Starnode PC Data Collection Network Operator's Manual

Rev B, February 2002

EM-67043-1B

Starnode PC Data
Collection Network
Operator's Manual
Rev B, February 2002

Factory Authorized Training

SICK Auto Ident, Inc. provides comprehensive product training. Contact the SICK Auto Ident, Inc. Training Coordinator at 1-888-264-4641 for in-house and on-site class schedules and rates.

Getting Assistance

If you have questions or comments, please contact SICK Auto Ident, Inc. at 1-888-264-4641. For additional assistance, contact:

- Order Processing (Option #1)
- Technical Support (Option #2)
- Field Service Contracts (Option #3)
- Customer Service Fax (781) 828-7592

Printed in the United States of America.

SICK Auto Ident, Inc.
5 Shawmut Road
Canton, MA 02021 USA
(781) 302-2500
Fax (781) 828-3150

Contents

PREFACE

Welcome!

Congratulations on your purchase of the Starnode PC Controller. We're glad to have you as a customer and we're sure you'll be pleased with your purchase.

How To Use This Manual

The purpose of the *Starnode PC Data Collection Network Operator's Manual* is to get you up and running with your system quickly and confidently.

This manual presents two major topics:

1. Instructions on how to install your Starnode Controller.
2. How to properly use the Starnode Controller.
3. Instructions on how to develop programs that use the Starnode Controller.

Guide Conventions

The following typographical conventions are used throughout this guide.

- Items emphasizing important information are *italicized* or **bolded**.
- Keyboard entries are indicated as an *italic*.
- Menu selections, menu items and entries in screen images are indicated as: Run (triggered), Modify..., etc.

Note: Provides useful information about the current topic.

Caution: Provides information for the prevention of damage to the software.

WARNING! PROVIDES INFORMATION FOR THE PREVENTION OF PERSONAL INJURY OR DAMAGE TO THE HARDWARE.

Getting Assistance

We hope this manual will be helpful. If you have any questions or comments, please contact SICK Auto Ident, Inc. at 1-888-264-4641. For additional assistance, please refer to the inside front cover.

SICK Auto Ident, Inc. Customer Service Policy

We care about your productivity and will go to great lengths to ensure that you have maximum up-time. Whether you call for a site survey, place an order, or request technical support, you are assured of prompt, courteous, and personalized attention.

Our state-of-the-art accounting and computer management systems permit us to instantly access customer order information. A trained staff member is available to assist you with:

- Order entry assistance
- Product information and application answers
- Product delivery status
- Technical support
- One-on-one problem resolution

Contact your sales representative. Or, to reach SICK Auto Ident, Inc. Customer Service directly, call 1-888-264-4641. The fax number is (781) 828-3150.

Return-to-Factory Instructions

Should your Starnode Controller system fail to operate correctly, verify the following:

- Confirm that it has been properly configured with the proper setup parameters, as ordered.
- Inspect and verify all cable connections.

If a problem persists, contact your sales representative or SICK Auto Ident, Inc. Customer Service by calling the numbers provided in “SICK Auto Ident, Inc. Customer Service Policy”.

Please call SICK Auto Ident, Inc. at 1-888-264-4641 to return a Starnode Controller for repair. Request the Return Authorization (R.A.) Department. Please be prepared to furnish the following information:

- Company name, address, and telephone number
- Contact name
- Return address (if different) and other pertinent shipping information
- Catalog number and serial number
- Description of the problem
- Purchase order and other invoicing information relative to the repair

SICK Auto Ident, Inc. will provide you an R.A. number. Please include this R.A. number on the shipping label and any correspondence concerning the return. Please include several sample barcode labels, a listing of setup parameters and a detailed description of the problem. Repair or upgrade estimates shall be furnished upon request.

Upon receiving a defective product with a valid Return Authorization number, SICK Auto Ident, Inc. will attempt to return the repaired or replacement equipment on a best-effort basis within five working days. You may have a different support plan specifying other terms.

For critical applications, SICK Auto Ident, Inc. recommends you keep a spare controller on hand for immediate replacement. Alternatively, you can select a support plan that specifies a quick response time or a controller swap.

SICK Auto Ident, Inc. shall pay surface transportation charges for the return shipment if the address is within the 48 contiguous states or the District of Columbia. Customers outside this area shall pay shipping costs, customs clearance, and any other related charges.

Your controller will be returned after inspection and repair. However, upon return, the controller may require re-configuration to the setup parameter values you were using.

Product Warranty

SICK Auto Ident, Inc. guarantees that its products are free from defects in materials or workmanship (under proper and normal use and maintenance) in accordance with SICK Auto Ident, Inc.'s operating instructions for a period of one year from the shipping date.

This warranty shall be null and void if equipment is modified, if it is improperly installed or used, if it is damaged by accident or neglect, or if components are improperly installed or replaced by the buyer.

Under no circumstances shall SICK Auto Ident, Inc. be liable to the buyer or any other party for lost profits, diminution of good will, or other special or consequential damages whatsoever.

The warranty appearing here supersedes all other warranties, express or implied, statutory or otherwise, including any implied warranty of merchantability or fitness for a particular purpose.

1

CHAPTER 1 *Introduction*

Introduction

Starnode is a sophisticated barcode data collection network that allows two-way communication with a multi-dropped network of compatible fixed-station laser scanner/decoders or barcode data collection terminals. Employing a flexible tree/bus topology to facilitate network cabling and circuit expansion, Starnode provides a practical alternative to more costly local area networks (LANs) for a wide variety of factory automatic identification applications.

The Starnode PC Data Collection Network Operator's Manual is written for anyone who must install, use, or maintain a Starnode network from a PC equipped with a Starnode PC interface board.

Controllers

The network controller coordinates communication activity on the Starnode network cable. The processor inside the controller polls groups of Starnode devices, gradually narrowing the poll to a specific Starnode device that has a pending message. The controller then passes the message to the host. When the host sends a message to a Starnode device, the controller verifies that the Starnode device addressed did, in fact, receive its message.

Polling is performed independently of the host, so the host is freed for other processing activities.

The controller can also be used to download customized Terminal Application Level Language (TALL) or C programs as Starnode devices are powered up.

SICK Auto Ident, Inc. offers three types of controllers for coordinating communication activity in a Starnode network:

- The PC-resident Starnode interface board
- The Starnode Asynchronous Controller (AC)/ CICALP interface
- The Starnode Asynchronous Controller (AC)/ MUX interface

All three controllers can contain two kinds of memory, Read Only Memory (ROM), which controls the operation of the controller itself, and Random Access Memory (RAM), which contains the user programs to be downloaded to Starnode devices, and buffered messages from Starnode devices.

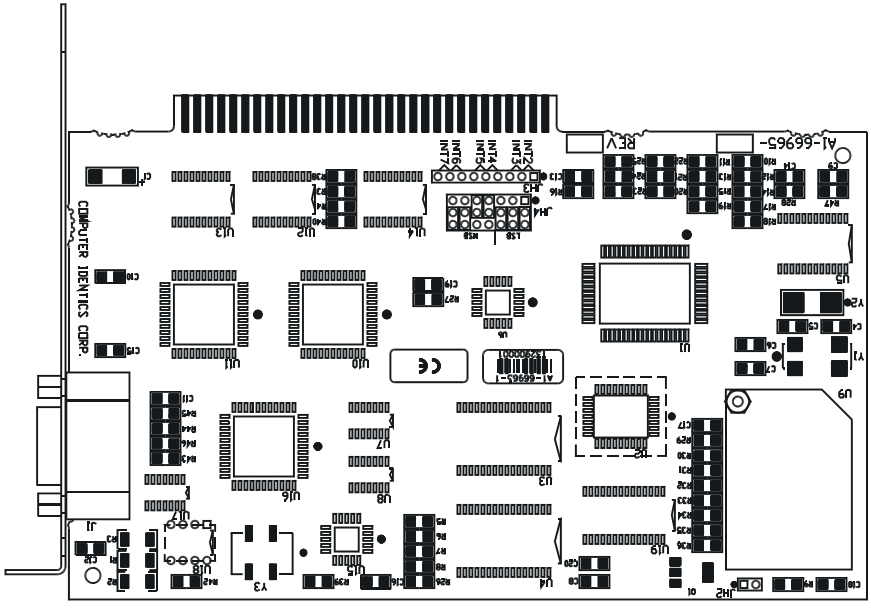
This manual describes the operation of a network via the Starnode PC interface board. While other controllers, repeaters, and accessories are briefly described, please refer to the appropriate operating manuals. Refer to the *Starnode Site Planning and Installation Guide* for information on installing Starnode cabling and hardware.

All three controllers can communicate with other Starnode devices on RS-485 lines at a baud rate of 125K bits per second. The following paragraphs describe each of these controllers.

Starnode PC Interface Board

The Starnode PC Interface Board can be installed in any IBM PC with an ISA slot or compatible PC with an available ISA slot. With the board installed the PC can continue to run resident programs or service other devices without interference from Starnode. The Interface Board with a front view depicting the location and configuration of the external connector is shown in Figure 1–1.

FIGURE 1–1. Starnode PC Interface Board—Front View



The Starnode PC board is available in four versions. All versions support a network of up to 4,095 Starnode addresses. All versions have a 16K character buffer capability to hold Starnode device-initiated messages.

The Starnode PC board may be reconfigured through software to reallocate a portion of the program download memory for use in buffering additional receive messages.

The versions are in the amount of download table space available, shown in Table 1–1.

TABLE 1–1. Downloadable Table Space

Version	Table Space
A1-66965-1	256K
A1-66965-2	512K
A1-66965-3	1024K
A1-66965-4	2048K

Starnode Asynchronous Controller (AC)/CICALP

The Starnode Asynchronous Controller (AC), CICALP interface (A1-66676-1) is a stand-alone unit that coordinates a network of up to 1,023 Starnode addresses. These would consist of Starnode-compatible fixed-station laser scanner/decoders or Starnode bar code data collection terminals. The Starnode AC/CICALP has a 256K-byte file memory.

Software written for the host in the versatile C programming language allows the host and Starnode AC/CICALP to communicate via the SICK Auto Ident, Inc. Asynchronous Link Protocol (CICALP). The Starnode AC/CICALP uses 128K bytes of the file memory for storing a download table; the remaining 128K bytes are used for spooling up to 900 incoming, Starnode device-initiated messages.

The programming and operation of the Starnode AC/CICALP is beyond the scope of this manual. Refer to the *Starnode AC/CICALP Operator's Manual* for more information.

Starnode Asynchronous Controller (AC)/MUX

The Starnode Asynchronous Controller (AC), MUX interface allows up to 1,023 Starnode addresses, consisting of Starnode devices multiplexed on a single host communications link.

The Starnode AC/MUX looks identical to the Starnode AC/CICALP, but is available in two models.

The Starnode AC/MUX (A1-66676-2) has a 256K-byte file memory module that allows the unit to spool messages from Starnode devices. A 128K-byte download table is maintained, while the remaining 128K bytes of file memory are reserved for spooling up to 900 incoming, Starnode device-initiated messages.

The Starnode AC/MUX (A1-66676-3) has no file memory; therefore, it has no download table and it cannot spool Starnode device-initiated messages. However, it is identical in all other respects to the (A1-66676-2).

The programming and operation of the Starnode AC/MUX is beyond the scope of this manual. Refer to the *Starnode AC/MUX Operator's Manual* for more information.

CHAPTER 2

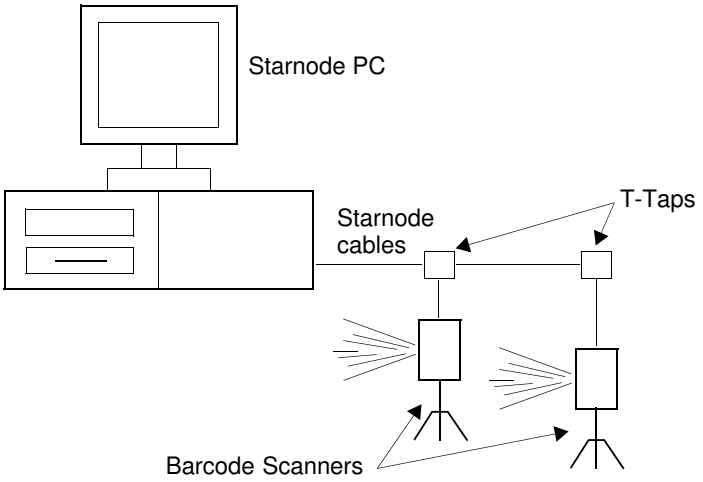
Installation

This chapter provides instructions for installing the Starnode controller, software provided with your controller, network cabling, and T-Taps required to create the Starnode system. Refer to Chapter 3, “Starnode Accessories” for optional software and hardware for use with Starnode.

Installation Overview

Building a Starnode system requires a PC equipped with the Starnode controller, a Starnode software driver and/or application, network cabling, T-Tap boxes for connecting from the Starnode controller to scanning devices, and barcode scanning devices as shown in Figure 2–1.

FIGURE 2–1. Starnode System—Typical Configuration



The software application uses Starnode interface routines within the appropriate driver to initialize the controller to begin polling barcode scanners for data, and to retrieve the buffered barcode data from the controller. The software may also instruct the controller to download programs to the barcode scanners, send messages to specific barcode scanners, and to perform additional functions. Refer to Chapter 6, “Starnode DOS Device Driver” for more information.

Software Installation

The diskette provided with your Starnode controller contains DOS, Windows 3.11, and DOS compatible driver level software, and utilities for DOS and Windows 95, which may be used to control and test the Starnode network.

- For Windows 95 environments, refer to Chapter 5, “Using Starndem” for information on the Starndem utility.
- When using the Starnode *dll* for developing Windows applications, refer to Chapter 7, “Writing an Application Program”.
- For DOS environments using DOS drivers and the Starn utility, refer to Chapter 6, “Starnode DOS Device Driver”.

- For other environments or for an overview of additional software applications for use with Starnode, refer to Chapter 3, “Starnode Accessories”.

To install the software:

1. Insert the Starnode Developers Disk in drive A:.
2. If you are running under Windows, open a DOS session.
3. From the DOS C:> prompt, create a directory on your hard drive to store the software, for example *c:\cisoft\starnpc*. Change to this directory and enter *xcopy a:*.*. /s*. This will re-create the floppy file structure under this sub-directory. Refer to the *readme* file for a full description of the sub-directories and files on the diskette as well as details on any changes to the programs and drivers.

Installing PC-Resident Starnode Board

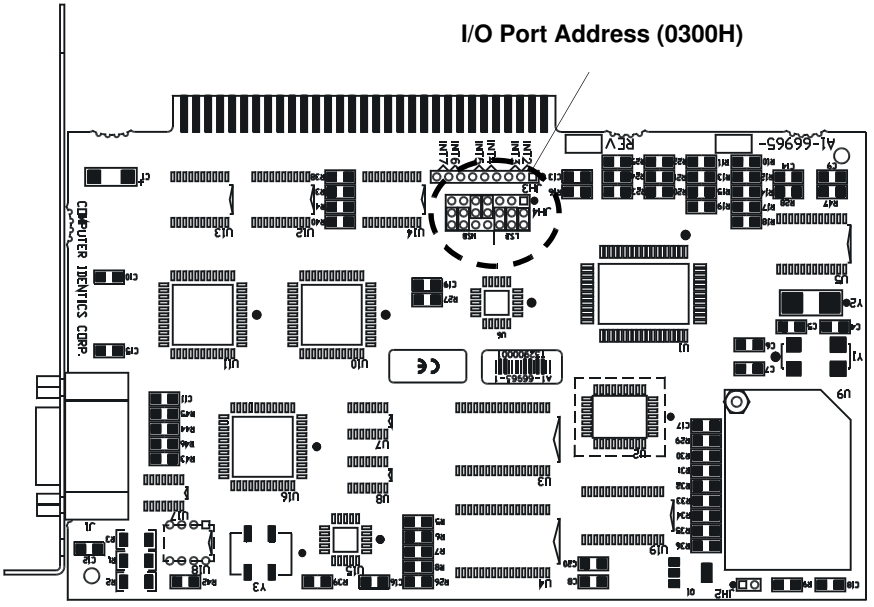
Installing your Starnode interface board may vary according to the model and manufacturer of your PC. Refer to the documentation provided with your computer for board installation.

Install the Starnode interface board as follows.

1. Touch a grounded metal surface (a desk or cabinet, for example). Static electricity may be built up in your body, especially in rooms with thick carpeting. Discharging the electricity into a grounded object avoids the possibility of damage to components on your Starnode Board or inside your PC.
2. Turn the power off to your PC.
3. Loosen all screws that secure the cover of your PC.
4. Remove the cover of your PC.
5. Locate an empty ISA slot in your PC. Most PC's provide ample slots for inserting extra printed circuit boards. Loosen and remove the slot cover for that slot. Retain the slot cover and screws for future use.
6. Being careful to handle it only by the edges, remove the Starnode interface board from its packing material. Do not touch any of the metal contacts on the board.

7. Locate the I/O port address jumpers, labeled JH4 on the board as shown in Figure 2–2.

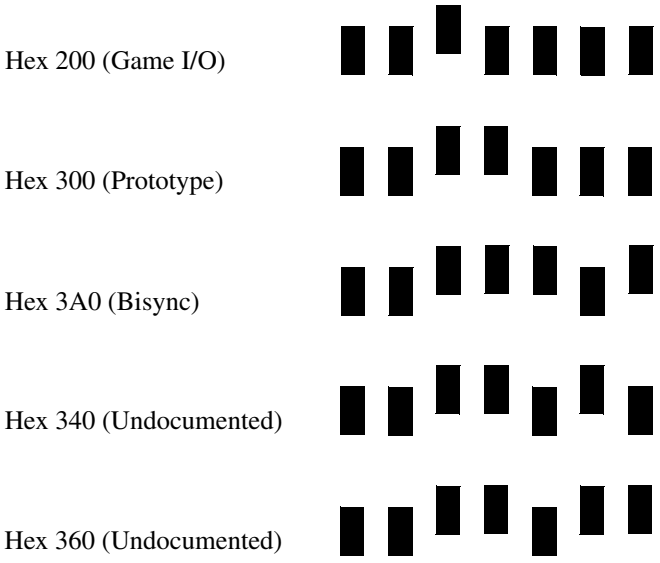
FIGURE 2–2. I/O Port Address Jumpers



The board's address is set as 300 (hex) at the factory. This is the default address that SICK Auto Ident, Inc. software uses. If this is not the desired port address, move the jumpers as necessary. The software must also be configured to use the newly set board address.

8. For the most commonly used board addresses, refer to Figure 2–3.

FIGURE 2–3. Address and Jumper Position Examples



Set the jumpers in the *Up* or *Down* positions to configure the address for your board. In interpreting the jumper settings on JH4, position the interface board with the gold fingers away from you. The LSB should be in the top right of the jumper header.

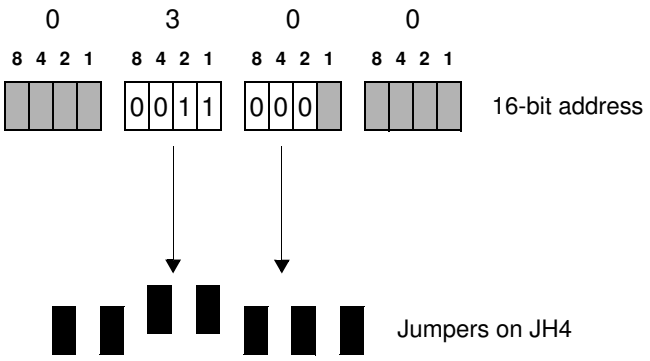
If none of the addresses in Figure 2–3 are available, consult the technical reference manual for your PC to determine which I/O port addresses can be used. To determine the jumper positions for your address, convert your hexadecimal board address into 16 binary digits. For example, if your hexadecimal address is 0300, the 16-bit representation is:

0000 **0011 000**0 0000

└— Only these 7 bits are used

Drop the left most four digits and the right-most five digits as shown in Figure 2–4.

FIGURE 2–4. Converting Hexadecimal Address into Jumper Settings



These bits are shaded and are assumed by software. The remaining seven bits are translated into jumper settings. *Up* for a binary 1, *Down* for a binary 0. Figure 2–4 shows hexadecimal address of 300 (default) translated into jumper settings on JH4.

9. Holding the Starnode interface board by the upper edges or corners, press it firmly down into the slot that you selected. The Starnode PC interface board may not fit perfectly into the PC's slot. Verify that the copper fingers along the bottom edge of the board make good contact with the socket connectors.

