GPS simulator.

```
Command:        clearalmanac
Ninja reply:    Clearing GPS Almanacs
```

resetlastgpswn

The GPS Receiver is designed to autonomously handle future rollovers of the 10-bit GPS week number received in data from the satellites. To do that, it maintains a record in non-volatile memory of the unambiguous, full GPS week number when it was last locked to the GPS system, i.e. the Last GPS Week Number. If this information is not available, then the full GPS week number corresponding to the build date of the GPS receiver firmware will be used as the Last GPS Week Number.

This Last GPS Week Number is the minimum possible full GPS week number. The Ninja will never allow the date and time to be set prior to the date and time corresponding to this Last GPS Week Number. For example, setting the simulator time to a date two weeks in the past will result in the Ninja setting the date 1022 weeks in the future.

This means that a simulator should not be set to a date in the past. Doing so will cause the time to be set a multiple of 1024 weeks in the future relative to the simulator time. If lock to the simulator is then achieved, this future time will then be saved to non-volatile memory as the Last GPS Week Number. Subsequent runs with the simulation time set in the past would cause this time to continue to advance.

Following simulator operation, it is imperative that the non-volatile Last GPS Week Number be reset to the firmware build date. This is accomplished by running the **`resetlastgpswn`** command prior to reconnecting to live GPS signals.

SUMMARY

The simulator should <u>not</u> be set to a date in the past.

You MUST disconnect the antenna or simulator from the Ninja and then execute the the **`resetlastgpswn`** command before reconnecting the Ninja to a live GPS signal.

IMPORTANT

This command should not be used unless you have reason to believe that operation with a GPS simulator has previously taken place. Arbitrary usage of this command could compromise the effectiveness of the algorithm in autonomously handling the GPS week number rollover events.

The command must be issued with one argument: **`TRUE`**

```
Command:        resetlastgpswn TRUE
Ninja reply:    Resetting Last GPS Week Number
```

resetleaphistory

The GPS Receiver performs extensive integrity/sanity checking on the data received from the GPS satellites. In particular, the current and future UTC leap second values received in the almanac data are compared to a historical record of previous UTC leap seconds and the GPS week numbers at which they were inserted into the UTC timescale. Should a simulator be set with arbitrary values for these UTC leap seconds that differ by more than a couple of seconds from the actual values currently being transmitted by the GPS, the almanac data will be rejected.

The simulator should be set with realistic UTC leap second values. If the simulator is set with incorrect UTC leap second values, yet they are close enough to pass the integrity screen, then they will be accepted and it will be necessary to run the **`resetleaphistory`** command prior to reconnecting to live GPS signals. Failure to do so would compromise the integrity checking algorithm.

Issuing this command will reset the stored GPS Leap Second History to the history that was correct at the build date of the receiver firmware. As long as the firmware build date is not too many years old, this will allow the receiver to operate properly when it receives signals from the actual GPS satellites.

SUMMARY

The simulator should be set with realistic UTC leap second values, such as those currently being transmitted by GPS. If you simulate a leap second insertion then you MUST execute the **`resetleaphistory`** command before reconnecting the receiver to a live GPS signal.

IMPORTANT

This command should not be used unless you have reason to believe that operation with a GPS simulator has previously taken place. Arbitrary usage of this command could cause important leap history events to be lost. This will compromise the effectiveness of the integrity checking algorithm.

The command must be issued with one argument: TRUE

```
Command:       resetleaphistory TRUE
Ninja reply:   Resetting Leap History
```

This page intentionally left blank.

Appendix J

Specifications

The following accuracy and stability specifications assume a stationary position (not dynamic mode), four satellite lock, and the antenna mounted with a full view-of-the-sky.

GPS Receiver:

L1 Band – 1575.42 MHz.
12 Channels, C/A Code (16 correlators).
Static mode and dynamic-platform mode (shipboard only).
15 dB minimum gain at receiver input.
Timing Receiver Autonomous Integrity Monitoring (TRAIM).
TNC connector (female) on rear panel, Z_{in} = 50Ω, 5 VDC to antenna.