10. Secure the screw that holds the connector plate. This will ensure that the interface board does not move.

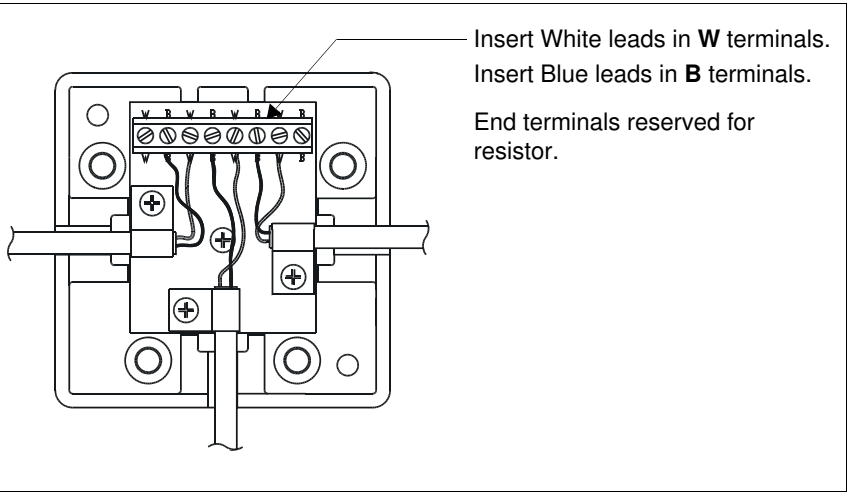
11. Plug the blue Starnode interface cable, 9-pin male connector into the female socket on the Starnode board. Tighten the screws on the connector so that the connector is firmly mated with the Starnode board.

12. Replace the cover on your PC.

T-Tap Boxes

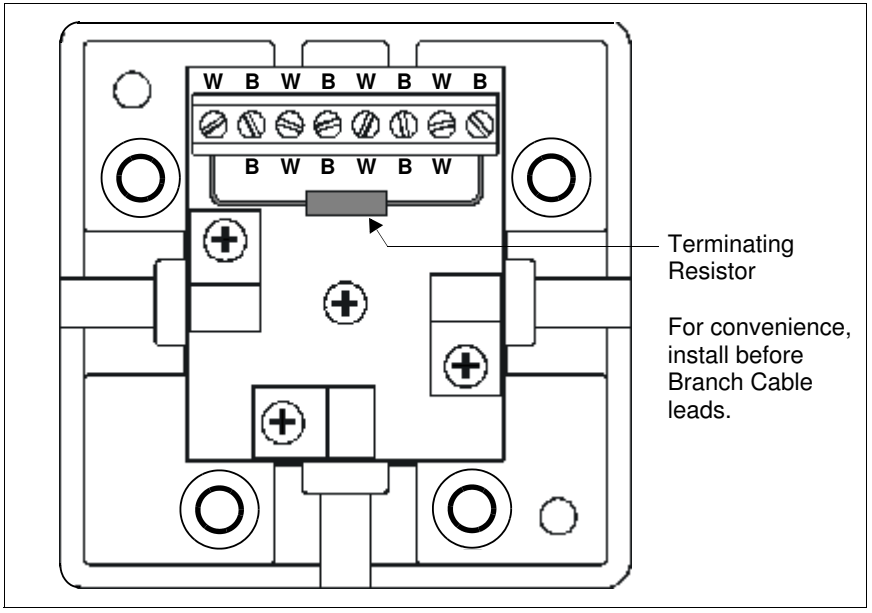
The standard T-Tap connector box is used to connect a compatible network device to a branch circuit as shown in Figure 2–5. It may also be used as a branch circuit cable extender.

FIGURE 2–5. T-Tap Connector



Three connections are made to each T-tap box. However, the T-tap box at the end of each branch circuit has two connections. To properly terminate the branch circuit, it is necessary to install a terminating resistor in the last T-tap box. This factory supplied, 75-ohm, 1/2-Watt resistor is installed across the blue and white bus strips as shown in Figure 2–6.

FIGURE 2–6. T-Tap with Terminating Resistor Installed



To clearly identify T-Tap boxes containing a terminating resistor, a yellow sticker is provided with each T-Tap box. Please remember to apply this sticker only to T-Tap boxes that contain terminating resistors.

Note: If your network is expanded in the future, you may need to remove the terminating resistor from its original T-Tap box and re-install it in another T-Tap box. As your network changes, remember to leave a T label affixed only to those T-Tap boxes that contain a terminating resistor. SICK Auto Ident, Inc. should verify your network after each major change.

Network Cabling

The shielded, twisted pair Starnode 20 AWG cabling meets RS-485 standards, supports high baud rates, and offers immunity to most sources of electrical noise. The shielding also protects the conductors from the effects of electromagnetic interference.

Electrical current originates at the Starnode controller or Repeater, travels down one wire and back on the other. Only one device in the network can be sending data at any instant in time. The Starnode controller uses a polling technique to recognize one device at a time for data transmission to the host.

The signal strength in an RS-485 line is a function of many factors. First, the power of the sending device's transmitter chip can affect signal strength. Also, the number of drops (devices) in the network can affect the circuit: the more drops, the weaker the current on the line. Finally, although all Starnode cabling is 78 ohm (low impedance) cabling, distance can have a major impact on Starnode communications. For example, the greater the cable-distance between the receiver and sender, the weaker the signal.

After the Starnode network has been installed, an ohmmeter should be used to verify that there are no connections in the network with high resistance values. Such points could cause echoes on the line, so the elimination of high-resistance connections is essential.

Signals pass over a Starnode network cable at the speed of light. An *echo* is a signal that bounces back from the end of an interface cable or a poor connection; the time lag between the original signal and the echo causes interference on the network line. The longer the cable involved, the greater the time lags between the signal and echo and the greater the interference.

All Starnode network cabling is standard twin-ax, 78-ohm cable, equivalent to Belden 9463, Alpha 9815, and Manhattan 4227. The dimensions and stiffness of both cable and insulating material are critical to proper electrical connections when using a T-Tap box. Fiber Optic cable can also be ordered if using a Starnode Optic Link.

Cabling Types

Two types of cabling can be used to connect the devices in a Starnode network.

- Standard, 78-Ohm Starnode cabling
- Plenum cable for locations where cabling must be run in the plenum (false ceiling or air chamber). Check the state and local fire codes that apply for your site. The plenum cable offered by SICK Auto Ident, Inc. has a red external shield of Teflon™ or other suitable material. Most state electrical codes require that ordinary cable installed in the plenum be installed in conduit. An SICK Auto Ident, Inc. plenum-rated cable may allow Starnode

network cabling to run through the plenum without the use of costly conduit tubing.

Both cables can be ordered in varying lengths, using SICK Auto Ident, Inc. part numbers as shown in Table 2–1.

TABLE 2–1. Cable Part Numbers

Part Number	Description
A1-70156-1	Standard Starnode cable, 25-foot length, terminated at one end with a ninepin, D-type connector. A 25-foot segment is required for each T-tap installed on the network.
966-0082-1-0100	Standard Starnode cable, 100-foot length (unterminated)
966-0082-1-0500	Standard Starnode cable, 500-foot length (unterminated)
966-0082-1-1000	Standard Starnode cable, 1000-foot length (unterminated)
A1-70181-2	Plenum stub cable, 15-foot length (nine-pin, D-type connector)
966-0103-0500	Plenum cable, 500-foot length (unterminated)
966-0103-1000	Plenum cable, 1000-foot length (unterminated)

Note: Do not substitute other cabling for those specified. On-site fabrication of cables is NOT recommended. Use of unshielded cabling is not supported. Also, using a single cable for multiple purposes other than Starnode is not recommended.

The first branch circuit, connected to the Starnode controller, must always begin with an interface cable and T-tap box, used as a cable splicer. Normally, this T-Tap box will have only two cables attached to it.

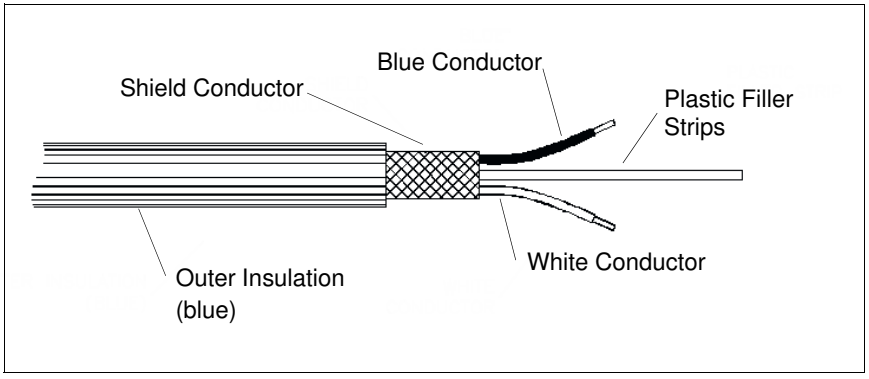
The short (25 or 15-foot) cables with connectors are sometimes referred to as stub cables. These cables are used to connect to a branch cable.

The standard Starnode cable consists of the following parts and is shown in Figure 2–7.

- The blue conductor

- The white conductor
- Two cable strengtheners/fillers
- The shield conductor

FIGURE 2–7. Network Cable Construction



The blue conductor is a wire with blue insulating material. The white conductor is a wire with opaque insulating material. The cable strengtheners/fillers are usually white or blue and of a smaller diameter than the white conductor. The surrounding woven web of silver wire is the shield conductor.

Most plenum-rated cabling sold by SICK Auto Ident, Inc. has a red exterior casing of Teflon™ or other suitable material. The conductors in this cable are black and white instead of blue and white as in the network cabling. For consistency, ensure that you use the same plenum-rated cabling throughout your network. Mixing cabling of different colors may cause confusion.

Barcode Scanners

A number of programmable Starnode terminals can be used in conjunction with the Starnode network. All read barcode data by an attached light pen or laser gun.

Alternatively, fixed-station laser scanner/decoders can be used. These fixed position scanners illuminate barcodes by projecting a laser spot across a moving conveyor carrying barcode-labeled packages or items.

Regardless of which Starnode device is selected, all SICK Auto Ident, Inc. scanners can independently identify and decode a barcode symbology. Thus, only decoded barcode data is sent out to the controller. The scanner or terminal, not the controller, performs the task of decoding.

Refer to the appropriate Starnode device operator's manual for more information.

Testing the Network

The utility Starndem under Windows 95 may be used to test the network as described in Chapter 5, “Using Starndem”. After you have supplied the network I/O base address when prompted, the software will attempt to initialize the controller. The software will begin polling the Starnode controller for new messages and displaying these to the screen. If you are using Starnode data collection terminals, put the terminals in Lan Access Mode. Barcode scanners automatically enter Lan Access mode when they are booted. Scan data from one or more of the barcode devices to insure that the network is functional. If any errors occur during this procedure the software will stop and display the appropriate error message. Should errors occur, refer to Chapter 10, “Maintenance & Troubleshooting”.

If you will be using the controller from a DOS environment, the network may be tested by installing the DOS driver and then using the Starn utility. Refer to “Using STARN Program” on page 6-5. Enter *Starn init* to initialize the network. Scan labels from any barcode scanning device that is currently in Lan Access mode. Run *Starn recv* to retrieve and display the barcode data.

3

CHAPTER 3 *Starnode Accessories*

This chapter describes software and hardware accessories for use with the Starnode Controller.

Software Products

SICK Auto Ident, Inc. offers a number of software products that may be of use when developing Starnode based systems, including drivers for additional host platforms, development tools such as OLE Servers and ActiveX controls, data collection applications, and applications for configuring, monitoring, and setting up your Starnode based barcode scanners.

- Starnode Driver for Windows NT and Windows 2000
- Starnode Driver for SCO UNIX
- Starnode Driver for OS/2

These products provide the basic functionality of the Starnode Developers disk for the corresponding host platform. The Starnode OLE Server for Windows NT requires the Starnode Windows NT driver.

- Starnode OLE Server for 95 and Windows 98
- Starnode OLE Server for Windows NT and Windows 2000

The Starnode OLE Servers contains the utility software Multidem for controlling the Starnode network and collecting data to text files. The utility DbConvert is also provided, for uploading the data to an ODBC compatible database / server.

An OLE Automation server is provided, which simplifies application development. The OLE Server allows one or more applications to share send and receive access to the Starnode controller, for example to run Multidem to log all Starnode data to a text file as an audit trail while a separate custom process control application processes this data. Data sent by the applications to the barcode devices will be buffered in a queue for a configured period of time until the destination device is ready to accept another message. COM Events may be used for notification that new data has been received, or that a queued message has been sent. For environments that do not support OLE Automation servers but do support ActiveX controls, a control has been provided that supports all of the OLE servers' capabilities.

The balance of the software products described support the Starnode OLE Server, as well as the Serial OLE Server and TCP/IP Server. Since the OLE Servers provide shared device access, one or all of these products may be used with the barcode scanning devices at the same time. Any functionality not provided by these applications could be implemented by developing a custom application using the Starnode OLE Server, so that the custom application may also be used at the same time as the other application(s).

- CiMenu32 (requires Starnode OLE Server)

  CiMenu32 is a user-friendly Windows utility for setting up your Starnode compatible barcode devices. It detects which devices are currently connected and displays their current setup for modification. On-screen help for each parameter displays what the parameter is used for and what effect each available setting will have on the device. After modifying the device parameters the new configuration may be downloaded to the device, downloaded to a new device, and saved or restored using a text file.

- DataManager+ 32 (requires Starnode OLE Server)

  DataManager+ 32 is a data collection application, which acts as a server to provide the barcode scanners with access to host database systems / servers. It supports approximately 30 different types of requests that barcode scanners may make including requests to execute SQL statements and procedures, transfer text files, log errors, and send messages to other devices on Starnode, serially, or through TCP/IP.

- CiView (requires Starnode OLE Server)

  CiView may be used to monitor barcode scanning device status and performance over a period of time. A summary display shows which devices

are online or offline, the last label each device has read, and performance statistics for each device such as minimum, maximum, and average read quality.

- OPC Server (requires Starnode OLE Server)

  The OPC Server (OLE for Process Control) can be used by any OPC compliant client software packages to access the barcode scanners. Just a few of the many OPC compliant software packages are CIMPLICITY, RSVIEW32, Fix Dynamics, FactoryLink, and Factory Suite 2000.

Starnode Switch

A Starnode Switch is used when a backup controller is installed in conjunction with the primary controller. This configuration allows the backup to assume the primary controller functions if the primary controller fails. In such a case, the Starnode Switch must be present.

If the Starnode Switch senses that the primary controller is no longer responding, it automatically switches the main branch circuit to the backup controller. The Starnode Switch is fully programmable through TALL.

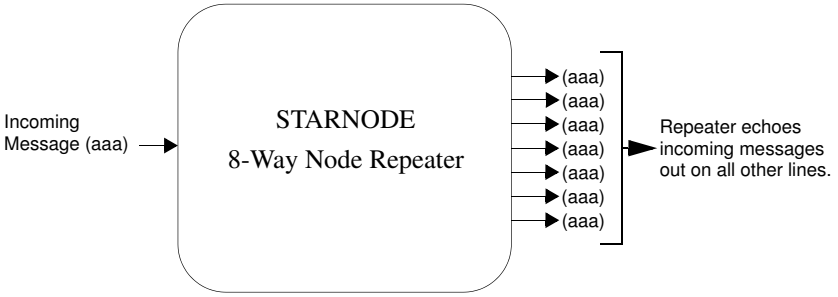
Instead of using two interface cables and a T-tap box to connect each controller to the Starnode Switch, use A1-70184-1. This 25-foot cable has a 9-pin connector installed on each end.

Starnode Repeater

Branch circuits are added to a Starnode network using a Starnode Repeater. A Repeater is connected to a branch circuit via an interface cable. Up to seven additional branches may be connected to a single repeater.

The Repeater works by holding all of its lines in an idle state. Upon receiving a character from a line, it repeats the character to all other lines connected to the Repeater as shown in Figure 3–1.

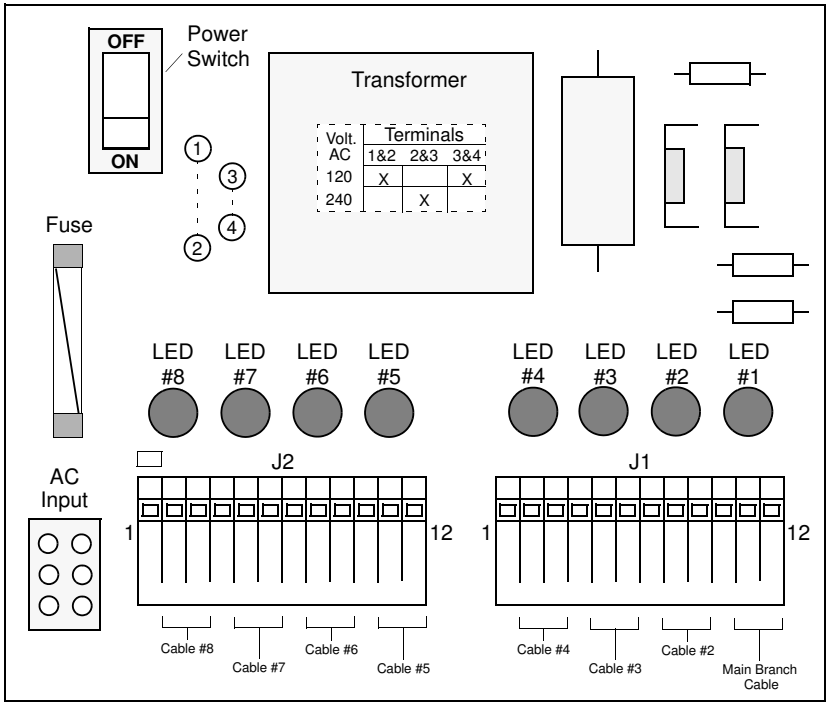
FIGURE 3–1. Data Repetition



Note: The Starnode Repeater is designated as the active device on each branch circuit that originates from it.

The Repeater contains a Power Switch, transformer, two terminal strips, cable clamps, and eight LEDs, as shown in Figure 3–2.

FIGURE 3–2. Starnode Repeater



The Power Switch turns the power to the Repeater box *On* and *Off*. The two terminal strips each have 12 screw-type connectors. The connectors are divided into groups of three. Each group of connectors is used for interfacing one cable.

For example, terminals 11 and 12 on Connector J1 are used to connect the blue and white conductors of a short piece of Starnode 25' cable to a T-Tap box. This T-Tap box may be anywhere on a branch. The repeater conforms to all rules for any terminal.

Inside the Repeater are eight LEDs (light emitting diodes). These are located on the circuit board, near the connectors. There is one LED for each group of three connectors. LED #1 indicates when data is being transmitted over the main branch circuit. LEDs 2 through 8 flash when data is being transmitted on one of the corresponding branch lines. Due to the controller's rapid polling of the network, the flashing of LEDs 2 through 8 appear continuously lit.

4

CHAPTER 4

Software & Programming Overview

This chapter provides an overview of the Starnode polling protocol that allows Starnode devices to communicate with the Starnode PC Interface Board and the other Starnode controllers.

Starnode Communication Protocol

The Starnode network uses a hybrid of probing and CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) polling techniques. The Starnode PC probes large groups of Starnode devices in an effort to determine which single Starnode device is attempting to transmit data.

After probing has narrowed the possibilities to a group of 64 Starnode devices, CSMA is used. Following a synchronization message, each Starnode device is given a specific time interval in which it may start to transmit data. If a Starnode device misses its opportunity to communicate, it must wait until the next polling cycle to try again.

If more than one Starnode device is trying to communicate at once, the other Starnode devices that have messages pending sense the start of the first Starnode device's transmission. These other Starnode devices wait for the next synchronizing message before beginning their own transmissions. Thus, a collision of message data by multiple Starnode devices is avoided.

The Starnode PC Interface Board begins by polling the first 1024 addresses, looking for any device wishing to transmit a message. If there is no response from this group, it continues on through the next 1024, and so forth until it sees a response to the poll. This response tells the board that one or more Starnode devices in the group of 1024 wish to send some data.

The board then polls each group of 256 within that group until another response is seen. Next, polls are performed on groups of 64 Starnode devices within the 256. When the transmission source has been narrowed to 64 terminal addresses, the board uses CSMA/CA. The first device with data sends its message, causing all other devices in the group of 64 to wait for the next synchronizing message. The next synchronizing message is sent to the same group of 64, starting with the Starnode device after the one that sent the last message.

Every Starnode message is sent with a block check character. The receiving device verifies that the block check character is accurate and sends either an acknowledgment (ACK) or indicates that the block check revealed a bad transmission (NAK). If repeated re-transmissions are unsuccessful, the sending device returns an error code to the program that tried to send the message.

Messages sent from Starnode devices to a controller may be prefixed with an <ESC> character. Such messages are automatically placed at the head of Starnode message queue. If the <ESC> buffer (with a capacity of one message) is full, the sending Starnode device re-transmits continuously until buffer space is available for storing the message. When the message is passed to the host, the <ESC> character is retained as the first character of the message.

Message traffic is not limited to controller-terminal exchanges. In a Starnode network, Starnode devices can also communicate with each other. To use this feature, messages must be prefixed with an <ESC> character, followed by the four-digit ID number of the Starnode device to which the message is addressed. The entire message text following the <ESC> character and four-digit terminal ID is transmitted. The first character following the four-digit terminal ID becomes the command byte for the receiving Starnode device.

When terminal-to-terminal messages are sent over the network, the sending Starnode device does receive any kind of acknowledgment from the receiving Starnode device. If it is important that the sending Starnode device be able to verify proper reception by the receiving Starnode device, it is best to configure the host computer to relay the message and verify reception.

Programming Overview

Programming of the Starnode data collection network is accomplished in two steps:

- Programming the Starnode devices to accept collected data
- Programming the Starnode PC to process the data

The PC software required to test and run the network is supplied with your Starnode board. You should first examine the README file for a description of the files and the latest updates.

Starnode Device Programming

SICK Auto Ident, Inc. Starnode devices are programmed using Terminal Application Level Language (TALL), or in C using the *C* Programmers Platform.

TALL compilers may be used on any PC running under DOS or SCO UNIX. The compilers generate output files that can be downloaded to and interpreted by a Starnode device.

TALL is a powerful language, not unlike BASIC, which allows greater programming flexibility. Calculations, sophisticated string manipulation, and Starnode device data storage are all easily controlled using TALL.

The *C* Programmers Platform may be used in the DOS environment to develop applications in ANSI compatible *C* language for any of the C68K based barcode scanners.

The compiled programs can be downloaded, tested, and debugged with the help of the PC-resident STARNDEM program. This program is supplied as a demonstration program on the Starnode Development Disk. It allows you to test or demonstrate most features of the Starnode network.

PC Programming

After a data collection program for the Starnode device has been written and compiled, an application program for the PC may need to be created.

The PC software needs of many users can be conveniently met by a separately available package called DATA MANAGER +32 or the Multidem utility

supplied with OLE Servers. Refer to “Software Products” on page 3-1 for more information.

For users with more complex data collection requirements, application program interface routines for the Starnode PC Interface Board are provided. These routines can be called from applications programs written in any language that supports Microsoft compatible *.obj* modules or DLL’s, including Microsoft Visual C++ 1.52, Microsoft Visual C++ 6.0, and Visual Basic.

5

CHAPTER 5

Using Starndem

Overview

The Starndem Utility is a general purpose, interactive Windows application program that may be used for downloading, sending & receiving messages etc. This program continuously polls the network for incoming data collection transactions. In addition, the utility allows an operator to perform a variety of functions, including the following:

- Transmitting messages or control commands to one or more terminals in the network
- Downloading compiled applications programs (*.*LAN*) or data files (*.*DAT*) to one or more terminals in the network
- Viewing data collection transactions sent by terminals in the network
- Viewing the communication status of a single, specific terminal or the status of all terminals in the network. The Starndem Utility offers the operator a choice of viewing detailed, specific status statistics or a broad, general summary
- Starting and/or stopping all terminals in the network with a single command
- Performing diagnostic tests on all terminals or on specific terminals
- Collecting data and storing it to disk

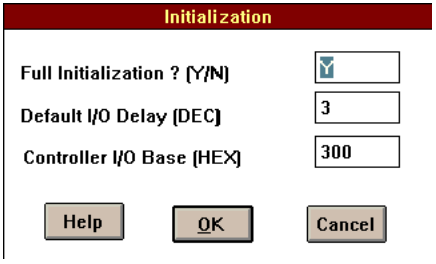
Selection of most of the main menu action bar items causes a dialog box to appear. While one dialog box is still being shown, you may select another main

menu item and cause a second dialog box to appear. Dialog boxes can be overlaid up to 20 deep. If desired, you can click on the dialog box and move it to view the main window or another dialog box.

Starndem uses a configuration file named *starndem.cnf* to determine which applications to download at startup. The format of this file is *type, file.lan*. The default starndem.cnf downloads several demo programs.

When you start the Starndem Utility, the Initialization dialog box is displayed as shown in Figure 5–1.

FIGURE 5–1. Initialization Dialog Box



In most cases, the three default values appearing in the Initialization dialog box will be correct for your Starnode network. You might need to change one, two, or all three values, depending on how your network is set up. For example, if you changed the jumpers on your Starnode PC Interface Board, the Controller I/O Base field's value should be changed.

Controller I/O Base

The Starnode PC Interface Board is configured with the W7 jumpers set for a default board address of hexadecimal 300. The address that you enter in the Controller I/O Base field must correspond to the hex setting of the W7 jumpers on the board, otherwise the Starndem Utility is unable to find the Starnode PC Interface Board.

For I/O ports and addresses commonly encountered on PC's, refer to “Installing PC-Resident Starnode Board” on page 2-3.

To change the value shown, click on the Controller I/O Base field and highlight the present value. Press the *Delete* or *Backspace* key to delete the current value. Then type in a new hex value.

Default I/O Delay

If the Starnode board seems to be generating random errors, try increasing the I/O delay through the Starnode utility.

To change the value, click on the Default I/O Delay field and highlight the present value. Press the *Delete* or *Backspace* key to delete the current value. Then type in a new decimal value.

Full Initialization

The Full Initialization field allows you to specify whether or not the Starndem Utility should download programs to terminals after it is started.

During a full initialization:

- The controller's clock is set
- The Starndem Utility solicits IDs from all terminals in the network
- The Starnode board's download area is cleared
- **.LAN* files are downloaded to terminals by type when *STARNDEM.CNF* is found

Click the Help button if you need to view Help text for this dialog box. Otherwise, after the values in the Initialization dialog box are all satisfactory, click OK.

Starndem.cnf

When you selected Full Initialization in the Initialization dialog box, the Starndem Utility looked for the *STARNDEM.CNF* file. This file is an ASCII text file containing terminal types and the **.LAN* programs that will be downloaded to each terminal.

Example

A typical *STARNDEM.CNF* file may appear as follows:

```
1, E:\cisoft\badge.lan
3, workord.lan
8, E:\cisoft\shortcut.lan
```

These lines specify that all terminals identifying themselves as:

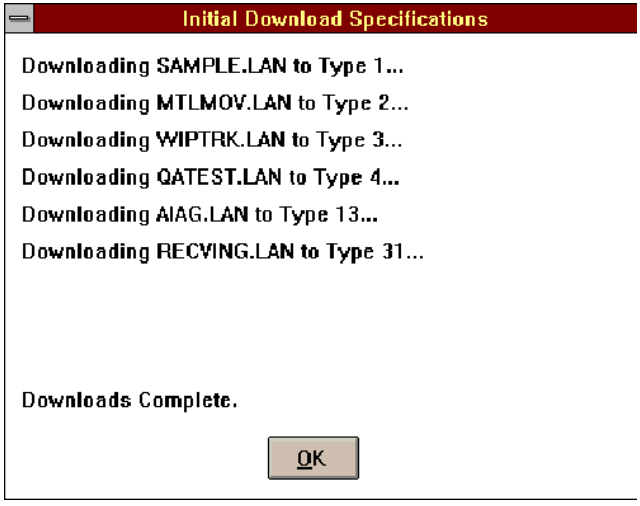
type 1 should receive *BADGE.LAN*;
type 3 terminals should receive *WORKORD.LAN*; and
type 8 terminals should receive *SHORTCUT.LAN*.

The Starndem Utility searches for *STARNDEM.CNF* in the same directory that contains the utility. When *STARNDEM.CNF* is found, the utility uses it to complete the full initialization. In using the contents of the configuration file to determine downloads, the utility looks again for the **.LAN* files in the same directory as the Starndem Utility. If a **.LAN* file is located in a different directory, you must specify the file's full path in the *.CNF* file record.

Example

When the sample configuration file in the previous example was accessed, the Initial Download Specifications dialog box is displayed as shown in Figure 5–2.

FIGURE 5–2. Initial Download Specifications Dialog Box

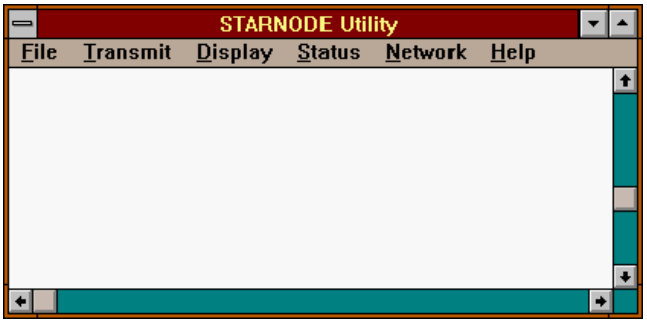


Click OK when you are ready to begin Starndem Utility operation. The Initial Download Specifications dialog box disappears and the Starnode Utility main menu is displayed.

Starnode Utility Menus

The Starnode Utility window contains the following menus, as shown in Figure 5–3.

FIGURE 5–3. Starnode Utility Window

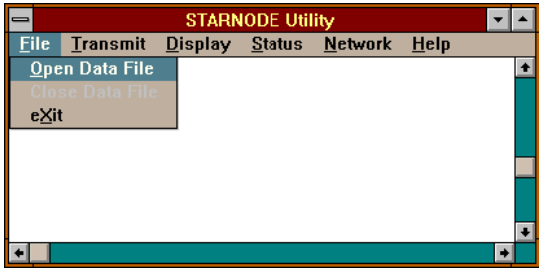


- "Transmit Menu" on page 5-7
- "Display Menu" on page 5-16
- "Status Menu" on page 5-17
- "Network Menu" on page 5-23

File Menu

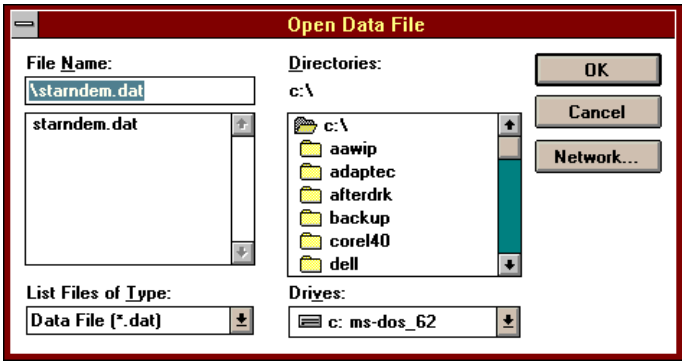
File menu selections are shown in Figure 5–4.

FIGURE 5–4. Starnode Utility Window—File Menu



Open & Close Data File—Opening the File menu and selecting Open Data File displays the Data File dialog box as shown in Figure 5–5. Open Data File causes the Starndem Utility to begin storing data to the indicated data file.

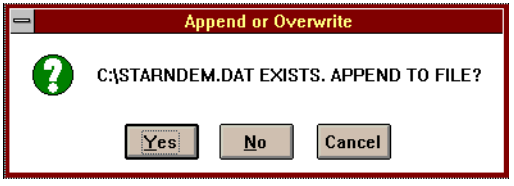
FIGURE 5–5. Open Data File Dialog Box



Click on the data file name or type a complete path and data file name in the File Name field. To change the list of data files, click on a different path in the Files In' list box. After selecting a data file, click OK.

If you specify a non-existent data file, the Starndem Utility creates the new data file and begins storing received transaction data to it. If a pre-existing data file is specified, the Append or Overwrite warning box is displayed as shown in Figure 5–6.

FIGURE 5–6. Append or Overwrite Warning Box

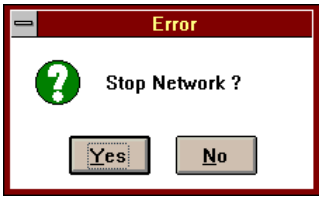


When you click Yes, new data records are appended to the end of the pre-existing data file.

Opening the File menu and selecting Close Data File causes the Starndem Utility to discontinue storing records to the indicated data file.

Exit—Opening the File menu and selecting Exit displays the Error dialog box as shown in Figure 5–7. Exit causes the Starndem Utility to exit.

FIGURE 5–7. Error Dialog Box



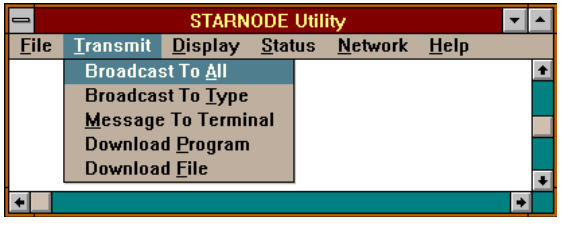
Clicking Yes will immediately stop the network. The next time that you start the Starndem Utility, a *Y* (Yes) appears as the starting Full Initialization field default. Clicking No will cause an *N* (No) to appear as the default at the next Starndem Utility startup.

To change the response, click on the Full Initialization field and highlight the present value. Press the *Delete* or *Backspace* key to delete the current response, then type in either *Y* or *N*.

Transmit Menu

The selections on the Transmit menu are all related to the sending of data to terminals in the Starnode network.

FIGURE 5–8. Starnode Utility Window—Transmit Menu



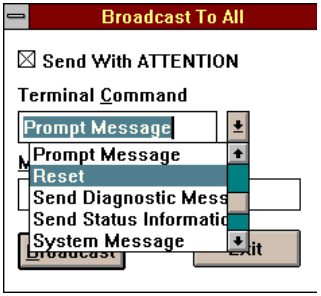
Broadcast to All—Allows you to send a message or command to every terminal in the Starnode network. For example, you might want to send the message *BREAK AT 10 AM* to every terminal.

1. Opening the Transmit menu and selecting Broadcast to All will display the Broadcast To All dialog box as shown in Figure 5–9.

FIGURE 5–9. Broadcast To All Dialog Box

2. Observe the Terminal Command box and determine whether the command shown is the one desired. If you wish to choose a different command, click the Terminal Command button to display the Terminal Command list box as shown in Figure 5–10.

FIGURE 5–10. Broadcast To All Dialog Box—Terminal Command List



Select the desired command type. Click OK when you are satisfied with the command type selected.

Each terminal control command results in the addition of a single character to the massage. This command character tells the Scanstar terminal's operating system what to do with the message.

3. If the command type selected allows you to send message text to the terminals, click on the Message entry field, otherwise, proceed to Step 5.

4. Type a message of up to 126 characters (bytes) in the Message entry field. Messages longer than 126 characters (bytes) will not be transmitted. If you selected a command type of *NONE*, the first character (byte) you enter is used as a command byte by the terminal operating system.

Example

When you select the command type *NONE* with the message *2HI THERE*, the ASCII *2* at the beginning would be interpreted as a command byte. The decimal equivalent of the ASCII *2* is 49, so the *2* would mark the message as a Prompt message.

The Message text can consist of non-printable ASCII characters or special Starnode screen control characters. For example, the prompt message *\007 BREAK AT 10 AM* would not only cause the message *BREAK AT 10 AM* to be displayed, but it would activate the beeper as well.

5. Observe the Send with Attention check box. If the box is checked, an attention character will precede the message that you broadcast. This character causes a receiving terminal to stop whatever it is doing, such as hand-held laser scanning, and process the Starnode message or command immediately. This helps to ensure that the terminals ignore no broadcast messages. If an attention is not necessary, click on the check box. The check disappears and no attention character will be sent.

6. When you are satisfied with the information shown in the Broadcast to All dialog box, click Broadcast. The selected command and message are transmitted to every terminal in the Starnode network.

Broadcast to Type—Allows you to send a message or command to terminals with a common type. For example, if you wanted to broadcast the Prompt message *System Shutting Down* to all type 008 terminals and terminal numbers 0001, 0005, and 0008 are type 008 terminals, only those terminals will receive the broadcast.

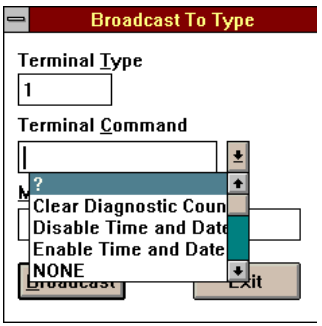
1. Opening the Transmit menu and selecting Broadcast to Type will display the Broadcast to Type dialog box as shown in Figure 5–11.

FIGURE 5–11. Broadcast To Type Dialog Box

2. Determine the terminal type that you want to broadcast to. Click Terminal Type and enter the terminal type.

3. Observe the Terminal Command box and determine whether the command shown is the one desired. If you wish to choose a different command, click the Terminal Command button to display the Terminal Command list box as shown in Figure 5–12.

FIGURE 5–12. Broadcast To Type Dialog Box—Terminal Command List



Select the desired command type. Click OK when you are satisfied with the command type selected.

Each terminal control command results in the addition of a single character command identifier at the beginning of the message. This character tells the Scanstar terminal's operating system what to do with the message.

4. If the command type selected allows you to send message text to the terminals, click on the Message entry field, otherwise, proceed Step 6.

5. Type a message of up to 126 characters (bytes) in the Message entry field. Messages longer than 126 characters (bytes) will not be transmitted. Remember that if you selected a command type of NONE, the first character that you enter is used as a command byte.

   Example

   If you selected command type NONE with the message *2HI THERE*, the ASCII *2* at the beginning would be interpreted as a command byte. The decimal equivalent of the ASCII *2* is 49, so the *2* would mark the message as a prompt.

   The Message text can consist of non-printable ASCII characters or special Starnode screen control characters. For example, the Prompt message *\021 TIME TO QUIT* would not only cause the message *TIME TO QUIT* to be shown, but it would cause the terminal's beeper to emit 3 short tones.

6. Observe the Send with Attention box. When the box is enabled (checked), the message that you broadcast will be preceded by an attention character. This character causes a receiving terminal to stop whatever it is doing, such as hand-held laser scanning, and process the Starnode message or command immediately. This helps to ensure that terminals ignore no messages. If an attention is not necessary, disable (uncheck) Send with Attention box. The check disappears and no attention character will be sent.

7. When you are satisfied with the information shown in the Broadcast to Type dialog box, click Broadcast. The selected command type message is transmitted to every terminal of the specified type in the Starnode network.

Message to Terminal—Allows you to send a message or command to a specific terminal in the Starnode network. For example, you may wish to send the prompt message *YOU'RE LATE* to terminal number 9.

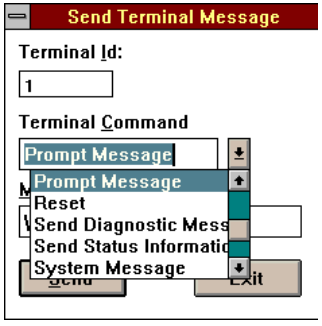
1. Opening the Transmit menu and selecting Message to Terminal will display the Send Terminal Message dialog box as shown in Figure 5–13.

FIGURE 5–13. Send Terminal Message Dialog Box

2. Determine the terminal number to which you want to transmit. Depending upon the type of Starnode controller, terminal numbers can be any positive integer between 0001 and 4095. Click on the Terminal Number entry field, then enter the terminal ID.

3. Observe the Terminal Command box and determine whether the command shown is the one desired. If you want to choose a different command, click the Terminal Command button to display the Terminal Command list box as shown in Figure 5–14.

FIGURE 5–14. Send Terminal Message—Terminal Command List



Select the desired command. Click OK when you are satisfied with the command selected.

Each terminal control command results in the addition of a single character to the beginning of the message. This character tells the Scanstar terminal's operating system what to do with the message.

4. If the command type selected allows you to send message text to the target Starnode terminal, click on the Message field. Remember that if you selected a command type of None, the first character (byte) that you enter is used as a command byte.

5. Type a message of up to 125 characters (bytes) in the Message entry field. Remember if you selected a command type of None, the first character you enter is used as a command byte.

Example

When you selected command type NONE with the message *2HI THERE*, the ASCII *2* at the beginning would be interpreted as a command byte. The decimal equivalent of the ASCII *2* is 49, so the *2* would mark the message as a prompt message.

The message text can consist of non-printable ASCII characters or special Starnode screen control characters. For example, the prompt message *\022 YOURE LATE* would not only cause the terminal's display to show the message, but it would cause the beeper to emit three short tones.

6. Observe the Send with Attention box. If the box is enabled (checked), the message will be preceded by an attention character. This character causes the terminal to stop whatever it is doing and process the message or command immediately. This helps to ensure that no messages are lost. If an attention is not necessary, disable (uncheck) the Send with Attention box. The check disappears and an attention character will be sent.

7. When you are satisfied with the information shown in the Message to Terminal dialog box, click Send button to transmit to the target terminal.

More than one Send Terminal Message dialog box can be displayed at the same time enabling you to see more than one message at a time.

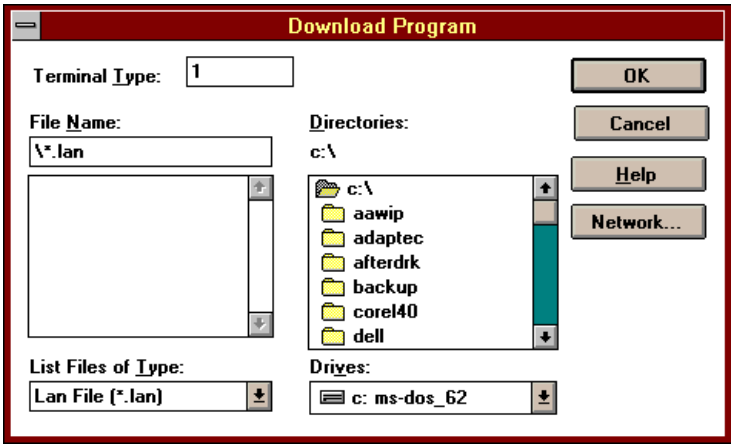
Download Program—Allows you to transmit a compiled **.LAN* program to all terminals in the network of a specified type. Such downloads might be used for testing or demonstration of the network's capabilities.

Note: If you specified Y (Yes) to the Full Initialization prompt and STARNDEM.CNF file was found, programs have already been downloaded.

---

1. Opening the Transmit menu and selecting Download Program displays the Download Program dialog box as shown in Figure 5–15.