Antenna:

TNC connector (female), Z_{in} = 50Ω, 5 VDC input.
Integral +40 dB gain LNA with bandpass filter for out-of-band interference rejection.
Rugged, all-weather housing capable of operation over –40°C to +85°C.
Mounting via 18” long, ¾” pipe with stainless steel clamps.
50’ low-loss RG-59 downlead cable standard.
Extension cables and low noise pre-amplifiers are available.

Time to Lock:

< 5 minutes, typical (TCXO).
< 10 minutes, typical (OCXO).

System Status LEDs:

Sync LED: Amber LED pulses to indicate GPS acquisition and lock status.
Alarm LED: Red LED indicates a fault condition.

Timing Characteristics:

Accuracy: < 25 nanoseconds RMS to UTC(USNO) when locked.*
< 10 nanoseconds RMS with calibration option.
Stability: TDEV < 10 ns, $\tau < 10^5$ seconds, $\sigma_y(\tau) < 6.0\text{x}10^{-14}$ @ $\tau=10^5$ secs.
TDEV < 2 ns, $\tau < 10^5$ seconds, $\sigma_y(\tau) < 4.0\text{x}10^{-14}$ @ $\tau=10^5$ secs with the RTIC Option.
See ***Chapter 12 - Real-Time Ionospheric Corrections, Specifications.***
User-Calibration: +/- 500 us, 1 ns resolution.
*See GPS-UTC Timing Specifications for details.

Platform:

Operating System Kernel Version: 4.14.88
Slackware Linux Distribution Version: 14.2
Processor: 500 MHz
RAM: 128MB
FLASH: 4GB

NTP Server Performance and Synchronization Accuracy:

NTP Timestamp Accuracy: <10 microseconds @ 2500 requests/second.

Supported IPv4 Protocols:

SNTP, NTP v2, v3, v4, SHA/MD5 authentication, broadcast/multicast and autokey
SSH client and server with “secure copy” utility, SCP
SNMP v1, v2c, v3 with Enterprise MIB
TIME and DAYTIME server
TELNET client/server
FTP client
DHCP client
SYSLOG
HTTPS
PTP/IEEE-1588 (option)
SyncE (option)

Supported IPv6 Protocols:

SNTP, NTP v2, v3, v4, SHA/MD5 authentication, broadcast/multicast and autokey
SSH client and server with “secure copy” utility, SCP
SNMP v1, v2c, v3 with Enterprise MIB
TIME and DAYTIME server
HTTPS
Note: See ***Chapter 8 - IPv6 Information*** for more details.

PTP/IEEE-1588 Grandmaster (Option):

IEEE-1588-2008 (v2) with 8-ns hardware timestamping.
Default or IEEE-802.1AS Profile.
Transport: IPv4. Layer-2 (L2) or Layer-3 (L3).
Delay Mechanism: E2E or P2P.
Transmission Mode: Multicast or Hybrid (mixed Unicast/Multicast).
Sync Interval: 1, 2, 4, 8, 16, 32, 64 or 128 packets / 1 second.
Announce Interval: 1 packet per 1, 2, 4, 8 or 16 seconds.
PTP Timestamp Resolution: 8 nanoseconds.
PTP Timestamp Accuracy to Reference Clock: 8 nanoseconds.
Note: See ***Chapter 10 - PTP/IEEE-1588*** for more information.

SyncE (Option):

Synchronous Ethernet in Ninja meets the specifications defined in the ITU-T:
G.8261 Architecture and wander performance
G.8262 Timing characteristics
G.8264 Ethernet Synchronization Message Channel (ESMC)
Note: See ***Chapter 11 - Synchronous Ethernet*** for more information.

Network I/O:

One front-panel RJ-45 jack.
10/100Base-T Ethernet.
Two green LEDs: One to indicate activity and one to indicate speed (PTP version only).