FIGURE 5–15. Download Program Dialog Box



2. Determine the terminal type that you want to download to. Click on Terminal Type field and enter the terminal type. Valid types are 001 to 128.

3. Specify the **.LAN* file to be downloaded by typing a complete path and program name in the File Name field or by clicking on the program name. To change the list of **.LAN* files shown, select on a different path from the File Name list box.

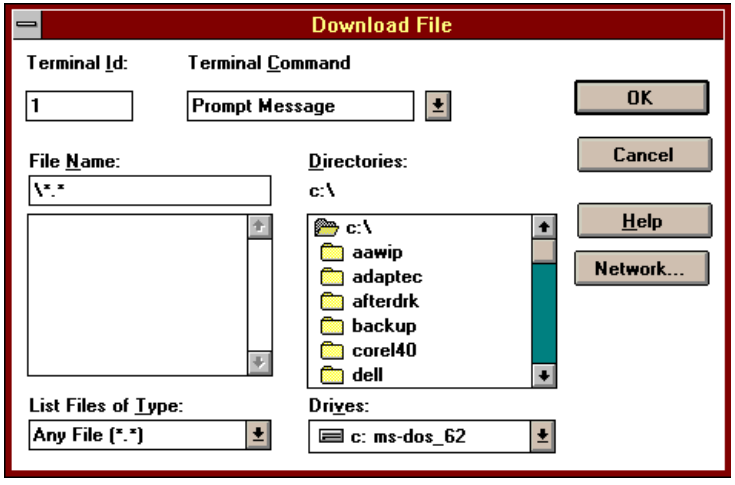
If a different program was downloaded to the target terminals during a previous session, the later download takes the place of the previous one.

After selecting a **.LAN* file, click OK. The hourglass is displayed while the download is in progress, then disappears when the download is complete. You may then select a new file name and/or terminal type and download to the next group of terminals.

Download File—Allows you to transmit a text file (*.DAT) to a single, specific terminal in the network. For example, this file may be used to store data, perform lookups, or may contain data to be sent to a printer. It may also be a compiled TALL program for downloading to a single terminal. Download File is most commonly used to send a file to a non-Starnode device. The serial port of a Starnode device serves as a gateway to the Starnode network. For example, you might want to download a text file called *TEST.DAT* down to terminal number 8.

1. Opening the Transmit menu and selecting Download File will display the Download File dialog box as shown in Figure 5–16.

FIGURE 5–16. Download File Dialog Box



2. Determine the terminal number that you want to transmit to. Depending upon the type of Starnode controller, terminal numbers can be any positive integer between 0001 and 4095. Click on the Terminal Id field and enter the terminal ID.

3. Observe the Terminal Command list. If any terminal command except None is selected, each record in the text file will be automatically be downloaded with the selected value as the first byte (command byte) in the record.

Example

The value entered in the Terminal Command entry field represents the command byte ASCII character, in decimal form. For example, decimal 49 represents an ASCII *2* character indicating to the Starnode network that this record is a prompt message.

If you specify None for the Terminal Command, you must check that each record in the downloaded data file starts with the desired command byte.

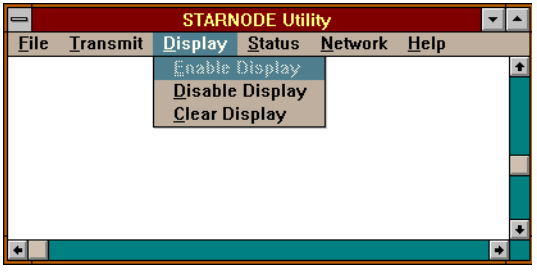
If you are trying to download to a Scanstar terminal's file memory, you must write and download a program to the terminal before starting the download. During the downloading of the second program, each line in the file must have an ASCII command byte of decimal 65 or greater.

4. Specify the **.DAT* file to be downloaded by typing a complete path and file name in the File Name entry field or by clicking on the file name in the Files In list box. To change the list of **.DAT* files shown, click on a different path from the Files In list box. After selecting a **.DAT* file, click OK. The selected file is downloaded to the desired terminal.

Display Menu

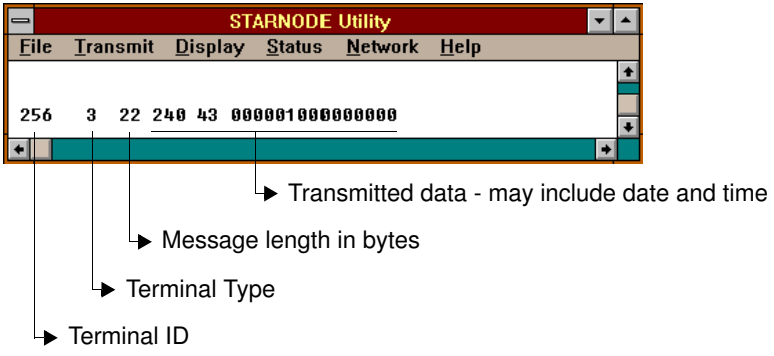
The Display menu controls the viewing of Starnode data collection as shown in Figure 5–17.

FIGURE 5–17. Starnode Utility Window—Display Menu



Enable Display—When selected, all data collection transactions performed by the Starnode network are displayed as shown in Figure 5–18.

FIGURE 5–18. Starnode Utility Window



The originating terminal's ID and type are shown, as well as the message length and the transaction data. If your program was written to include them, the transaction's system time and date can be included in the record.

Transaction data is shown first at the bottom of the screen. New transactions appear at the bottom of your PC's screen and older transactions are scrolled upward until they are no longer visible.

To view previously collected transaction data, use the scroll bar to scroll back to the desired data. Similarly, if the collected transaction data is too long to be viewed, use the scroll bar to view the extreme right-most edge of transaction data records.

In applications where data is being collected rapidly, a scanner, for example, the data display should be disabled completely or the Starndem Utility window should be re-sized so that only one or two lines of transaction data are visible. Either of these alternatives will allow messages to be collected faster. In addition, any multi-tasked programs running on the PC should run noticeably faster if the screen display is disabled.

Disable Display—Any data collection transactions performed by the Starnode network are not shown on your PC's screen. If any transactions are performed on the network while the Display is disabled, none of the data is shown, even if the Display is subsequently re-enabled.

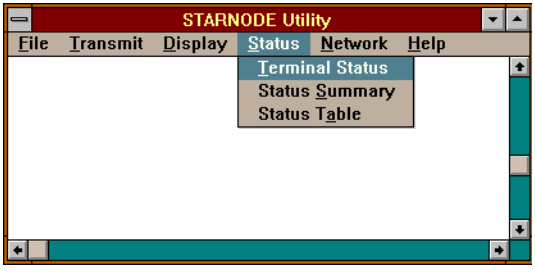
As described previously, in applications where data is being collected rapidly, the data display should be disabled.

Clear Display—Clears the Starndem Utility's main display of all data shown.

Status Menu

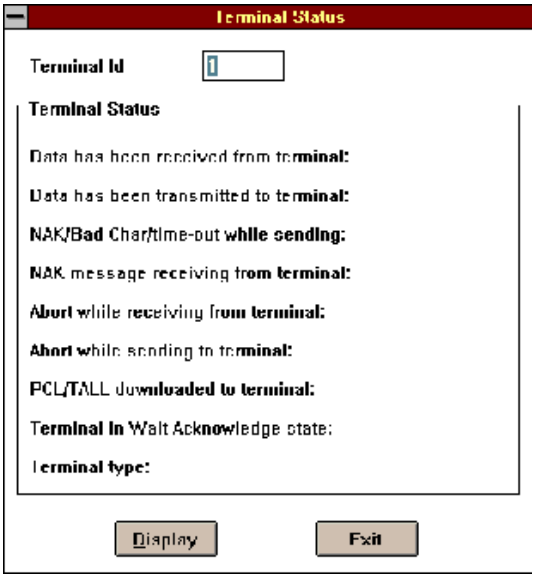
The Status menu controls the viewing of various aspects of the Starnode data collection system's status on your PC's screen as shown in Figure 5–19.

FIGURE 5–19. Starnode Utility Window—Status Menu



Terminal Status—Allows you to view a summary of the status of specific terminals in the network. Since multiple dialog boxes can be displayed, the status of more than one terminal can be viewed simultaneously. Open the Status menu and select Terminal Status to display the Terminal Status dialog box as shown in Figure 5–20.

FIGURE 5–20. Terminal Status Dialog Box



Click the Terminal ID field. Type the number of the terminal whose status you wish to view. Depending on the type of Starnode controller, terminal numbers can be any positive integer between 0001 and 4095.

Click Display to view the status. Each time you click Display, that window's status information is updated. When you are finished viewing the status, click Exit.

For the terminal selected, the following information is displayed:

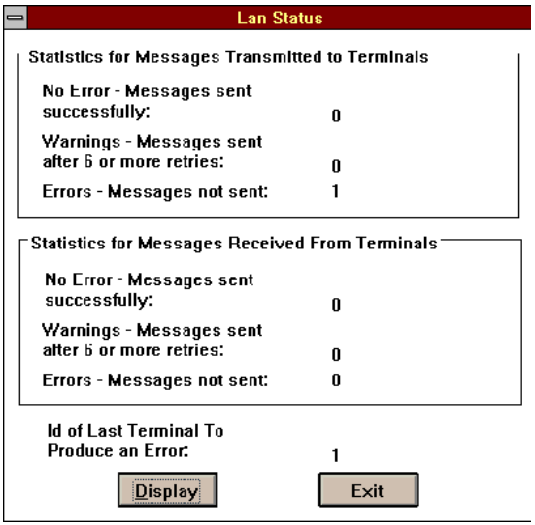
- If Starnode successfully received any data from the terminal since the last network initialization
- If Starnode successfully transmitted any data to the terminal since the last network initialization

- If Starnode received any NAKs or bad characters, or if any communication time-outs occurred while Starnode was sending data to the terminal
- If any messages were NAK'd while Starnode received data from the terminal
- If while receiving data from the terminal, the Starnode controller exceeded the maximum number of NAKs it is supposed to send and aborted out of the communication attempt
- If while transmitting data from the terminal, the Starnode controller exceeded the maximum number of NAKs it is supposed to send and aborted out of the communication attempt
- If a compiled program was successfully downloaded to the terminal since the last network initialization
- If the terminal is in a Wait Acknowledge (WACK) state. This state indicates the terminal is still processing the last message. The terminal will leave the WACK state when it is ready to process another message.

When you have finished viewing statuses, click Exit to display the Starndem Utility main menu.

Status Summary—Allows you to track various communication statistics. This information reflects the number of transmission attempts between terminals and the Starnode network, as well as the number of failures generated as a result of those transmission attempts. Open the Status menu and select Status Summary to display the Lan Status dialog box as shown in Figure 5–21.

FIGURE 5–21. Lan Status Dialog Box



Every message transmitted over the Starnode network in some way changes the statistics in the LAN Status dialog box: either a message is sent successfully or it could not be sent. If a terminal is busy and fails to acknowledge a message, the Starnode controller re-transmits it up to 10 times. If all 10 attempts are unsuccessful, the controller gives up and reports an error. The statistics shown in the Lan Status dialog box are really a subset of a more complete status table. The comprehensive set of statistics can be viewed through the Status Table selection.

The LAN Status dialog box reflects the following statistics:

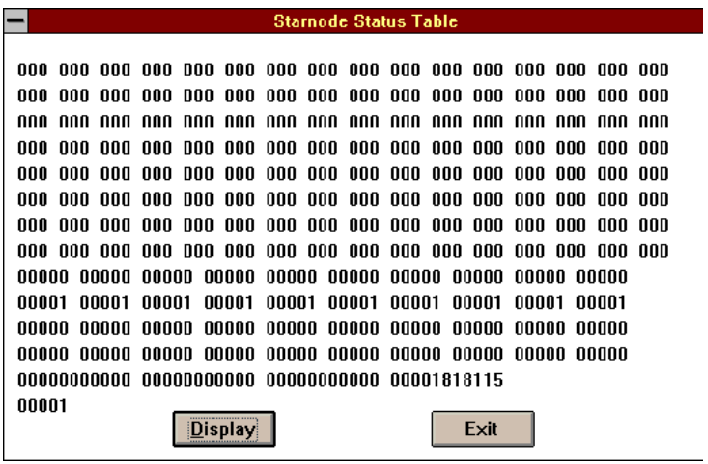
- The number of messages successfully sent to terminals since the last network initialization
- The number of messages successfully sent to terminals that required six or more retries before communication was successful
- The number of messages to terminals that could not be transmitted after 10 attempts
- The number of messages successfully received from terminals since the last network initialization

- The number of messages received from terminals that required six or more retries before communication was successful
- The number of messages from terminals that could not be transmitted after 10 attempts
- The ID number of the last terminal to generate an error status

Each time you click the Display button, the LAN Status information is updated. When you are finished viewing the status, click Exit.

Status Table—Provides access to detailed information about Starnode communications. Status Table is primarily used by SICK Auto Ident, Inc. during network troubleshooting. Open the Status menu and select Status Table to display the Starnode Status Table dialog box as shown in Figure 5–22.

FIGURE 5–22. Starnode Status Table Dialog Box



Lines 1 through 4—These lines represent groups of 64 Starnode addresses. The digits shown, which should always be zeros, indicate the number of times that the Starnode controller received a positive response to a probe poll, but never received a message from a terminal in that group.

A Starnode controller polls groups of 1024 Starnode-compatible terminals. If any terminal in the group of 1024 has a message, it responds to the probe poll. The controller then re-polls in groups of 256, then 64, until it isolates the responding terminal.

The controller sends a synchronizing message, and the terminal has a pre-determined amount of time in which to send its data. If this response window is missed, the terminal must wait until the next synchronizing message.

The first (left-most) group of zeros in Line 1 indicates status for Starnode addresses 1 through 64, the second for 65 through 128, etc.

Lines 5 through 8—The second group of four lines indicates how may times a group of 64 Starnode addresses saw a positive response to a probe poll since the last network initialization. As with Lines 1 through 4, the first (left-most) group of zeroes in Line 1 indicates status for Starnode addresses 1 through 64, the second for 65 through 128, etc.

Lines 9 through 12—When a message is sent to a Starnode terminal, there can only be one of five possible responses: ACK, NAK, WACK, no answer at all, or unintelligible data. ACK acknowledges successful reception of a message.

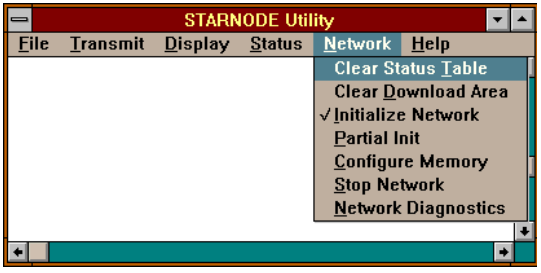
- *NAK*—Line 9 indicates how many NAKs (unsuccessful message attempts) were received by Starnode. The first number shows how many NAKs were received after one transmission attempt. The second number shows how many NAKs were received after two transmission attempts, etc. Normally, only the first two to four numbers are non-zero. If values are seen beyond the sixth position, check the Starnode cabling for layout and integrity.
- *WACK*—Line 10 indicates how many times no response was received at all. The first number shows how many non-responses occurred after one transmission attempt. The second number shows how many after two transmission attempts, etc. Again, non-zero values should not appear beyond the fourth position.
- *Unintelligible Response*—Line 11 shows how many times unintelligible responses were seen after one attempt, two attempts, etc.
- *Receive Errors*—Line 12 shows the number of receive errors accumulated since the Starndem Utility was last started.
- *Line 13*—The first number in Line 13 shows the number of messages received by the Starnode controller. The second number in Line 13 reflects the number of messages sent by the controller. The third number is reserved for future use. The fourth number shows the total number of probe polls sent out.
- *Line 14*—Shows the terminal ID of the last terminal to report an error.

Each time you click Display, the Status Table information is updated. When you are finished viewing the network's status, click Exit.

Network Menu

The Network menu controls various aspects of system operation, such as starting or stopping network polling, resetting the communication statistics to zero, and more as shown in Figure 5–23.

FIGURE 5–23. Starnode Utility Window—Network Menu



Clear Status Table—Allows you to initialize the network status table described in the section above on the Status Table. You generally should initialize the network status table just before running the network diagnostics. After you select Clear Status Table from the action bar, the statuses are all reset to zero and begin accumulating again.

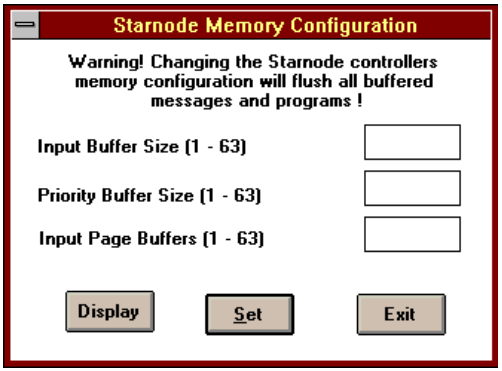
Clear Download Area—Allows you to delete all compiled **.LAN* files from the controller's download area.

Initialize Network—Allows you to begin polling the network and download programs automatically to all Starnode terminals. Unless the network has been stopped, the network should require initialization only once (usually at startup).

Partial Init—Initializes the Starnode PC Interface Board without erasing program, error and type tables or downloading to the terminals. The board is made operational, and starts polling the network for messages.

Configure Memory—Displays the Starnode Memory Configuration dialog box, which enables you to configure memory on the Starnode controller board, as shown in Figure 5–24. The warning against losing programs and messages stored the memory buffers.

FIGURE 5–24. Starnode Memory Configuration Dialog Box



Input message buffer and Priority message buffer share a pool of 64 buffers. (Each buffer is 512 bytes.) You must set a minimum of 1 buffer for normal input messages (default = 63) and 1 buffer for priority messages (default = 1).

Input Page Buffers (16K each) configures the amount of memory to take from the user program download area and use for input messages (default = 0). Refer to Table 5–1 for the maximum you can take for each system option.

Maximum settings for parameter 3, Input Page Buffers are shown in Table 5–1.

TABLE 5–1. Input Page Buffer Settings

System with	Maximum Input
2 MB option	64
1 MB option	32
512K option	16

If you input invalid values for any of the three settings, you will cause the memory to reset to its defaults.

Note: Starnode ROM 9.2 or higher is required. Use the "About" selection in the Help menu to see your ROM version. Call SICK Auto Ident, Inc. technical support if you need to upgrade your ROM.

Clicking on the Set button provides the following actions:

- Flushes all downloaded programs

- Flushes all buffered messages
- Stops network polling
- Requires 2 minutes to complete settings in memory

Set the Starnode Memory Configuration only once.

Wait 2 minutes before clicking Display to show your settings. If you attempt to display before 2 minutes are up, the Error dialog box will be displayed as shown in Figure 5–25.

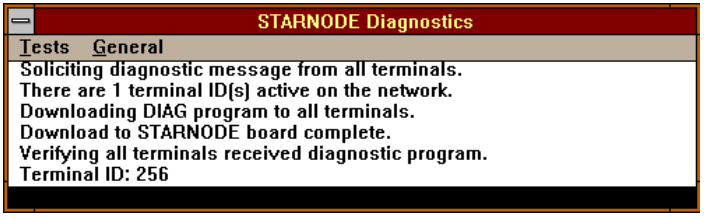
FIGURE 5–25. Error Dialog Box

Stop Network—Allows you to stop the controller from polling terminals in the network. After selecting Stop Network, the network may not appear to stop immediately. This is because the controller may have already buffered up to 25 messages in memory before Stop Network was selected.

A way to utilize Stop Network is to stop the network, close any open data files, then restart the Starndem Utility without full initialization. Data files could then be opened and the network initialized. This prevents any broadcast messages from being missed.

Network Diagnostics—Open the Network menu and select Network Diagnostics to display the Starnode Diagnostics dialog box as shown in Figure 5–26.

FIGURE 5–26. Starnode Diagnostics Window



Network Diagnostics

Network Diagnostics is intended to test network communications by sending messages to and receiving messages from every terminal in the network. This communication can help you to determine if messages are being transmitted and received successfully and if there are any terminals in the network with identical terminal ID numbers.

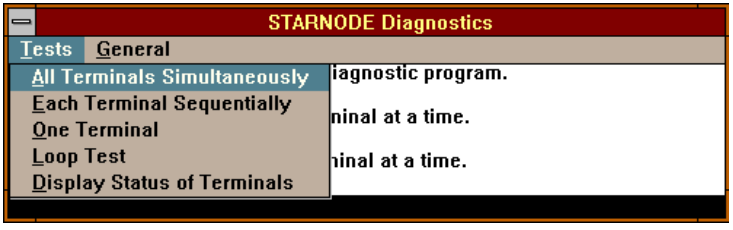
Note: The Windows Installation program should have stored a program named DIAG.LAN *in the same directory as the Starndem Utility. Upon receiving* DIAG.LAN*, the Scanstar terminal should display:*

DIAGNOSTICS IN PROGRESS...DO NOT POWER OFF.

Starndem Utility solicits a Diagnostic String message from each terminal in the network. The utility recognizes each active terminal and then downloads *DIAG.LAN* to each.

The Starnode Diagnostics window offers the two menus, Tests and General as shown in Figure 5–27.

FIGURE 5–27. Starnode Diagnostics Menus



When you select Tests, the various diagnostics tests are listed. Select a test to begin diagnostics.

Diagnostic tests are supplied primarily for use by SICK Auto Ident, Inc. service personnel or when you are directed to perform diagnostics by a SICK Auto Ident, Inc. representative.

Setting Loop Count

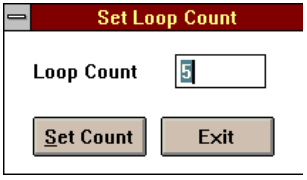
Many of the tests allow you to set a loop count. The default loop count is 4, however, this value can be overridden. The following tests use a loop count:

- All terminals simultaneously test
- Each terminal sequentially test
- One terminal test
- Loop test

Each of these tests causes the Starndem Utility to send a message to the target terminals the number of times defined by the loop count. Before selecting a test, check the loop count and determine if it is set as desired.

To set the loop count in preparation for a test, open the General menu and select Set Loop Count. This will display the Set Loop Count dialog box as shown in Figure 5–28.

FIGURE 5–28. Set Loop Count Dialog Box



To set the loop count to a value other than the one shown, select the Loop Count field. Press *Delete* or *Backspace* to erase the current value. Type the new value. Click Set Count to apply the new value. You can move the dialog box and begin diagnostics, or you can select Exit to close the dialog box.

Note: Ensure that you select Set Count *before selecting* Exit*, otherwise no change in the loop count will be implemented.*

With the loop count set, you are ready to begin one of the tests.

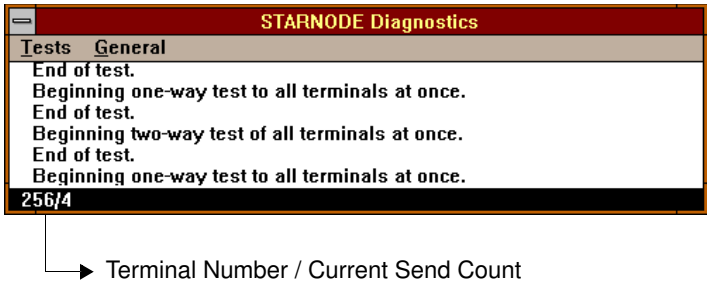
All Terminals Simultaneously—Open the Tests menu and select All Terminals Simultaneously to start the test. This test consists of two sub-tests. First, a series of messages is sent to every terminal in the network. The number of messages sent depends upon the loop count.

Each message instructs the terminal to increment a counter. When all messages have been sent, a final message is broadcast to all terminals, instructing them to

return their counter values. Any terminal that fails to return the proper counter value is identified on your monitor.

The terminal currently being sent to, the current loop send count and the status of the tests are shown in a status line at the bottom of the Starnode Diagnostics window as shown in Figure 5–29.

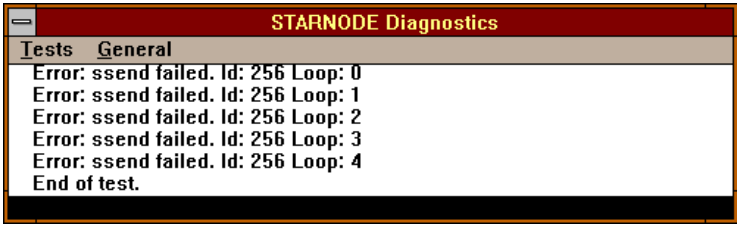
FIGURE 5–29. Starnode Diagnostics Window—Terminal No. & Send Count



During the one-way test, data flows in only one direction: either to or away from the Starnode terminals. During the two-way test, data is sent in both directions over the network.

Instead of each terminal being instructed to increment a counter, the terminal is instructed to reply immediately. When the transmission loop count has been satisfied, the Starndem Utility verifies that each terminal responded the proper number of times. Any unexpected communication events are displayed as shown in Figure 5–30.

FIGURE 5–30. Starnode Diagnostics Window — Unexpected Communications



If messages like this appear, examine the terminal that could not be contacted. Determine whether the terminal is still receiving the time from the Starnode controller.

Note: If the terminal's display has a period (.) to the right of the time and date, the terminal's own processor is supplying the time and date and may indicate a Starnode communication problem. Determine whether other terminals on that particular branch are communicating or not.

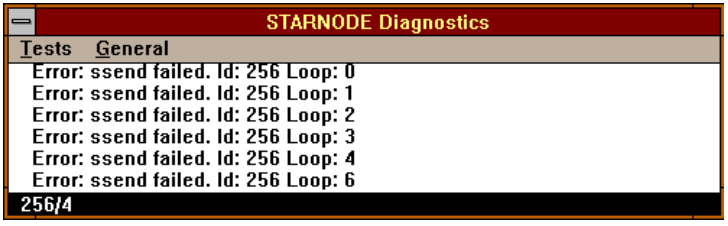
Each Terminal Sequentially—Open the Tests menu and select Each Terminals Simultaneously to start the test. This test consists of two sub-tests and is similar to the All Terminals Simultaneously, except that it tests each terminal, one at a time, sequentially, by terminal ID.

First, the utility sends a series of one-way messages to every terminal on the network. These messages are sent sequentially, by terminal ID. The response of each terminal is then checked. Once all terminals have been tested for one-way communication, two-way communication is tested.

The number of messages sent during each test depends on the loop count. The terminal currently being sent to, the current loop send count and the status of the tests are shown in a status line at the bottom of the Starnode Diagnostics window.

When messages similar to the one shown in Figure 5–31 are sent, examine the terminal that could not be contacted. Determine whether the terminal is still receiving the time from the Starnode controller.

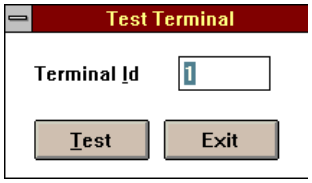
FIGURE 5–31. Starnode Diagnostics



Note: If the terminal's display has a period (.) to the right of the time and date, the terminal's own processor is supplying the time and date and may indicate a Starnode communication problem. Determine whether other terminals on that particular branch are communicating or not.

One Terminal—Open the Tests menu and select One Terminal to start the test and display the Test Terminal dialog box as shown in Figure 5–32.

FIGURE 5–32. Test Terminal Dialog Box



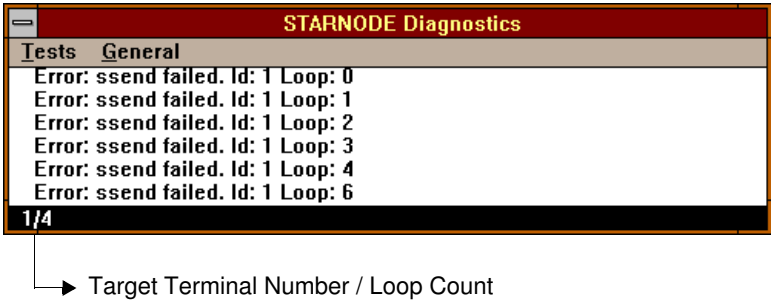
This test consists of two sub-tests. First, the utility sends a series of one-way messages to the target terminal. The response is checked. Once it has passed the one-way test, two-way communication is tested.

The number of messages sent during each test depends on the loop count. The target terminal, the loop count, and the status of the tests are shown in a status line at the bottom of the Starnode Diagnostics window.

This diagnostic tests one specified terminal. To specify the desired terminal, click on the Terminal ID field. Press *Delete* or *Backspace* to erase the current value. Type the new value. Click Test to start the test. You can move the dialog box and begin diagnostics, or you can select Exit to close the dialog box

When the transmission loop count has been satisfied for each terminal, the Starndem Utility verifies that each terminal responded the proper number of times. Any unexpected communication events are reported as shown in Figure 5–33.

FIGURE 5–33. Starnode Diagnostics—Target Terminal No. & Loop Count



When messages similar to the one shown in Figure 5–33 are sent, examine the terminal that could not be contacted or view the terminal status through the Terminal Status selection. Determine whether the terminal is still receiving the time from the Starnode controller. If the terminal's display has a period (.) to the

right of the time and date, the terminal's own processor is supplying the time and date and may indicate a Starnode communication problem. Determine whether other terminals on that particular branch are communicating or not.

Loop Test—Causes the Starndem Utility to thoroughly test communications in the network. After Loop Test has been run, you may select Status Table or Status Summary.

Display Status of Terminals—Causes the Starndem Utility to individually poll every Starnode ID address. When selected, the message *Retrieving Terminal Status Information* is displayed. All possible Starnode addresses are polled for status information and all non-zero statuses are shown.

Ending the Current Test—Causes the diagnostics test currently running to be terminated.

6

CHAPTER 6 *Starnode DOS Device Driver*

This chapter describes the Starnode Device Driver and its functions and provides detailed information on the installation and use of the driver is presented.

Overview

SICK Auto Ident, Inc. Starnode DOS Device Driver is a complete software package designed to allow the user full control of the Starnode PC Data Collection Network while operating in a DOS environment.

The Starnode DOS Device Driver is installed on any PC compatible computer that utilizes DOS Version 3.0 or higher. With the device driver installed, Read/Write operations and DOS input/output function calls (IOCTL) may be used to send and receive messages to specific Starnode devices on the Starnode Network. This allows database software like Lotus 1-2-3 to be used to read data directly from the Starnode interface board.

To successfully boot and run the Starnode DOS Device Driver, you will require the following minimum configuration:

- PC compatible computer with ISA slot
- Display and display adapter
- 1.44M floppy disk drive and a hard disk
- MS-DOS version 3.0 or higher
- EDIT.COM (supplied with DOS), or another ASCII text-editing program.

Driver Installation

Floppy drive *A:* will be the source drive and the hard disk drive *C:* will be the target drive. Install the Starnode DOS Device Driver as follows:

1. Place the Starnode DOS Device Driver floppy disk into drive A.
2. Copy DOS Driver into the target drive's root directory using the command.

   COPY A:\DOSDRV\PRIMARY\STARNDRV.SYS C:

3. Edit the file *C:\Config.sys* in the root directory to contain the line.

   DEVICE=STARNDRV.SYS.

Note: Refer to "Modifying CONFIG.SYS File" on page 6-2 for information on modifying CONFIG.SYS.

---

4. Reboot your PC. When the system boots, the following message should be displayed:

   Starnode Device Driver installed...

5. Read through this chapter, which provides the instructions necessary to run the Starnode Dos Device Driver
6. If you plan on performing Read/Write operations to the STARNBRD device, instead of making input/output control calls, modify your AUTOEXEC.BAT file as described in "Using STARN Program" on page 6-5.
7. The installation is now complete.

Modifying CONFIG.SYS File

When your PC begins loading the DOS operating system, it searches for a special file called *CONFIG.SYS*. This file is used to create a non-standard system configuration. In order to use the Starnode DOS Device Driver on your PC system, you must modify the CONFIG.SYS file. Using EDLIN/EDIT (supplied with DOS) or another ASCII text editor, add the following line to the CONFIG.SYS file:

DEVICE=STARNDRV.SYS PPPP ER EW EF "NNNNNNNN"

where:

DEVICE=STARNDRV.SYS

Loads the Starnode DOS Device Driver with the default values:

I/O port address	0300(Hex)
End-of-Read	to be indicated by 1A(Hex) (<CTRL>Z - DOS End -of-File Marker)
End-of-Write	to be indicated by 04(hex) (EOT)
End-of-File	to be indicated by a 1A(Hex) (<CTRL>Z)
Alternate Device Name	None

All parameters after *STARNDRV.SYS* are optional, however, if a parameter value is specified, all preceding parameters must also be specified as shown in Table 6–1.

TABLE 6–1. STARNDRV.SYS Parameters

Parameter	Description	Value
PPPP	I/O Port Address.	4 Hex characters.
ER	End-of-Read Char.	2 Hex characters.
EW	End-of-Write Char.	2 Hex characters.
EF	End-of-File Char.	2 Hex characters.
NNNNNNNN	Alternate Device Name	8 characters in quotes

Note: All of the above parameters must be exactly the length specified. You may elect to supply no parameters, only the first parameter, only the first two parameters, etc. All remaining fields after the last one filled will use the default values listed above.

The Alternate Device Name, if present, must be exactly eight characters long. If you wish to use a name shorter than eight characters, fill in the remaining places with space characters.

The *CONFIG.ADD* file contains the instruction line

DEVICE=STARNDRV.SYS

You may add this line directly to your *CONFIG.SYS* file in order to load the Starnode DOS Device Driver with the defaults listed above, or you may modify

the file to include parameters, which you specify. Several examples of modified load instructions are presented below.

Example 1

DEVICE=STARNDRV.SYS 03A0 0D 3B 03 "STARNEW "

This instruction loads the device driver with I/O Port Address 03A0H, End-of-Read indicated by a 0DH (carriage return), End-of-Write indicated by 3BH (;), End-of-File indicated by 03H (ETX character), and the Alternate Device Name set to *STARNEW.*

Note: STARNBRD is always a valid device name to use when referring to the Starnode board, however using an Alternate Device Name (STARNEW in this case) allows you to customize the device name to match existing software packages or programs. It is important also to note that a space has been included following the STARNEW name, thus making it exactly 8 characters long.

Example 2

DEVICE=STARNDRV.SYS 03A0 0D 3B 03

This instruction loads the device driver with port address 03A0H, with End-of-Read indicated by a ODH (carriage return), End-of-Write indicated by 3BH (;), and End-of-File indicated by a 03H (ETX character). There is no Alternate Device Name.

Example 3

DEVICE=STARNDRV.SYS 03A0 0D 3B

This instruction loads the device driver with port address 03A0H, with End-of-Read indicated by a 0DH (carriage return), and End-of-Write indicated by a 3BH (;). Default parameters set End-of-File to be indicated by 1AH (<CTRL>Z - DOS End-of-File) with no Alternate Device Name.

Example 4

DEVICE=STARNDRV.SYS 03A0 0D

This instruction loads the device driver with port address 03A0H and End-of-Read indicated by a 0DH (carriage return). Default parameters will set End-of-

Write to be indicated by a 04H (EOT), End-of-File to be indicated by a 1AH (<CTRL>Z DOS End-of-File), with no Alternate Device Name.

Example 5

DEVICE=STARNDRV.SYS 03A0

This instruction loads the device driver with port address 03A0H. Default parameters will set End-of-Read to be indicated by a 0DH (carriage return), End-of-Write to be indicated by a 04H (EOT), End-of-File to be indicated by a 1AH (<CTRL>Z DOS End-of-File), and no Alternate Device Name.

Using STARN Program

This section describes the application of the STARN program to control the Starnode board.

Provided on your Starnode Development Disk is the Starnode Board Control Program (STARN). If you are not using input/output control (IOCTL) calls to control the Starnode board, you will need a way to initialize the board, download programs, broadcast messages, etc. The STARN program supports all of these Starnode board capabilities. STARN is to be run from your AUTOEXEC.BAT file or the DOS command line. The program takes all of its information from the command line, executes one operation, and then returns to DOS.

Modifying AUTOEXEC.BAT File

AUTOEXEC.BAT is a special file that DOS searches for every time you boot your PC system. After DOS has completed all the configuration tasks specified in the *CONFIG.SYS* file, it will execute the commands listed in the batch file *AUTOEXEC.BAT*. Before beginning to Read or Write messages over the Starnode network the typical user will need to:

1. Initialize the Starnode board.
2. Synchronize the PC date and time with that on the Starnode Board.
3. Download any user programs that will be used.
4. Perform other startup tasks; such as broadcast a greeting message to all Starnode devices.

Using EDIT (supplied with DOS) or another ASCII text-editor, add all necessary STARN command lines to the end of your *AUTOEXEC.BAT* file. The following sample command lines use STARN to accomplish the typical tasks a user would need completed prior to performing Read/Write operations.

Initialize the Starnode board using I/O Port Address 300 (hex)
STARN init 300

Synchronize Starnode board real-time clock with DOS date & time
STARN time

Download *wip.lan* to Starnode board for all Starnode devices type 1.
STARN downt 1 wip.lan

Download *quality.lan* to Starnode board for all Starnode devices type 3.
STARN downt 3 quality.lan

Download *inv.lan* directly to term. # 100
STARN downl 100 inv.lan

Broadcast message to all Starnode devices announcing network is up.
STARN brdc 48 "\005\022Starnode Network Active"

STARN Routine

This section provides a listing of all the routines STARN uses to control the Starnode interface board. The routine's name and parameters are listed in bold type. The parameters are listed within parenthesis.

Note: When including parameters in your command lines, do not use parentheses as delimiters, but instead use single spaces between parameters.

STARN delay (delay value)

This routine establishes the time that the PC should wait after each communication with the Starnode board. This allows that higher values may be used on PC's, which have faster clock speeds in the ISA bus than the industry standard of 8Mhz.

STARN init (port address (hex))

This routine initializes the Starnode board. The default I/O port address (0300) hex will be used if no parameter is supplied. When the routine is called, all

download and error tables on the Starnode board are cleared, but the received message queue is not affected.

STARN iop (port address (hex))

Use this routine in order change the I/O port address without re-initializing the Starnode board. If no port address is provided, a default address of 300 (hex) will be used. This routine is useful if you are operating with more than one Starnode board in a single PC, each with a different I/O port address.

STARN time

This routine retrieves the current date and time from DOS, and instructs the Starnode board to synchronize its real-time clock with the values from DOS.

STARN tblc

This routine clears all user program download tables from the Starnode board.

STARN downt (term type) (file name)

This routine downloads a user program to the Starnode board for all Starnode devices with the specified type. All parameters are required. The file name may include a drive letter and directory path. Do not enclose the file name in quotes. Once the routine has returned to DOS, the Starnode board will automatically download the program to any Starnode device on the network that has a matching type.

STARN down1 (term id) (file name)

This routine downloads a user program to a single terminal ID. This action bypasses the Starnode board's download table, and sends the program directly to the Starnode device. All parameters are required. The file name parameter may include a drive letter and directory path. Do not enclose the file name in quotes. When this routine has completed, the Starnode board is unaware that the specified Starnode device has had a program downloaded to it.

STARN errcl

This routine clears the error log on the Starnode board.

STARN errgt

This routine retrieves the error log from the Starnode board and displays it on the PC screen. Knowledge of the Starnode network protocol and operating characteristics is required to correctly interpret this table. You should consult

with the SICK Auto Ident, Inc. application support group if you need additional information.

STARN stat (term id)

This routine retrieves the status of the specified Starnode device from the Starnode board's status table, and displays the value on the PC screen in hex. The terminal ID parameter is required.

STARN send (term id) (command) ("message")

This routine sends a message to the specified Starnode device on the Starnode Network. The command value tells the Starnode device what to do with the message (display it, hold it for the program running in the Starnode device, etc.). Consult the Starnode device manual for a list of valid command values. All parameters are required.

STARN attn

This routine sends an attention signal over the Starnode Network, causing all Starnode devices to suspend operations (such as laser gun scanning) until the next message is sent over the network. This routine is useful when broadcasting important messages that you do not want Starnode devices to ignore. A broadcast message, sent without a prior attn signal, may be ignored if the Starnode devices are busy.

STARN brdc (command) ("message")

This routine broadcasts a message to all Starnode devices on the network. The command value tells the Starnode device what to do with the message (display it, hold it for the program running in the Starnode device, etc.). Consult the Starnode device manual for a list of valid command values. All parameters are required.

STARN brdt (term type) (command) ("message")

This routine broadcasts a message to all Starnode devices on the network having the specified terminal type. The command value tells the Starnode device what to do with the message (display it, hold it for the program running in the Starnode device, etc.). Consult the Starnode device manual for a list of valid command values. All parameters are required.

STARN recv

This routine retrieves one message from the Starnode board (if any are waiting) and prints it on the screen. The ID number and type of the Starnode device, which sent the message, will also be displayed.

STARN stop

This routine stops all Starnode network activity. When this routine has completed, the Starnode board stops polling the network and all time transmissions on the network are stopped, and all network polling is halted. Bar code terminals may be programmed to detect network stoppage, and operate independently while the network is not available.

STARN pinit (port address (hex))

This routine initializes the Starnode board in a manner similar to *STARN init* except the *STARN pinit* does not erase program error tables. This routine may be used if you do not desire to re-download a user program to the Starnode board after initializing the Starnode network.

STARN ver

This routine returns two bytes of information. The low order byte contains the Starnode board firmware version in hex format, while the high order byte contains a number representing the size of the download table as shown in Table 6–2.

TABLE 6–2. Download Table Size

1	256K bytes download table size
2	512K bytes download table size
3	1M bytes download table size
4	2M bytes download table size

STARN reset

This routine completely resets the Starnode board setup parameters to the factory defaults settings, and runs full hardware diagnostics. Diagnostics may require up to 2 minutes to complete. Access to the Starnode board is not possible during this interval.

STARN setmem (input size) (priority size) (download size)

Additional buffer space may be made available to buffer either or both of received messages and priority messages, by reducing the number of download pages. Use STARN *getmem* to retrieve the current configuration. Run STARN *setmem* specifying a larger input size and a correspondingly smaller download pages size. Allow up to 2 minutes for the Starnode Controller to complete re-configuration before attempting to use the board. Once the controller has been re configured it will retain this configuration.