Serial Port I/O:

Signal: I/O port at RS-232 levels for secure, local terminal access.
Parameters: 19200 baud, 8 data bits, no parity, 1 stop bit.
Connector: Front-panel DB-9M connector labeled "RS-232".
To connect to a computer, a null-modem adapter must be used. The serial cable provided with the shipment is wired as a null-modem. Pinout for the RS-232 console port is shown below.
Note: For operational details see ***Chapter 3 - Console Port Control and Status***.

Ninja DB9M Pin	Signal Name
1	Not Connected
2	Receive Data (RX)
3	Transmit Data (TX)
4	Not Connected
5	Ground
6	Not Connected
7	Not Connected
8	Not Connected
9	Not Connected

Size:

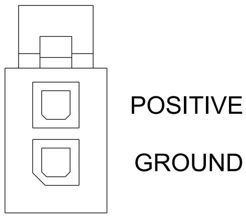
Chassis:	1.5"H x 5.3"W x 4.44"D.
Weight:	< 1 lb. (0.45 kg.)
Antenna:	3.25" H x 3" Diameter

Environmental:

Operating Temperature:	0° to +50° C
Storage Temperature:	-40° to +85° C
Antenna Operating Temperature:	-40° to +85° C
Operating Humidity:	5% to 90% RH, non-condensing.
Storage Humidity:	5% to 95% RH, non-condensing.
Maximum Operating Altitude:	13,125 ft. / 4000 meters

DC Power:

9-18 VDC, 0.9A maximum.
8W maximum.
Connector: Molex Micro-Fit 3.0 2-pin jack.
Mate: Molex 43025-0200 Housing.
Molex 43030-0008 20-24 AWG terminal (2 required).



POWER INPUT CONNECTOR

External AC Power Supply (Option):

Input: 100-240 VAC, 50-60 Hz, 0.3A Max
Output: 12 VDC, 5.0A 60W Max.
See dimensional drawing at the end of this Appendix J.

System Oscillator

One of the following is used as the System Oscillator:

Standard High-Performance TCXO (HP-TCXO):
Temp Stability: 1x10^{-6} over 0º to 70º C, Ageing Rate/Year: 1x10^{-6}.
NTP Stratum 1 Holdover: 24 hours

Optional Medium-Stability OCXO (MS-OCXO):
Temp Stability: 4x10^{-9} over 0º to 70º C, Ageing Rate/Year: 3x10^{-8}.
NTP Stratum 1 Holdover: 35 days.

Optional High-Stability OCXO (HS-OCXO):
Temp Stability: 1x10^{-9} over 0º to 70º C. Ageing Rate/Year: 3x10^{-8}.
NTP Stratum 1 Holdover: 35 days.

Optional Ultra-Stable OCXO (US-OCXO):
Temp Stability: 5x10^{-10} over 0º to 70º C. Ageing Rate/Year: 3x10^{-8}.
NTP Stratum 1 Holdover: 35 days.

System Oscillator Stability (Allan Deviation) Table (@ 10 MHz):

Tau in Seconds	**HP-TCXO***	**MS-OCXO***	**HS-OCXO***	**US-OCXO***
1	1.0x10^{-10}	3.0x10^{-12}	1.0x10^{-12}	6.0x10^{-13}
10	4.0x10^{-11}	3.9x10^{-12}	1.3x10^{-12}	6.0x10^{-13}
100	4.0x10^{-11}	3.0x10^{-12}	1.7x10^{-12}	8.5x10^{-13}
1000	4.0x10^{-12}	2.0x10^{-12}	1.5x10^{-12}	8.0x10^{-13}
10000	4.0x10^{-13}	4.0x10^{-13}	4.0x10^{-13}	4.0x10^{-13}
100000	6.0x10^{-14}	6.0x10^{-14}	6.0x10^{-14}	6.0x10^{-14}

*****For RTIC Option see ***Chapter 12 - Real-Time Ionospheric Corrections, Specifications***.

System Oscillator Phase Noise Table (dBc/Hz @ 10 MHz):