STARN getmem

This routine returns the status of prior STARN SETMEM attempts and the number of message buffers or priority buffers. This routine also retrieves and displays the current memory configuration on the Starnode controller. Example results may be as shown below:

- (63) 512-byte buffer(s) used for receive messages
- (1) 512 byte buffers used for priority messages
- (0) 16K buffer(s) of the download area are used for receive messages specifying:
  - 512 byes are reserved for priority messages (messages beginning with an escape character)
  - 32K bytes are used to store received messages
  - All of the download area is used for storing download programs

Using Read/Write Interface

This section details how to perform Reads and Writes directly from the DOS prompt line.

The Starnode DOS Device Driver can be used in two ways:

1. Using Reads and Writes directly from the DOS prompt to the file called *STARNBRD*.
2. Using DOS input/output control function calls.

The Read/Write capability allows standard business application programs such LOTUS1-2-3 or DBASE III to read data directly from the Starnode board by importing the contents of a file called *STARNBRD*.

Write Interface

A message may be sent to a specific Starnode device on the Starnode Network by writing to the character device (file) called *STARNBRD*. The data written to *STARNBRD* must use the following format:

Terminal, Command, Message EOW

where:

Terminal—Is the terminal number (in decimal) that is to receive the message. Leading zeros or spaces are ignored, and a comma must follow this field.

Command—Is the command code (in decimal) that tells the receiving Starnode device what to do with the message (display it, hold it until picked up the program running in the Starnode device, etc.). Leading zeros or spaces are ignored, and a comma must follow this field.

Message—Is a string of at most 126 characters that is to be sent to the Starnode device. All data that follows the comma used to delimit the command field is considered part of the message. Leading zeros and blanks are not ignored.

EOW—Is the End-of-Write character specified in the *CONFIG.SYS* file. The default value is 04 (hex), the ASCII EOT character. All characters in the message preceding the EOW will be sent to the Starnode device; therefore, you should not have a comma between the message and the EOW character unless you wish the comma to be part of the message. The EOW character is not sent to the Starnode device.

Examples

You may send a message to a Starnode device directly from the DOS prompt line by typing the following:

```
COPY CON: STARNBRD
63,49,Hello, this is a test2<CTRL>D<CTRL>Z
```

Note: <CTRL>D and <CTRL>Z are Control/D and Control/Z).

The Control/D informs the Starnode DOS Device Driver where the message ends. The Control/Z is a DOS delimiter that signifies the end of the DOS copy command. The above message causes terminal number 63 to receive and display the message *Hello, this is a test*.

```
COPY CON: STARNBRD
63,49,test1<CTRL>D100, 49,test2<CTRL>D<CTRL>Z
```

This command sends two messages: *test1* to terminal 63, and *test2* to terminal 100. The command value of 49 instructs both Starnode devices to display the received messages.

```
COPY CON: STARNBRD
63,49,test1<CTRL>D
100,49,test2<CTRL>D<CTRL>Z
```

This command does the same thing as the previous command, illustrating how control characters or spaces that precede the terminal number are ignored. The first line has a carriage return/line feed at its end, that was inserted when *Enter* key was pressed.

Refer to “Terminal Control Commands” on page 9-1 for a complete list of the two-digit decimal commands that may be used to control what a Starnode device does with a message.

Read Interface

A received message may be read from the Starnode board by reading from the character device *STARNBRD*. The data read from the Starnode board will have the following format:

```
Terminal, Type, Message EOR
```

where:

Terminal—Is the terminal number (in decimal) of the Starnode device that sent the message. This field will always be four characters long, with leading zeros if the number is less than 1000. A comma will always follow the terminal number.

Type—Is the type (in decimal) of the Starnode device that sent the message. This field will always be 3 characters long, with leading zeros if the number is less than 100. A comma will always follow the terminal type.

Message—Is the data sent by the Starnode device. All data that follows the comma after the terminal type is message data. The message will always begin in the 10th character position of the data returned by performing the READ.

EOR—Is the End-of-Read character that the Starnode DOS Driver was instructed to use by the *CONFIG.SYS* file. If the default value <CTRL>Z (End-of-File) is used, then no character will be seen following the message, and an End-of-File condition will exist instead. When a <CTRL>Z is used, one message at a time will be retrieved from the Starnode board (if there is a message waiting).

If the EOR is <CTRL>M (carriage return), messages will be retrieved from the Starnode board until there are no more messages waiting. A <CTRL>M (carriage return), causing many software packages to advance to the next line or the next record will follow each message.

Example

TYPE STARNBRD

If EOR has been set to <CTRL>Z (the default value) one message will be displayed on the monitor, if there is one waiting on the Starnode board.

If EOR has been set to <CTRL>M (carriage return), messages will continuously be displayed on the PC screen until there are no more messages waiting on the Starnode board. Each message will be followed by a carriage return (the end-of-read character), causing the display of the next message to overwrite the display of the previous message.

When there are no more messages on the Starnode board, the Starnode Dos Driver will return the End-of-File character. If the End-of-File character is set to <CTRL>Z, then DOS will terminate the TYPE command and return to the DOS prompt. If EOF has been set to something other than <CTRL>Z, DOS will continually attempt to read from the Starnode board and a <CTRL>Z will have to be typed in order to abort the TYPE command.

EOR, EOW, & EOF Characters

EOR, EOW, and EOF characters are only relevant when you are using the READ and WRITE options, and have no effect upon input/output (IOCTL) calls to the Starnode DOS Device Driver. The Starnode DOS Device Driver collects data from the Starnode board and passes it to DOS. DOS then passes the data on to whatever program is requesting it, but examines every character before passing it on. If DOS encounters a <CTRL>Z (DOS End-of-File Mark) in the data, it interprets this character to indicate that the end-of-file has been reached. Once this has occurred, DOS will not allow any further reads from *STARNBRD* until it is closed and re-opened.

End-of-Read Character

The STARNODE DOS Device Driver at the end of every message sends the End-of-Read character (specified in the *CONFIG.SYS* file) to DOS. If the EOR is set to its default value (<CTRL>Z), STARNBRD will have to be closed and reopened after every attempt to read a message. The default EOR is appropriate when using a software package, and you desire to read only one message at a time.

Many software packages automatically detect EOR characters. For example, when LOTUS 1-2-3 encounters a <CTRL>M, it places the next message received at the start of a new row. Setting the EOR to something like <CTRL>M will allow your software package to read data from the Starnode board until all waiting messages have been read, and then read them as a group.

If you are using a programming language, you may wish to avoid having to open and close STARNBRD after every READ, and therefore may wish to change the EOR character. However, if you do choose to change the EOR, you may have to examine each character received to find the EOR, since DOS will no longer give you an End-of-File condition at the end of each message. Some languages may support *string* input, and may return a string to you as soon as a <CTRL>M or <CTRL>J is detected (so you don't have to look for it).

End-of-Write Character

The End-of-Write character, specified in the *CONFIG.SYS* file, is the character that is used to terminate a send. The default value is <CTRL>Z. however; some software packages do not allow the construction of a message that includes an ASCII control character. You may change the EOW character in the CONFIG.SYS file to a value that your software package allows.

End-of-File Character

The STARNODE DOS Device Driver in response to a READ sends the End-of-File character, specified in the *CONFIG.SYS* file, to DOS when there are no messages waiting on the Starnode board.

The default EOF is <CTRL>Z (DOS End-of-File). If you leave the EOF set to its default value, then you will have to close and re-open STARNBRD every time you empty the received message queue on the Starnode board.

The default EOF is very useful when using a software package, and for most software (such as LOTUS 1-2-3, Paradox, etc.) it should not be changed. If you are importing data into a software package, a <CTRL>Z EOF character should be used, else the software will not detect the end-of-file and may continue trying to read STRNBRD forever without giving you a chance to process collected data.

If you are using a programming language, you will most always wish to change the EOF to something other than the default value (like ETX, CR, or LF). This will allow you to continue reading from the Starnode Board without having to constantly close and re-open STARNBRD.

Using IOCTL Calls

This section describes the methods used to access the Starnode board via standard DOS input/output control calls and he means to make IOCTL calls via Reads and Writes. We also describe how to perform input/output calls to the Starnode Device Driver in order to control the Starnode Board. The C programming interface, and the interfaces used with other languages are described. A listing of the actual calls is given.

Access to the Starnode Board is provided via standard DOS input/output Control (IOCTL) calls on the special character device *STARNBRD*. Opening a file called STARNBRD will return a file descriptor, which may be used to control the Starnode board using the DOS IOCTL Function calls.

C Programming Interface

The format of the header block that must be passed to the IOCTL function is described in the file STARLIB.H, and in the IOCTL routine section Programs written in Microsoft C, may use the header file *STARLIB.H* and the library file *STARLIB.OBJ*, to control the Starnode board.

The interface routines provided in *STARLIB.OBJ* emulate the interfaces provided on the Starnode Development Disk. Microsoft C Small Model Programs written

to use the Starnode interface routines do not require any modifications; they may be re-linked with *STARLIB.OBJ* instead of *STARNMS.OBJ*, causing them to use the DOS Device Driver instead of the original interface routines.

Prior to re-linking, users of Microsoft memory models other than the small model should re-compile *STARLIB.C* for their particular model. Refer to the program *STARLIB.C* to see examples of how to control the Starnode DOS Device Driver using IOCTL calls directly from programs written using C.

Programming Interface for Other Languages

Any programming language or software package that permits file input/output can be used to control the Starnode board via the Read/Write interfaces to the Starnode DOS Device Driver. To perform this, open the file called *STARNBRD*, then write to it or read from it. For more information, refer to "Using Read/Write Interface" on page 6-10.

However, if you require direct access to the interface routines, you will have to implement IOCTL calls from the programming language you are working with. Direct support for IOCTL calls varies depending upon which programming language is used. Consult your programming language's reference manual for the section describing DOS function calls, IOCTL calls, or using INT 21H.

Note: Many programming languages may be able to use the STARN program instead of IOCTL calls by creating a sub-shell of DOS and calling up the STARN program. This is much less efficient, however.

IOCTL Routines

The steps involved in performing an IOCTL call are language dependent. However, generally you will need to set the registers in the PC to the following:

Opening a Device

TABLE 6–3. Opening a Device

Call	Description	Value
AH-	DOS function specifier	03(hex) - Open a Device
AL-	Operation to be performed by the Function	02(hex) - READ/WRITE to/from an open device
BX-	DOS File Descriptor	later use when doing IOCTL calls.
DS-	Segment address of parameter buffer area	
DX-	Offset within segment of parameter buffer area	Point to a NULL terminated string containing the device name.

Executing an IOCTL Call

TABLE 6–4. Executing an IOCTL Call

Call	Description	Value
AH-	DOS function specifier	44(hex) - IOCTL function
AL-	Operation to be performed by the function	03(hex) - WRITE (For IOCTL)
BX-	DOS File Descriptor	This register must be set to the value returned by Function 03. (The Open Call).
DS-	Segment address of parameter buffer area	
DX-	Offset within segment of parameter buffer area	Point to the Starnode buffer area

Starnode Parameter Buffer Area

When an IOCTL call is executed, any data you have placed in the parameter buffer area will be passed to the IOCTL routine. The routine will complete I/O processing, and store any return data in the parameter buffer area.

The format of the Starnode DOS Device Driver IOCTL Parameter Buffer is:

2 bytes	Starnode function request code
2 bytes	Status returned by Starnode
2 bytes	Terminal Number
2 bytes	Terminal Type
2 bytes	Count of bytes used in *data buffer area*
256 bytes	Data buffer area

Starnode Function Request Codes

The list of the valid Starnode Function Request Codes is shown in Table 6–5.

TABLE 6–5. Starnode Function Request Codes

Request Code	Description	Input	Output
0	Initialize the Starnode board	Count = 2 I/O Port address in first 2 bytes of Data Buffer.	Status
1	Clear download table	None	Status
2	Begin a new download table	Terminal type Count = 2 Table length in first 2 bytes of Data Buffer.	Status
3	Next line of download table	Count Download line in Data Buffer.	Status
4	Set date and time	First 6 Data Buffer bytes contain— Month (0 -11) Day (1 - 31) Year (0 = 1900) Hour (0 - 23) Minute (0 - 59) Second (0 - 59)	Status

TABLE 6–5. Starnode Function Request Codes (Continued)

Request Code	Description	Input	Output
5	Receive a Message from the Starnode board	None	Status Terminal Number Terminal Type Count Message in Count Bytes of Data Buffer
6	Send attention signal over network	None	Status
7	Broadcast a message to all Starnode devices	Count (add 1 for Command Byte) Command in 1st byte of Data Buffer Message in remaining bytes of Data Buffer	Status
8	Broadcast a message to all Starnode devices of a type	Terminal Type Count (add 1 for Command Byte) Command in 1st byte of Data Buffer Message in remaining bytes of Data Buffer	Status
9	Send a message to a particular Starnode device	Terminal Number Count (add 1 for Command Byte) Command in 1st byte of Data Buffer Message in remaining bytes of Data Buffer	Status
10	Stop polling, time transmissions on network	None	Status
11	Set I/O Port Address	Count = 2 I/O Port address in first 2 bytes of Data Buffer.	Status

TABLE 6–5. Starnode Function Request Codes (Continued)

Request Code	Description	Input	Output
12	Clear error log	None	Status
13	Get error log	None	Status Count Log in 226 bytes of Data Buffer
14	Get device status	Terminal Number	Status Terminal status in 2 bytes of Data Buffer.
15	Set delay value	Delay Value	Status (always 1)
16	Partially initializes the Starnode board	Count = 2 I/O port address in first 2 bytes of Data Buffer	Status
17	Get Starnode board version and download table size	None	Status Version in 2 bytes: Low order byte = Starnode board version (decimal) Size of Data Buffer High order byte = download table size, where: 1 = 256K bytes download table size 2 = 512K bytes download table size 3 = 1M bytes download table size 4 = 2M bytes download table size
18	Reset Starnode board to factory default state	None	Status

This routine completely resets the Starnode board operating parameters to the factory default settings and runs a full hardware diagnostic test. This may require up to 2 minutes to run. Access to the board is not permitted during this interval.

IOCTL Calls Using Reads & Writes

The Write interface to the Starnode Dos Device Driver supports a subset of the IOCTL routines used to control the Starnode board. Commands written to the driver that are directed to the Starnode board must use the following format:

CNN,P

where:

C—Is the ASCII character *C*

NN—Is the two digit ASCII decimal value representing the IOCTL routine that is to be performed.

P—Is the parameter(s) for the routine; each separated from the previous field by a comma.

Reading from the device STARNSTS can retrieve the status or result of each call.

Examples of each of the IOCTL commands supported via a write and the result of the accompanying read from STARNSTS are provided below. For all examples, the EOW value in the *CONFIG.SYS* file is assumed to be set to its default value 04H.

Example—Initialize Starnode Board

```
COPY CON: STARNBRD
C00<CTRL>D<CTRL>Z
```

This command will initialize the Starnode board at the default I/O base address of 300(hex). The following read indicates the write was accepted and no data was returned:

```
TYPE STARNSTS
0001 0000
```

Example—Clear Download Table

```
COPY CON: STARNBRD
C01<CTRL>D<CTRL>Z
```

This command clears the download table from the Starnode board. The following status indicates that the download command was accepted and no data was returned.

```
TYPE STARNSTS
0001 0000
```

Example—Set Time/Date

```
COPY CON: STARNBRD
C04,0,1,88,3,4,5<CTRL>D<CTRL>Z
```

This command sets the time and date on the Starnode board to 03:04:05 January 1, 1988. The parameters are the month (0-11), day (1-31), year (0=1900), hour (0-23), and minute (0-59). The following status indicates the set-time command was accepted and no data was returned.

```
TYPE STARNSTS
0001 0000
```

Example—Send Attention

```
COPY CON: STARNBRD
C06<CTRL>D<CTRL>Z
```

This command will send an attention signal out over the Starnode network. The following status indicates the attention command was accepted and no data was returned.

```
TYPE STRNSTS
0001 0000
```

Example—Broadcast

```
COPY CON: STARNBRD
C07,49,hello world<CTRL>D<CTRL>Z
```

This command will broadcast *hello world* to all Starnode devices on the network. The first parameter is the command byte (49 - display), followed by the message to be broadcast. The following status indicates that the broadcast command was accepted and no data was returned.

```
TYPE STRNSTS
0001 0000
```

Example—Hello Type

```
COPY CON: STARNBRD
C08,3,49,hello type 3<CTRL>D<CTRL>Z
```

This command will broadcast *hello type 3* to all type 3 Starnode devices on the network. The first parameter is the terminal type, followed by the command byte, followed by the message. The following status indicates that the send command was accepted and no data was returned.

```
TYPE STARNSTS
0001 0000
```

Example—Hello Terminal ID

```
COPY CON: STARNBRD
C09,5,49,hello terminal 5<CTRL>D<CTRL>Z
```

This command will send the message *hello terminal 5* to terminal ID #5. The first parameter is the terminal ID #, followed by the command byte, followed by the message. The following status indicates that the send command was successful and no data was returned.

```
TYPE STARNSTS
0001 0000
```

The following status indicates that the send command was unsuccessful and no data was returned.

```
TYPE STARNSTS
0000 0000
```

Example—Stop Polling

```
COPY CON: STARNBRD
C10<CTRL>D<CTRL>Z
```

This command tells the Starnode board to stop polling the network. The following status indicates the stop command was successful and no data was returned.

```
TYPE STARNSTS
0001 0000
```

Example—Set Base I/O Address of Starnode Board

```
COPY CON: STARNBRD
C11,768<CTRL>D<CTRL>Z
```

This command tells the Starnode board to set the base I/O address of the Starnode board to 768 decimal. The Starnode board is not initialized. The following status indicates that the set port address command was accepted and no data was returned.

```
TYPE STRNSTS
0001 0000
```

Example—Clear Error Table

```
COPY CON: STARNBRD
C12<CTRL>D<CTRL>Z
```

This command clears the error table stored on the Starnode board. The following status indicates that the clear error table command was accepted and no data was returned.

```
TYPE STRNSTS
0001 0000
```

Example—Acquire Error Log Table

```
COPY CON: STARNBRD
C13<CTRL>D<CTRL>Z
```

This command acquires the contents of the Starnode board's error log table. The values are returned as ASCII decimal strings. The following status indicates that the error log request command was accepted and that 173 values are to follow in a table format.

```
TYPE STARNSTS
0001 0173
```

Example—Return Type & Status

```
COPY CON: STARNBRD
C14,1<CTRL>D<CTRL>Z
```

This command causes the Starnode board to return the type and status of terminal ID #1. The following status indicates that the type and status request was accepted, 1 value was returned, and that value was 0259: term type 1 = [int (value/256)] + received a message [Bit 0 set] + sent a message [Bit 1 set].

```
TYPE STARNSTS
0001 0001 0259
```

Example—Set I/O Delay

```
COPY CON: STARNBRD
C15,200<CTRL>D<CTRL>Z
```

This command tells the DOS driver to set the I/O delay to 200 decimal. The following status indicates that the delay command was accepted and no data was returned.

```
TYPE STARNSTS
0001 0000
```

Example—Partial Initialization of Starnode Board

```
COPY CON: STARNBRD
C16,<CTRL>D<CTRL>Z
```

This command partially initializes the Starnode at the default I/O address of 300 (hex). The following status indicates that the write was accepted and no data was returned.

```
TYPE STARNSTS
0001 0000
```

Example—Return Version Number & Download Table Size

```
COPY CON: STARNBRD
C17,<CTRL>D<CTRL>Z
```

This command causes the Starnode board to return the ROM version number and download table size. The following status indicates that the write was accepted and a value was returned indicating:

- Download table size of 2 (0002)
- Starnode version 80 (hex) (0128)

```
TYPE STARNSTS
0001 0001 0002 0128
```

Example—Reset Board to Factory Defaults & Run Diagnostics

```
COPY CON: STARNBRD
C18,<CTRL>D<CTRL>Z
```

This command resets the Starnode board to the factory default state and runs full hardware diagnostics. The diagnostics may need up to 2 minutes to run. During this interval, the board is inaccessible. The following status indicates that the write was accepted and no data was returned.

```
TYPE STARNSTS
0001 0000
```

7

CHAPTER 7 *Writing an Application Program*

This chapter briefly describes each of the Starnode application interface routines. Exact calling parameters of each routine vary according to the language with which it is used. Refer to Chapter 8, “Using Interface Routines” for syntax examples.

Introduction

Should you need to tailor your data collection needs to some specific requirements, you may write your own application program. Such programs can be written using the Microsoft C, version 1.52 or any compiler compatible with a 16 or 32-bit windows DDL. You will need to use the application interface routines described in this chapter to control the Starnode PC Interface Board.

Starnode Application Interface Routines

There are 19 Starnode Application Interface routines. These routines perform the following tasks:

- Initialize the Starnode PC Interface Board
- Partially initialize the Starnode PC Interface Board
- Change the I/O port address used to communicate with the Starnode board
- Clear the download table memory on the Starnode board

- Begin adding a download table to the Starnode board's memory
- Continue adding a download table to the Starnode board's memory
- Send the current date and time to the Starnode board
- Send an attention signal
- Broadcast a message to all Starnode devices on the network
- Broadcast a message to all Starnode devices with a specific assigned terminal type
- Send a message to a specific Starnode device on the network
- Collect a message that was received from the network
- Retrieve the status of any Starnode device on the network
- Clear the error table on the Starnode board
- Retrieve the error table from the Starnode board
- Stop the Starnode board's polling of Starnode devices
- Increase the time delay between transmission of characters to the Starnode board
- Retrieve the Starnode version number and download table size
- Reset the Starnode operating parameters to the factory Default State

Application Interface Routine Descriptions

PINIT—Partially Initialize Controller

PINIT initializes the Starnode board in a manner similar to *STARN init* except that *STARN pinit* does not erase user programs, error and type tables. This routine might be used if you do not desire to re-download a user program to the Starnode board after initializing the Starnode network.

SATTN—Send An Attention Signal

SATTN sends an attention signal to all Starnode devices on the network. This function is used to ensure that a Starnode device receives a broadcast message. If a bar code data collection terminal is in use when a broadcast message is

transmitted, it is possible for the Starnode device to miss the message. The SATTN function disables bar code reading until the next message is sent over the Starnode network, thereby ensuring that all Starnode devices see the broadcast message.

SBRDC—Broadcast A Message

SBRDC broadcasts a message over the Starnode network to all Starnode devices.

SBRDT—Broadcast A Message To Type

SBRDT broadcasts a message over the Starnode network to all Starnode devices, which are assigned a specific terminal type. Valid terminal types are 001 through 127. This routine is more reliable than SBRDC because it tries to send a message up to 10 times to each Starnode device before giving up and moving on to the next Starnode device.

SDLAY—Set Delay Loop Count

PC's that use some form of input/output instruction pipelining or have a bus speed of over 8 MHz may send data too rapidly to the Starnode board. This can result in lost messages, garbled data, or error returns from a number of interface routines.

The SDLAY routine sets the length of a delay loop, which occurs after every in or out instruction. The default loop count is 1. If you are experiencing erratic Starnode board behavior, set the count to a value between 2 and 50.

SERRCL—Clears Error Table

SERRCL clears the 226 bytes of data in the error table. Refer to Figure 5–22, "Starnode Status Table Dialog Box," on page 5-21 for more information.

SERRGT—Retrieve Error Table

SERRGT retrieves the 226 bytes of data in the error table. Refer to Figure 5–22, "Starnode Status Table Dialog Box," on page 5-21 for more information.

SGETMEM—Get Memory Configuration

Use SGETMEM to retrieve the current configuration on the Starnode Controller.

SINIT—Initialize Controller

SINIT initializes the controller, clears all error tables, resets all terminal status and type values, and instructs the board to request all Starnode devices on the

network to send their type to the board. Data in the received message queue is not deleted.

The board will not begin polling the network or sending time messages until SINIT has been called. Once SINIT has been called, the board will continue polling and sending time messages until either SSTOP is called or the PC is powered off.

SINIT also tells the other interface routines which I/O port address to use. No message may be sent or received over the network until this routine has been called. SINIT returns nine possible values in the status parameter as shown in Table 7–1.

TABLE 7–1. Status Parameter Values

Value	Description
1	Starnode Board has passed all diagnostics and is operating normally.
0	There is no response from the Starnode Board.
-1	ROM failed the power-up CRC check.
-2	Bad internal CPU RAM.
-3	Bad RAM chip U19.
-4	Bad RAM chip U3.
-5	Bad RAM chip U4.
-6	Bad Real-time clock chip U5.
-7	Bad Backup Battery chip U9.

SRECV—Receive A Message

SRECV checks the Starnode PC Interface Board to see if any messages have been received. If the board has messages waiting, the first message in the queue will be sent to the calling program. The Starnode PC Interface Board has a buffer capacity of 16K characters for messages. If the incoming message buffer fills up, the Starnode PC Interface Board will suspend polling the Starnode devices on the network until there is room in the buffer for an incoming message.

If no message is waiting, the terminal ID number and type parameters will be set to 0. Therefore, if a status of 1 is returned, and the terminal number is 0, there are no messages waiting on the Starnode PC Interface Board.

SRESET—Reset Controller Parameters To Factory Default

This routine completely resets the Starnode board setup parameters to the factory default settings, and runs full hardware diagnostics. Diagnostics may require up to two minutes to complete. Access to the Starnode board during this time is not possible.

SSEND—Send A Message

SSEND sends a message over the Starnode network to a specific terminal number. If the Starnode device does not acknowledge the message, the Starnode board will attempt to re-transmit the message (up to nine additional times). If all ten attempts fail, or if there are problems communicating with the Starnode board, SSEND will return a status value of 0.

Usually, a Status value of 0 is the result of either attempting to send a message to a terminal ID that is not on the network, or attempting to send multiple messages to a Starnode device that has not had time to digest the first message.

A command is sent with each message to tell the Starnode device what to do with the message. If a command value greater than 64 (hexadecimal 40) is sent, the Starnode device will pass the message to the program that is running in the Starnode device. If such messages are being sent, the program running in the Starnode device must be written to retrieve messages arriving over the Starnode network.

SSETMEM—Set Memory Configuration

Additional buffer space may be made available to buffer either or both of received messages and priority messages by reducing the number of download pages. Run SSETMEM specifying a larger input size and a correspondingly smaller download pages size.

SSIOP—Set I/O Port Address

SSIOP sets the I/O Port Address that the interface routines will use; it does not re-initialize the board. If you are calling SINIT to initialize the board and set the I/O port address, this routine does not serve any purpose. However, if you have left the board operating while the program that called SINIT exits to DOS, then another program is loaded in, this routine allows the second program to address the board correctly. If the board is not present at the address provided, the routine will return a status value of 0. This routine also allows two Starnode PC Interface Boards to be installed in the same PC.

SSTAT—Status Of Data Collection Terminal

SSTAT retrieves two bytes of status information for the specified Starnode device from the board's status tables. They contain information on the terminal type, WACK State, program download, and errors.

SSTOP—Stop Network

SSTOP tells the controller to stop accepting messages from the network and to stop sending time information to all devices. Note that there may still be uncollected messages in the message buffer on the Starnode PC Interface Board, so you should continue to call SRECV after SSTOP has been called until no messages are returned.

STBLB—Begin Download Table

STBLB tells the controller that a new download table is about to be sent to it. It tells the board what terminal type the table is for and how many lines will be in the table (how many times STBLN will be called).

If the Starnode PC Interface Board has already been given a download table for a terminal type, but you now wish to replace that table, you may send the new definition to the board without clearing the table. The old definition will be replaced by the new definition.

After the new table is completely downloaded, the board may require several seconds to delete the old version and re-pack its storage areas.

STBLC—Clear Download Tables

Calling STBLC will clear all download tables that are stored on the controller. All download table space will be available after this routine is called.

STBLN—Next Line Of Download Table

STBLN supplies the next line of a download file to the Starnode controller. STBLB must have been called before this routine is used. This routine should then be called numlines (the parameter supplied to STBLB) times in order to complete the transfer of a download file for a particular terminal type.

Approximately 256K, 512K, 1M, or 2M bytes of *.LAN* file data may be downloaded to the Starnode board, depending upon the Starnode device used. Use the file sizes listed by the DOS directory command, not the program size information listed by the user program compilers, when estimating the program storage capacity of the Starnode board.

Note: You will not receive any indication that you have downloaded too much data to fit on the Starnode board. If you download too much information, the programs that could not be stored will not be downloaded to the Starnode device.

STIME—Sets Time On Controller

STIME retrieves the date and time from DOS, and sends them to the Starnode controller. This ensures the date and time sent by the Starnode controller to the Starnode devices is synchronized with that of the PC. This routine should be called after initializing the board, then periodically thereafter (once or twice a day) to ensure that the clocks remain synchronized. Note that you may have to use the DATE and TIME DOS commands to set the PC's clock to the desired time.

Note: Some versions of DOS prior to Version 3.1 may fail to update the date if the PC is left running 24 hours a day without rebooting. Upgrading to a more recent version of DOS will correct this problem.

SVER—Retrieve Starnode Version Number & Table Size

SVER returns two bytes of information. The low order byte contains the Starnode board firmware version number in hex format, while the high order byte contains a number representing the size of the download table as shown in Table 7–2.

TABLE 7–2. Download Table Size

1	256K bytes download table size
2	512K bytes download table size
3	1M bytes download table size
4	2M bytes download table size

Developing Application Programs

Starnode drivers are sold separately for OS/2, Windows NT, and SCO UNIX. To build an application for these environments, please refer to the manual that came with your driver.

The following directions may be used to build applications under the environments supported by the Starnode Developers Disk, which is shipped with the Starnode controller.

- 32-bit MS Visual C++ 4.0 and above on Windows 95
  - Include *starndll.h* in any source module that calls a Starnode Interface Routine
  - Link with *starn95.lib*
  - Make sure *starn95.dll* is in your \windows\system directory
- 32-bit MS Visual Basic 4.0 and above on Windows 95
  - Include *strndemo.bas* in your project
  - In *strndemo.bas* set #Const WIN95 = 1 “Set to Non Zero Value for Windows 95”
  - Make sure *starn95.dll* is in your \windows\system directory
- 16-bit MS Visual C++ 1.52 or below on Windows 3.0 + 3.11
  - Include *starndll.h* in any source module that calls a Starnode Interface Routine
  - Link with *starndll.lib*
  - Make sure *starndll.dll* is in your \windows\system directory
- 16-bit MS Visual Basic 4.0 or below on Windows 95
  - Include *strndemo.bas* in your project
  - In *strndemo.bas* set #Const WIN311 = 1 “Set to Non Zero Value for Windows 3.11 or 3.1”
  - Make sure *starndll.dll* is in your \windows\system directory
- 16-bit MS C or C++ for DOS Platform
  - Include *starlib.h* in any source module that calls a Starnode Interface Routine

– Link with the items shown in Table 7–3.

TABLE 7–3. Language Object Module Names

Object Module Name	Application
STARNMS.OBJ	Microsoft or TURBO C (Small)
STARNMC.OBJ	Microsoft or TURBO C (Compact)
STARNMM.OBJ	Microsoft or TURBO C (Medium)
STARNML.OBJ	Microsoft or TURBO C (Large)

CHAPTER 8 *Using Interface Routines*

This chapter provides an alphabetical reference of the available interface routines. Included with the description of the routine, you will find syntax models and programming examples that demonstrate the routines in use.

Introduction

SICK Auto Ident, Inc. currently offers interface routines for the following programming languages:

- Any language compatible with Microsoft *.obj* modules that use the *C* calling convention
- Any language that supports 16 or 32-bit windows *.dll*'s including:
  - Microsoft Visual C++
  - Microsoft Visual Basic

In general, the routines provided are used as follows:

- Initialize the Starnode PC Interface Board
- Partially initialize the Starnode PC interface board
- Change the I/O port address used to communicate with the Starnode board
- Clear the download table memory on the Starnode board

- Begin adding a download table to the Starnode board's memory
- Continue adding a download table to the Starnode board's memory
- Send the current date and time to the Starnode board
- Send an attention signal
- Broadcast a message to all Starnode devices on the network
- Broadcast a message to all Starnode devices with a specific assigned terminal type
- Send a message to a specific Starnode device on the network
- Collect a message that was received from the network
- Retrieve the status of any Starnode device on the network
- Clear the error table on the Starnode board
- Retrieve the error table from the Starnode board
- Stop the Starnode board's polling of Starnode devices
- Increase the time delay between transmission of characters to the Starnode board
- Retrieve the Starnode version number and download table size
- Reset the Starnode operating parameters to the factory Default State.

Interface Routine Parameters

When a parameter varies according to the routine, it will be described according to its use within each routine.

C Parameters

This section provides a general description of the parameters used in C interface routines. Refer to *starndll.h*, *starn95.h*, and "Developing Application Programs" on page 7-7 for specific parameter types and function prototypes.

command - (used in SSEND, SBRDC, SBRDT)—This parameter is a constant or variable of type *char*, and contains the command value to be sent preceding the

message. This byte is received by the terminal as the first character of the message, and determines what the terminal will do with this message.

delay - (used in SDLAY)—This parameter is a constant or variable of type short that contains the number of NOP's to execute after every in or out instruction.

length - (used in STBLN, SSEND, SBRDC, SBRDT)—This parameter is a constant or variable of type *short*, which contains the length of the message to be sent to the terminal. Do not include the command byte in this count! The maximum value for this parameter is 126.

&length - (used in SRECV)—This parameter is a pointer to a short integer that will contain the length of the received message (in bytes).

port - (used in SINIT, SIOP)—This parameter is a constant or variable of type *short* containing the I/O port address that the interface routines should use to communicate with the Starnode PC interface board. This address (normally hexadecimal 300) must correspond with the jumper settings on the board.

&string[?] - (used in SRECV, SSEND, SBRDT, SBRDC, STBLN)—The use of this parameter varies according to the routine, which is being called.

When calling the SRECV routine, &string[?] is a pointer to the array of characters that will contain the received message. It must be defined to be at least 128 characters long, e.g. char array[128]. The first length characters in the string will be altered to contain the received message. A NULL will be placed in the length+ 1 character position.

When calling the SSEND, SBRDT, SBRDC, or STBLN routines, &string[?] is a pointer to the first character of the array containing the message to be sent to the terminal. The array itself does not have to be NULL terminated since only length characters will be sent.

&stat - (used in SSTAT)—This parameter is a pointer to a short integer that will contain the status data for the selected terminal after calling SSTAT.

termnum - (used in SSEND, SSTAT)—This parameter is a constant or variable of type *short,* which contains the terminal number of the terminal that this message is to be sent to.

&termnum - (used in SRECV)—This parameter is a pointer to a short integer that will contain the terminal number of the terminal that sent the message. This variable will always be modified by SRECV, and will be set to 0 if there is no message waiting.

type - (used in SBRDT, STBLB)—This parameter is a constant or variable of type short containing the type of the terminal to receive this broadcast or download.

&type - (used in SRECV)—This parameter is a pointer to a short integer that will contain the terminal type of the terminal that sent the message. This variable will always be modified by SRECV, and will be set to 0 if there is no message waiting.

&romver - (used in sver)—This parameter is a pointer to a short integer that contains the Starnode PC ROM version and the download table size after SVER is called.

Basic Parameters

This section provides a general description of the parameters used in the BASIC interface routines. Refer to *starndemo.bas* and "Developing Application Programs" on page 7-7 for specific parameter types and function prototypes.

COMMAND% - (used in SBRDC, SBRDT, SSEND)—This parameter is an integer constant or variable containing the value of the command byte that is to be sent preceding this message. This byte is received by the terminal as the first character of the message, and determines what the terminal will do with this message.

DELAY% - (used in SDLAY)—This parameter is an integer constant or variable of type *short* that contains the number of NOP's to execute after every in or out instruction.

ERRSTR$ - (used in SERRGT)—String variable, which must have a current length of 226 or more (e.g., ERRSTR$ = SPACE$(227), into which the error table is placed by SERRGT.

LENGTH% - (used in SRECV)—This parameter is an integer variable that will contain the length of the received message, in bytes.

NUMLINES%—This parameter is an integer constant or variable containing the number of lines in the download table.

PORT% - (used in SINIT, SIOP)—This parameter is an integer constant or variable that contains the I/O port address that the interface routines should use to communicate with the Starnode PC interface board. This address, normally hexadecimal 300, must correspond with the jumper settings on the board.

STATDAT% - (used in SSTAT)—This parameter is an integer variable that will contain the status data for the selected terminal after calling SSTAT.

STATUS%—This parameter is an integer variable. Its value indicates whether or not the function was executed successfully. The variable will either be set to a 1 if successful, or a 0 if NOT successful (generally, problems communicating with the Starnode board).

STRING$ - (used in SBRDC, SBRDT, SSEND, SRECV, STBLN)—In the STBLN routine, this parameter is a string constant or variable containing the next line of the download table. In the SBRDC, SBRDT, and SSEND routines, this parameter is a string constant or variable that contains the message to be sent. In the SRECV routine, this parameter is a string variable that will contain the message. It must be pre-defined to contain at least 127 characters, e.g. STRING$ = SPACE$(128). The first LENGTH% characters in the string will be altered to contain the received message.

TERMNUM% - (used in SSTAT, SSEND, SRECV)—In the SSTAT and SSEND routines, this parameter is an integer constant or variable that contains the terminal number of the data collection terminal. In the SRECV routine, this parameter is an integer variable that will contain the terminal number of the data collection terminal that sent the message.

TYPE%—In the SBRDT and STBLB routines, this parameter is an integer constant or variable containing the type of the terminal to receive the broadcast or download. In the SRECV routine, this parameter is an integer variable that will contain the terminal type of the data collection terminal that sent the message.

ROMVER%—This parameter is an integer variable that contains the Starnode PC ROM version, and the download table size after SVER is called.

Interface Routines

Syntax models and programming examples are provided for each of the two languages supported, C and BASIC.

PINIT

Partially Initialize the Starnode board

Upon completion, the function will have a value of 1 (TRUE) if it completed successfully or 0 through -7 if there was a problem in communicating with the Starnode board. If a value of 0 is returned, try calling SINIT() a second time.

Return values -1 through -5 indicate hardware diagnostic failures.

Return values -6 or -7 are warning messages. The Starnode board is initialized and in operational state.

C Language

SYNTAX
PINIT(port);

EXAMPLE 1

```
short port;
port = 0x300;
stat = PINIT (port);
switch (stat)
    case 1:printf("board on );
           break;
    case -1:
    case -2:
    case -3:
    case -4:
    case -5:
    case -6:
    case -7: printf ("Hardware failure on board%D \n",stat);
           break;
    default: printf("Problem with Starnode board \n");
```

Example 2

```
if (!PINIT(0x300))
    if (!PINIT(0x300))
           printf("Problem with Starnode board. \n");
```

Visual Basic

SYNTAX
STATUS% = PINIT (PORT%)

Example

```
Dim STATUS%
Dim PORT%

PORT% = &H3A0     ' Starnode controller at address 0x3A0
STATUS% = PINIT(PORT%)   ' Perform partial initialization

Select Case STATUS%
   Case Is = 0
          MsgBox ("Problem with Starnode board.")
   Case Is = -1
          MsgBox ("ROM Checksum error")
   Case Is = -2
          MsgBox ("CPU RAM error")
   Case Is = -3
          MsgBox ("RAM error U19")
   Case Is = -4
          MsgBox ("RAM error U3")
   Case Is = -5
          MsgBox ("RAM error U4")
   Case Is = -6
          MsgBox ("Real Time clock U5 error")
   Case Is = -7
          MsgBox ("Backup Battery error U9")
   Case Else
          MsgBox ("Successful Initialization")
End Select
```

SATTN

Send an Attention Signal

C Language

The SATTN routine returns a value of 1 (TRUE) if it completes successfully or 0 (FALSE) if there is a problem in communicating with the Starnode board.

SYNTAX
SATTN();

Example

```
if (!SATTN( ))
   printf("Problem with Starnode board. \n");
   if ( !SBRDC (49, 11, Break time!))
          printf ("Problem with Starnode board \n");
```

Visual Basic

SYNTAX
STATUS%=SATTN()

Example

```
Dim STATUS%)

STATUS%=SATTN()  'send with attention

IF (STATUS% = False) Then
   MsgBox ("Problem with Starnode board.")
End If
```

SBRDC

Broadcast Message to All

The function will, upon completion, have a value of 1 (TRUE) if successful or 0 (FALSE) if there was a problem communicating with the Starnode board.

C Language

SYNTAX
SBRDC(command,length, &string[?]);

Example 1

```
char msg[50];
short len;
len = sprintf(msg,"BREAK TIME!");
comm = 49;
if (!SBRDC(comm,len,msg))
   printf("Problem with Starnode board. \n");
```

Example 2

```
if (!SBRDC(49, 11, BREAK TIME!))
   printf("Problem with Starnode board. \n");
```

Visual Basic

The value returned by the SBRDC function will be placed in the STATUS% variable.

SYNTAX
STATUS%=SBRDC(COMMAND%,LENGTH%, STRING_VAR$)

Example

```
Dim COMMAND%
Dim STRING_VAR$
Dim LENGTH%
Dim STATUS%

COMMAND% = 49       ' Prompt command
STRING_VAR$ = "BREAK TIME"
LENGTH% = 10    ' 10 character message

STATUS% = SBRDC(COMMAND%, LENGTH%, STRING_VAR$)

If (STATUS% = False) Then
   MsgBox ("Problem with Starnode board.")
End If
```

SBRDT

Broadcast Message to Type

Upon completion, the function will have a value of 1 (TRUE) if it completed successfully or 0 (FALSE) if there was a problem in communicating with the Starnode board.

C Language

SYNTAX
SBRDT(termtype,command,length, & string[?]);

Example 1

```
char msg[50];
short len;
len = sprintf(msg,"BREAK TIME!");
comm = 49;
if (!SBRDT(122,comm,len,msg))
   printf("Problem with Starnode board. \n");
```

Example 2

```
if (!SBRDT(122,49, 11, BREAK TIME!))
   printf("Problem with Starnode board. \n");
```

Visual Basic

Upon completion, this routine places a value of 1 or 0 into the STATUS% variable.