Hz	HP-TCXO (Typical)	Spurs
1	-70	
10	-100	-90
100	-130	-100
1k	-140	-110
10k	-145	-120
100k	-145	-120

Hz	MS-OCXO	HS-OCXO	US-OCXO	Spurs
1	-95	-105	-110	
10	-120	-130	-135	-120
100	-135	-140	-148	-125
1k	-145	-150	-152	-125
10k	-145	-150	-153	-120
100k	-145	-150	-153	-110

System Oscillator Phase Noise Table (dBc/Hz @ 5 MHz):

Hz	MS-OCXO	HS-OCXO	US-OCXO
1	-100	-110	-115
10	-130	-135	-140
100	-140	-145	-152
1k	-150	-155	-155
10k	-150	-155	-155
100k	-150	-155	-155

Note: Phase noise for all OCXOs is guaranteed with the Low-Phase-Noise Option.

Optional Outputs

Optional 5 MHz Low-Phase-Noise Output:

Output Level: +13 dBm, +/- 2 dBm at 50Ω.
Harmonics: < -35 dBc at 50Ω.
Channel-to-Channel Isolation: > 60 dB
Stability: See ***System Oscillator Stability (Allan Deviation) Table*** above.
Phase Noise: See ***System Oscillator Phase Noise Table*** above.
Alignment: Not aligned with other outputs in this unit.
Connector: Front-panel SMA jack labeled “A”, “B”, “C”, “D”.
Note: See ***Chapter 9 - Inputs/Outputs, Output Options*** for more information.

Optional 10 MHz Low-Phase-Noise Output:

Output Level: +13 dBm, +/- 2 dBm at 50Ω.
Harmonics: < -40 dBc at 50Ω.
Channel-to-Channel Isolation: > 60 dB
Stability: See ***System Oscillator Stability (Allan Deviation) Table*** above.
Phase Noise: See ***System Oscillator Phase Noise Table*** above.
Alignment: Not aligned with other outputs in this unit.
Connector: Front-panel SMA jack labeled “A”, “B”, “C”, “D”.
Note: See ***Chapter 9 - Inputs/Outputs, Output Options*** for more information.

Optional 1 PPS Output:

Signal: Positive TTL pulse into 50Ω.
TTL (50Ω to GND): V_{OL}(max)=0.4V, V_{OH}(min)=2.4V, V_{OH}(nom)=2.5V, V_{OH}(max)=3.0V.
Width: User selectable to 20 us, 1 ms, 100 ms, 500 ms.
User-Calibration: +/- 500 us, 1 ns resolution.
Accuracy: < 10 ns RMS to UTC(USNO) when locked.*
Stability: TDEV < 10 ns, $\tau < 10^5$ seconds, $\sigma_y(\tau) < 6.0 \times 10^{-14}$ @ $\tau = 10^5$ secs.
TDEV < 2 ns, $\tau < 10^5$ seconds, $\sigma_y(\tau) < 4.0 \times 10^{-14}$ @ $\tau = 10^5$ secs with the RTIC Option.
See ***Chapter 12 - Real-Time Ionospheric Corrections, Specifications***.
Rise Time: < 2 ns.
Alignment: Within 1 ns of the other TTL outputs in this unit.
Connector: Front-panel SMA jack labeled “E”.
Note: See ***Chapter 9 - Inputs/Outputs, Output Options*** for more information.
*See GPS-UTC Timing Specifications for details.

Optional IRIG-AM Output:

Signal: Amplitude-modulated (AM), 3:1 ratio, 1 kHz carrier.
Drive: 1 V_{RMS} into 50Ω.
Format: User selectable to IRIG-B (120/IEEE-1344/C37.118-2005, 122, 123), NASA-36, 2137.
Alignment: Within 5 us of 1 PPS.
Connector: Front-panel SMA jack labeled “F”.
Note: See ***Chapter 9 - Inputs/Outputs, Output Options*** for more information.

Optional Programmable Pulse Output (PPO):

User-selectable Output Type: On-time pulse rate or a digital time code or a trigger pulse output.
Signal: Positive TTL pulse into 50Ω.
TTL (50Ω to GND): V_{OL}(max)=0.4V, V_{OH}(min)=2.4V, V_{OH}(nom)=2.5V, V_{OH}(max)=3.0V.
Accuracy: < 10^{-13} to UTC for 24-hour averaging times when locked.
Stability: TDEV < 10 ns, $\tau < 10^5$ seconds, $\sigma_y(\tau) < 6.0 \times 10^{-14}$ @ $\tau = 10^5$ secs.
TDEV < 2 ns, $\tau < 10^5$ seconds, $\sigma_y(\tau) < 4.0 \times 10^{-14}$ @ $\tau = 10^5$ secs with the RTIC Option.
See ***Chapter 12 - Real-Time Ionospheric Corrections, Specifications***.
Rise Time: < 2 ns (TTL).
Alignment: Within 1 ns of the other TTL outputs in this unit.
On-Time Pulse Rates:
Rate: User selectable to 1, 10, 100, 1k, 10k, 100k, 1M, 5M, 10M PPS, 1PPM, 1PP2S.
Duty Cycle: 50% except 1PPS which mimics the 1PPS Output defined above.
TriggerPPO: This function allows you to program the time for a pulse to occur.
Digital Time Code:
Format: User selectable to IRIG-B (000/IEEE-1344/C37.118-2005 compliant, 002, 003).
Connector: Front-panel SMA jack labeled “G”, “H”, “I”.
Note: See ***Chapter 9 - Inputs/Outputs, Output Options*** for more information.
Note: Specifications for the Trigger PPO function are also in ***Chapter 9***.

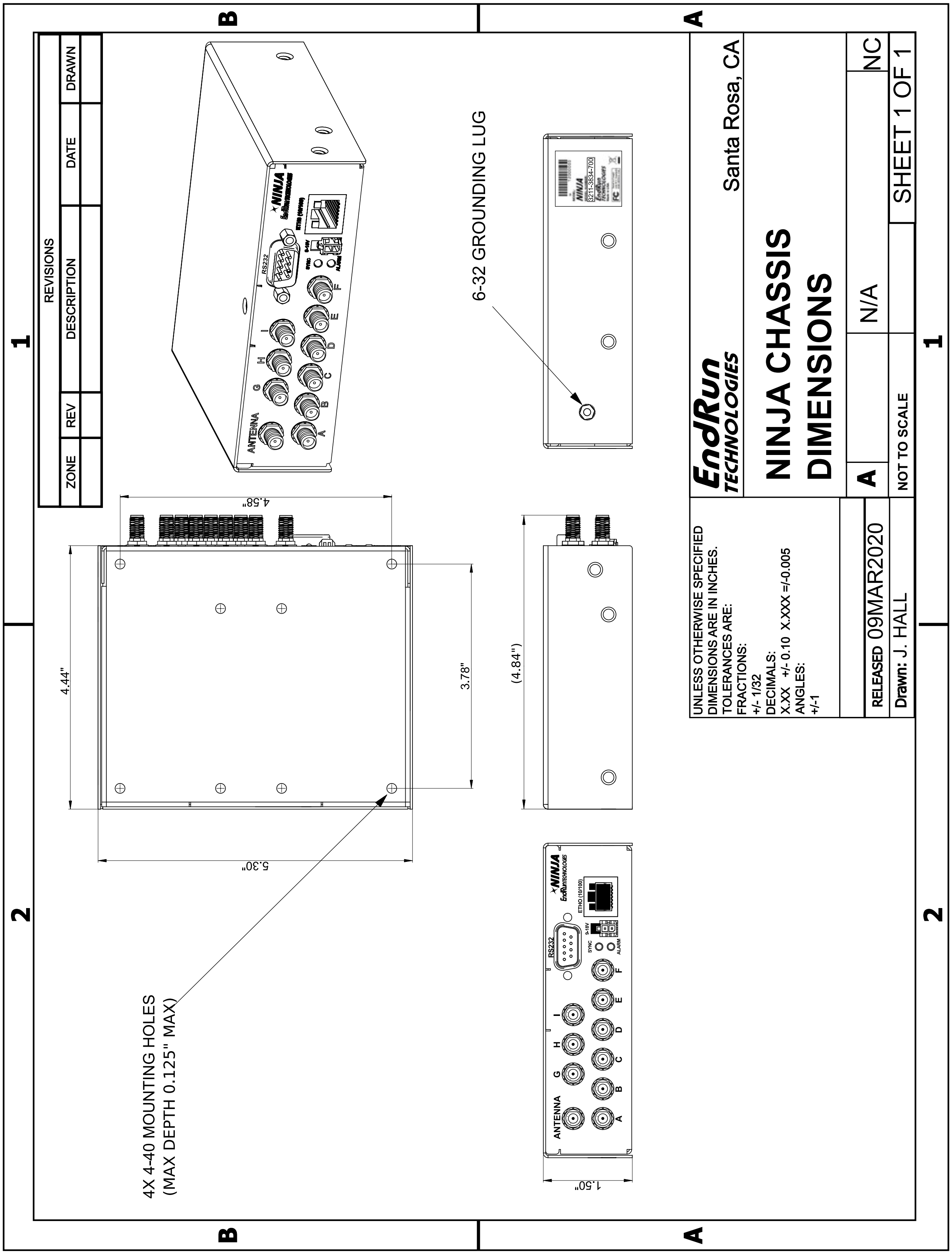
Optional Alarm Output:

Signal: MMBT2222A open collector, grounded emitter. High impedance in alarm state.
Voltage: 40 VDC, maximum.
Saturation Current: 100 mA, maximum.
Connector: Front-panel SMA jack labeled “I”.
Note: See ***Chapter 9 - Inputs/Outputs, Output Options*** for more information.

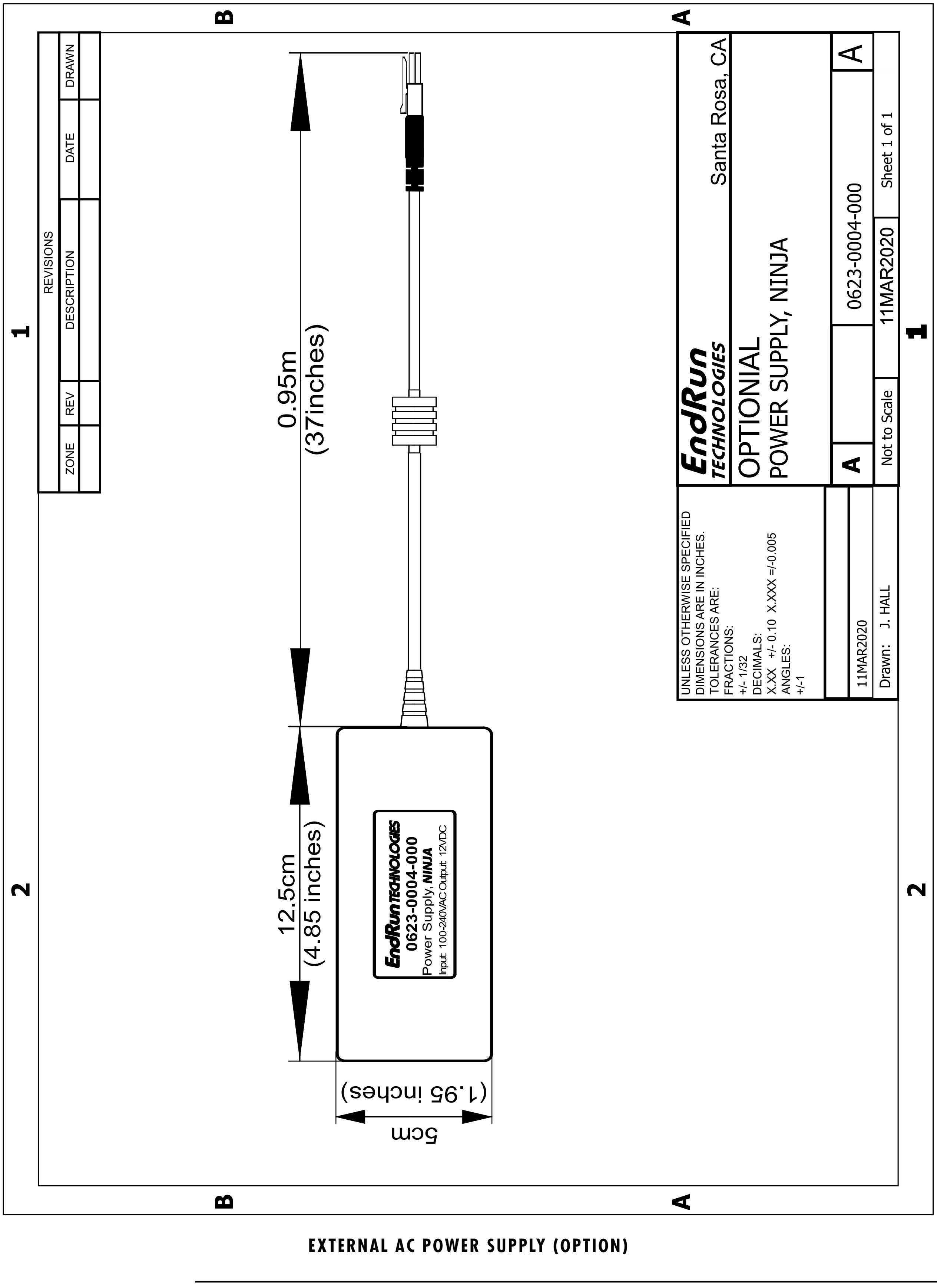
Compliance

CE, FCC, RoHS, WEEE

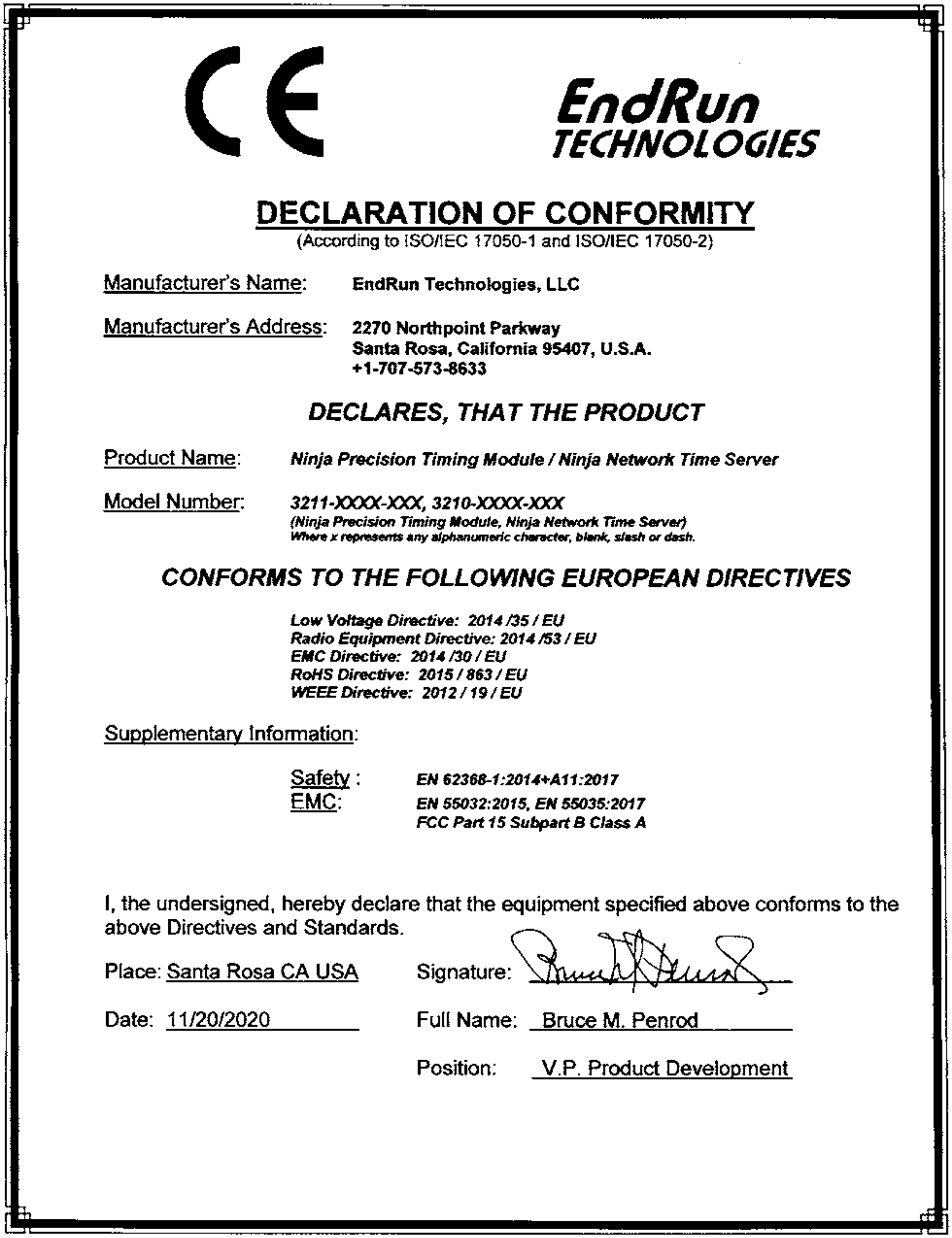
Data subject to change.
EndRun Technologies may make changes to
specifications and product descriptions
at any time, without notice.



NINJA DIMENSIONS SHOWING 9 OPTIONAL OUTPUTS (A - I)



EXTERNAL AC POWER SUPPLY (OPTION)



CE

EndRun TECHNOLOGIES

DECLARATION OF CONFORMITY

(According to ISO/IEC 17050-1 and ISO/IEC 17050-2)

Manufacturer's Name: **EndRun Technologies, LLC**

Manufacturer's Address: **2270 Northpoint Parkway**
Santa Rosa, California 95407, U.S.A.
+1-707-573-8633

DECLARES, THAT THE PRODUCT

Product Name: ***Ninja Precision Timing Module / Ninja Network Time Server***

Model Number: ***3211-XXXX-XXX, 3210-XXXX-XXX***
(Ninja Precision Timing Module, Ninja Network Time Server)
Where x represents any alphanumeric character, blank, slash or dash.

CONFORMS TO THE FOLLOWING EUROPEAN DIRECTIVES

Low Voltage Directive: 2014 /35 / EU
Radio Equipment Directive: 2014 /53 / EU
EMC Directive: 2014 /30 / EU
RoHS Directive: 2015 / 863 / EU
WEEE Directive: 2012 / 19 / EU

Supplementary Information:

Safety : ***EN 62368-1:2014+A11:2017***
EMC: ***EN 55032:2015, EN 55035:2017***
FCC Part 15 Subpart B Class A

I, the undersigned, hereby declare that the equipment specified above conforms to the above Directives and Standards.

Place: Santa Rosa CA USA   Signature:

Date: 11/20/2020   Full Name: Bruce M. Penrod

Position: V.P. Product Development

Special Modifications

Changes for Customer Requirements

From time to time EndRun Technologies will customize the standard Ninja for special customer requirements. If your unit has been modified then this section will describe what those changes are.

This section is blank.

This page intentionally left blank.

EndRun
TECHNOLOGIES
"Smarter Timing Solutions"

2270 Northpoint Parkway
Santa Rosa, CA 95407
TEL 1-877-749-3878
FAX 707-573-8619
endruntechnologies.com