SYNTAX
STATUS% = SBRDT (TYPE_NUM%, COMMAND%, LENGTH%, STRING_VAR$)

Example

```
Dim COMMAND%
Dim STRING_VAR$
Dim LENGTH%
Dim STATUS%
Dim TYPE_NUM%

COMMAND% = 49       ' Prompt command
STRING_VAR$ = "BREAK TIME"
LENGTH% = 10    ' 10 character message
TYPE_NUM% = 1' Send to type 1

STATUS% = SBRDT (TYPE_NUM%, COMMAND%, LENGTH%,
STRING_VAR$)

If (STATUS% = False) Then
   MsgBox ("Problem with Starnode board.")
End If
```

SDLAY

Set Delay After I/O

Upon completion, the function will have a value of 1 (TRUE).

C Language

SYNTAX
SDLAY(delay);

Example

```
SDLAY(2);
```

Visual Basic

SYNTAX
STATUS% = SDLAY(DELAY%)

Example

```
Dim STATUS%
Dim DELAY%

DELAY% = 2
STATUS% = SDLAY(DELAY%)
```

SGETMEM

Get Memory Configuration

This routine returns the status of prior SSETMEM attempts and the number of message buffers or priority buffers. Refer to “SSETMEM” on page 8-27 for more information.

sgetmem() returns:

- Config[0] = status of prior ssetmem attempt (if any); 0 = no error
- Config[1] = #of 512 byte buffers for normal input messages; 1-63
- Config[2] = #of 512 byte priority buffers; 1-63 (total input+priority=64)
- Config[3] = #of 16K download buffers taken for receive messages

C Language

Syntax

sgetmem(ucConfig);

Example

```
unsigned char ucConfig[4];
if(!sgetmem(ucConfig)) {
    printf("Problem with STARNODE network");
    }
else {
    if(ucConfig[0]) {
            printf("Last memory configuration attempt failed\n");
            }
    else {
            printf("Last memory configuration attempt succeeded\n");
            }
    printf("Current input buffer setting is: %d\n", (short)ucConfig[1]);
    printf("Current priority buffer setting is:%d\n",
(short)ucConfig[2]);
    printf("Current program buffer setting is:%d\n", (short)ucConfig[3]);
    }
```

Visual Basic

SYNTAX
STATUS% = SGETMEM (CONFIG$)

Example

```
Dim DOWNLOADS%
Dim STATUS%
Dim CONFIG$

CONFIG$ = Space$(3)
STATUS% = SGETMEM(CONFIG$)

If (STATUS% = False) Then
   MsgBox ("Problem with Starnode board.")
Else
   DOWNLOADS% = Asc(Mid$(CONFIG$, 3, 1))
   MsgBox CStr(DOWNLOADS%) + " program download buffers
reserved
for receiving"
End If
```

SERRCL

Clear Error Table

C Language

SYNTAX
SERRCL();

Example

```
if (!SERRCL());
   printf("Problem communicating with the Starnode
board);\n");
```

Visual Basic

SYNTAX
STATUS% = SERRCL()

Example

```
Dim STATUS%

STATUS% = SERRCL()      ' clear the error table

If (STATUS% = False) Then
   MsgBox ("Problem with Starnode board.")
End If
```

SERRGT

Get Error Table

Refer to the program STARNTST on the Starnode PC disk for a complete example of how to print out the error table.

C Language

SYNTAX
SERRGT (&errbuf.errstring[0]);

Example

```
union errunion {
    char errstring [226];
    struct errstruc {
           char proberr[64], probecnt[64];
                  short
xnak[10],xtimeo[10],xbadch[10],rnak[10];
                  long revnum, sendnum;
                  long probehigh, probelow;
                  short badterm;
    } errtbl;
} errbuf;
```

Visual Basic

SYNTAX
STATUS% = SERRGT(ERRSTR$)

Example

```
Dim LASTBAD%
Dim STATUS%
Dim ERRSTR$

ERRSTR$ = Space$(227)
STATUS% = SERRGT(ERRSTR$)

If (STATUS% = False) Then
   MsgBox ("Problem with Starnode board.")
Else
   LASTBAD% = Asc(Mid$(ERRSTR$, 225, 1))
   If LASTBAD% <> 0 Then
          MsgBox ("Last Failed Send Was to ID ") +
CStr(LASTBAD%)
   End If
End If
```

SINIT

Initialize the Starnode board

Upon completion, the function will have a value of 1 (TRUE) if it completed successfully or 0 through -7 if there was a problem in communicating with the Starnode board. If a value of 0 is returned, try calling SINIT() a second time.

Return values -1 through -5 indicate hardware diagnostic failures.

Return values -6 or -7 are warning messages. The Starnode board is initialized and in operational state.

C Language

SYNTAX
SINIT(port);

Example 1

```
short port;
port = 0x300;
stat = SINIT (port);
switch (stat)
    case 1:printf("board on );
           break;
    case -1:
    case -2:
    case -3:
    case -4:
    case -5:
    case -6:
    case -7: printf ("Hardware failure on board%D
\n",stat);
           break;
    default: printf("Problem with Starnode board \n");
```

Example 2

```
if (!SINIT(0x300))
    if (!SINIT(0x300))
           printf("Problem with Starnode board. \n");
```

Visual Basic

SYNTAX
STATUS% = SINIT (PORT%)

Example

```
Dim STATUS%
Dim PORT%

PORT% = &H3A0    ' Starnode address is 0x3A0
STATUS% = SINIT(PORT%) ' initialize the controller

Select Case STATUS%
   Case Is = 0
          MsgBox ("Problem with Starnode board.")
   Case Is = -1
          MsgBox ("ROM Checksum error")
   Case Is = -2
          MsgBox ("CPU RAM error")
   Case Is = -3
          MsgBox ("RAM error U19")
   Case Is = -4
          MsgBox ("RAM error U3")
   Case Is = -5
          MsgBox ("RAM error U4")
   Case Is = -6
          MsgBox ("Real Time clock U5 error")
   Case Is = -7
          MsgBox ("Backup Battery error U9")
   Case Else
          MsgBox ("Successful Initialization")
End Select
```

SIOP

Set I/O Port Address

Upon completion, the function returns a value of 1 (TRUE) upon successful completion or 0 (FALSE) if there was a problem in communicating with the Starnode board, or if the board was not found at the specified address.

C Language

SYNTAX
SIOP(port);

Example

```
status = SIOP(0x3A0);
if(status != 1)
   printf(" \nStarnode board not responding at address
3A0.");
```

Visual Basic

SYNTAX
STATUS% = SIOP (PORT%)

Example

```
Dim STATUS%
Dim PORT%

PORT% = &H3A0
STATUS% = SIOP (PORT%)     ' Check the controller
If (STATUS%  = False) Then
   MsgBox ("Starnode board is not responding at 3A0")
End If
```

SRECV

Get Message from Starnode board

Upon completion, the function will have a value of 1 (TRUE) if it completed successfully, or 0 (FALSE) if there was a problem in communicating with the Starnode network.

C Language

SYNTAX
SRECV(&termnum, &type, &length, &string[?]);

Example

```
short termnum, type, length;
char string[128];
if (!SRECV(&termnum, &type, &length, &string[?]))
   printf("Problem with Starnode network.");
else if (termnum != 0)
   printf("%s /n", string);
```

Visual Basic

SYNTAX
STATUS% = SRECV(TERMNUM%, TYPE_NUM%, LENGTH%, STRING_VAR$)

Example

```
Dim STRING_VAR$
Dim LENGTH%
Dim STATUS%
Dim TERMNUM%
Dim TYPE_NUM%

STRING_VAR$ = Space$(128)
STATUS% = SRECV(TERMNUM%, TYPE_NUM%, LENGTH%,
STRING_VAR$)
If (STATUS% = False) Then
   MsgBox ("Error communicating with controller")
Else
   If LENGTH > 0 Then
          MsgBox ("ID " + Format(TERMNUM%) + " sent " +
STRING_VAR$)
   End If
End If
```

SRESET

Reset To Factory Default Settings

Upon completion, the function will have a value of 1 (TRUE) if it completed successfully, or 0 (FALSE) if there was a problem in communicating with the Starnode board. The function performs full diagnostics with all data being lost. The diagnostics may require up to 2 minutes to complete.

C Language

SYNTAX
SRESET();

Example

```
if (!SRESET())
   printf("Problem with Starnode board. \n");
```

Visual Basic

SYNTAX
STATUS% = SRESET ()

Example

```
Dim STATUS%

STATUS% = SRESET ()

If (STATUS% = False) Then
   MsgBox ("Problem with Starnode board.")
End If
```

SSEND

Send Message to Terminal

Upon completion, the function will have a value of 1 (TRUE) if it completed successfully, or 0 (FALSE) if NOT successful (the terminal was busy and did not receive this message in ten transmission attempts). For REALIA COBOL, the routine returns the value 1 or 0 to the RETURN-CODE variable.

C Language

SYNTAX
SSEND(termnum, command, length, &string[?]);

Example 1

```
short termnum, command, length;
char string[80];
termnum = 150;
command = 49;
length = sprintf(string, "BREAK TIME!");
SSTAT(termnum, &stat);
if ((stat & 0x0080) == 0)
   if (!SSEND (termnum, command, length, string))
          printf("Error...);
else
   printf("Terminal in WACK state \n");
```

Example 2

```
if (!SSEND(150, 49, 11, "BREAK TIME!"))
   printf("Error sending message to termnum 150");
```

Visual Basic

SYNTAX
STATUS% = SSEND(TERMNUM%, COMMAND%, LENGTH%, STRING_VAR$)

Example

```
Dim LENGTH%
Dim STATUS%
Dim TERMNUM%
Dim STRING_VAR$
Dim COMMAND%

COMMAND% = 49   'Prompt command
TERMNUM% = 67   ' Starnode ID address
STRING_VAR$ = "Break Time"  ' message to display
LENGTH% = 10

STATUS% = SSEND(TERMNUM%, COMMAND%, LENGTH%, STRING_VAR$)

If (STATUS% = False) Then
   MsgBox ("Error sending to termnum " + Format
(TERMNUM%))
```

SSETMEM

Set Memory

This routine inputs the number of priority or message buffers.

ssetmem() input is:

- Config[0] = #of 512 byte buffers for normal input messages; 1-63
- Config[1] = #of 512 byte priority buffers; 1-63 (total input+priority=64)
- Config[2] = #of 16K download buffers taken for receive messages

Input message buffer and Priority message buffer share a pool of 64 buffers. (Each buffer is 512 bytes.) You must set a minimum of 1 buffer for normal input messages (default = 63) and 1 buffer for priority messages (default = 1).

Input Page Buffers (16K each) configures the amount of memory to take from the user program download area and use for input messages (default = 0). Refer to Table 8–1 for the maximum you can take for each system option.

Maximum settings for parameter 3 (Input Page Buffers) as listed in Table 8–1.

TABLE 8–1. Parameter 3—Maximum Settings

System with...	Maximum Input
2 MB option	64
1 MB option	32
512K option	16

If you input invalid values for any of the three settings, you will cause the memory to reset to its defaults.

Note: Starnode ROM 9.2 or higher is required. Contact SICK Auto Ident, Inc. technical support if you need to upgrade your ROM.

Ssetmem—

- Flushes all downloaded programs
- Flushes all buffered messages

- Stops network polling
- Requires 2 minutes to complete settings in memory
- Use Ssetmem only once.

C Language

Syntax
ssetmem(ucConfig);

Example

```
unsigned char ucConfig[4];
ucConfig[0] = 62;                       // 512*62 input buffers
ucConfig[1] = 2;                        // 512*2 for priority messages
ucConfig[3] = 2;                        // 16K*2 additional input buffers
if(!ssetmem(ucConfig)) {
    printf("Problem with STARNODE network");
    }
else {
    while(TRUE) {                       // Could take 2 minutes to recover
            if(!sgetmem(ucConfig)) continue;
            if(ucConfig[0]) {
                    printf("Memory configuration attempt failed\n");
                    }
            else {
                    printf("Memory configuration succeeded\n");
                    }
            break;
            }
    }
```

Visual Basic

SYNTAX
STATUS% = SSETMEM (CONFIG$)

Example

```
Dim CONFIG$
Dim STATUS%

' Restore factory default memory configuration
CONFIG$ = Chr$(1) + Chr$(0) + Chr$(63)
STATUS% = SSETMEM(CONFIG)

If (STATUS% = False) Then
   MsgBox ("Error configuring memory on controller")
End If
```

SSTAT

Get Terminal Status

The value returned by this function will consist of a 15-bit integer. The high-order 7 bits (8 - 15) contain the terminal type. The low order 7 bits representations' are shown in Table 8–2.

TABLE 8–2. Terminal Status Interpretation

Bit #	Value	Description
0 (LSB)	1	Starnode board has successfully received data from the data collection terminal
1	2	Starnode board has sent data to the data collection terminal successfully.
2	4	Starnode board has received a NAK, bad character, or has timed-out when sending a message to the data collection terminal.
3	8	Starnode board has sent a NAK to the data collection terminal when the data collection terminal was attempting to send the board a message.
4	16	In the process of receiving a message from the data collection terminal, the Starnode board has exceeded the maximum number of NAK's (10) it is supposed to send and aborted out of the attempt.
5	32	In the process of sending a message to the data collection terminal, the Starnode board has exceeded the maximum number of NAK's (10) it is supposed to receive and aborted out of the attempt.
6	64	A user program was successfully downloaded to the data collection terminal.
7(MSB)	128	The data collection terminal is in a WACK (Wait Acknowledge) state. This means that the data collection terminal is busy processing the last message. This bit will be cleared automatically when the terminal indicates that it is free again.

C Language

SYNTAX
SSTAT (termnum, &stat);

Example

```
if (!SSTAT(termnum, &stat))
   printf ("Problem with Starnode board \n");
else
   printf("Status is: %4X", stat);
```

Visual Basic

SYNTAX
STATUS% = SSTAT (TERMNUM%, STATDAT%)

Example

```
Dim STATUS%
Dim STATDAT%
Dim TERMNUM%

TERMNUM% = 67' Starnode address ID is 67

STATUS% = SSTAT (TERMNUM%, STATDAT%)

If (STATUS% = False) Then
   MsgBox ("Problem with Starnode board.")
End If
```

SSTOP

Stop the Network

Upon completion, the function will have a value of 1 (TRUE) if it completed successfully or 0 (FALSE) if there was a problem in communicating with the Starnode board.

C Language

SYNTAX
SSTOP();

Example

```
if (!SSTOP())
printf("Problem with Starnode board.\n");
```

Visual Basic

SYNTAX
STATUS% = SSTOP()

Example

```
Dim STATUS%

STATUS% = SSTOP()

If (STATUS% = False) Then
   MsgBox ("Problem with Starnode board.")
End If
```

STBLB

Begin Download Table

Upon completion, the function will have a value of 1 (TRUE) if it completed successfully or 0 (FALSE) if there was a problem in communicating with the Starnode board.

C Language

SYNTAX
STBLB(type, numlines);

Example

```
FILE *pclfile;
short i, rec_count, block_count;
char buf[128];
pclfile = fopen(PPICK.LAN, 'rb');
rec_count = (short) fgetc(pclfile);
rec_count = rec_count | ((short) fgetc(pclfile) < 8);
if (!STBLB(25, rec_count))
    printf("Problem with Starnode board. \n");
else
for (i=0; i<rec_count; i++) {
    block_count = (short) fgetc(pclfile);
    fread(buf, block_count, 1, pclfile);
    if (!STBLN(block_count, buf)) {
            printf("Problem with Starnode board.\n");
            break;
    }
}
fclose(pclfile);
```

Visual Basic

SYNTAX
STATUS%=STBLB(TYPE_NUM%, RECORD_COUNT%)

Example

```
Dim STATUS%
Dim RESULTS%
Dim RECORD_COUNT%
Dim TYPE_NUM%
Dim INBUF$
Dim I%
Dim BYTE1 As Byte
Dim BYTE2 As Byte

TYPE_NUM% = 1
Open "gen_civ.lan" For Binary As #98
Get #98, , BYTE1
Get #98, , BYTE2
RECORD_COUNT% = BYTE1 + (BYTE2 * 256)

STATUS% = STBLB(TYPE_NUM%, RECORD_COUNT%)

If (STATUS% <> 1) Then _
   Print "ERROR: non-1 return from stblb: "; STATUS%:
GoTo STBLB1
For I% = 1 To RECORD_COUNT%
   Get #98, , BYTE1
   INBUF$ = String(BYTE1, " ")
   Get #98, , INBUF$
   STATUS% = STBLN(BYTE1, INBUF$)
   If (STATUS% <> 1) Then _
          Print "ERROR: non-1 return from stbln: ";
STATUS%: GoTo STBLB1
Next I%
Close #98

STBLB1:
RESULTS% = STATUS%
```

STBLC

Clear Download Table

Upon completion, the function will have a value of 1 (TRUE) if it completed successfully or 0 (FALSE) if there was a problem in communicating with the Starnode board.

C Language

SYNTAX
STBLC();

Example

```
if (!STBLC())
   printf("Problem with Starnode board. \n");
```

Visual Basic

SYNTAX
STATUS% = STBLC()

Example

```
Dim STATUS%

STATUS% = STBLC()
```

STBLN

Next Line of Download Table

C Language

SYNTAX
STBLN(length, &string[?])

Example

```
FILE *pclfile;
short i, rec_count, block_count;
char buf[128];
pclfile = fopen(PPICK.LAN, 'rb');
rec_count = (short) fgetc(pclfile);
rec_count = rec_count | ((short) fgetc(pclfile) < 8);
if (!STBLB(25, rec_count))
    printf("Problem with Starnode board. \n");
else
    for (i=0; i <rec_count; i++) {
            block_count = (short) fgetc(pclfile);
            fread(buf, block_count, 1, pclfile);
            if (!STBLN(block_count, buf)) {
                    printf("Problem with Starnode board. \n");
                    break;
            }
    }

fclose(pclfile);
```

Visual Basic

SYNTAX
STATUS% = STBLN(RECORD_LEN%, INBUF$)

Example

```
Dim STATUS%
Dim RESULTS%
Dim RECORD_COUNT%
Dim TYPE_NUM%
Dim INBUF$
Dim I%
Dim BYTE1 As Byte
Dim BYTE2 As Byte

TYPE_NUM% = 1
Open "gen_civ.lan" For Binary As #98
Get #98, , BYTE1
Get #98, , BYTE2
RECORD_COUNT% = BYTE1 + (BYTE2 * 256)

STATUS% = STBLB(TYPE_NUM%, RECORD_COUNT%)

If (STATUS% <> 1) Then _
   Print "ERROR: non-1 return from stblb: "; STATUS%:
GoTo STBLB1
For I% = 1 To RECORD_COUNT%
   Get #98, , BYTE1
   INBUF$ = String(BYTE1, " ")
   Get #98, , INBUF$
   STATUS% = STBLN(BYTE1, INBUF$)
   If (STATUS% <> 1) Then _
          Print "ERROR: non-1 return from stbln: ";
STATUS%: GoTo STBLB1
Next I%
Close #98

STBLB1:
RESULTS% = STATUS%
```

STIME

Send DOS Time to Starnode board

Upon completion, the function will have a value of 1 (TRUE) if it completed successfully, or 0 (FALSE) if there was a problem in communicating with the Starnode board.

C Language

SYNTAX
TIME();

Example

```
if (!STIME())
   printf("Problem with Starnode board. \n");
```

Visual Basic

SYNTAX
STATUS% = STIME()

Example

```
Dim STATUS%

STATUS% = STIME ()'send DOS time to board

If (STATUS% = False) Then
   MsgBox ("Problem with Starnode board.")
End If
```

SVER

Get Starnode ROM version and Download Table size

The value returned by this function is a 16-bit integer. The high order bits (8-15) contain the information regarding the size of the download table, while the low order bits (0-7) contain the Starnode PC Interface board ROM version number.

C Language

SYNTAX
SVER (&romver);

Example

```
if (!SVER(&romver))
   printf("Problem with Starnode board. \n");
else
   printf ("Download table size and version are : %4X",
romver);
```

Visual Basic

SYNTAX
STATUS% = SVER (ROMVER%)

Example

```
Dim STATUS%
Dim ROMVER%

STATUS% = SVER (ROMVER%) ' get the version of ROM

If (STATUS% = False) Then
    MsgBox ("Problem with Starnode board.")
    Else
    MsgBox ("Download table and size are: " + Format (ROMVER%))
    End If
```

CHAPTER 9

Controlling System's Data Collection Terminals

Messages may be sent from the host to Starnode devices, which control certain terminal functions. Each message starts with a command byte that tells the Starnode device what to do with the message. The remainder of the message may contain screen control characters.

Terminal Control Commands

The commands must be sent preceding every message and allow you to control the Starnode devices or request certain types of information. The command values below are passed to the SSEND, SBRDC, or SBRDT routines in the command parameter, *not* as part of the message. The single exception to this rule is the *Program Message* where the command value is the first character of the message.

A description of each command is show in Table 9–1. The commands are listed alphabetically, by function.

TABLE 9–1. Terminal Control Commands

Message Type	Decimal	Hex	ASCII
System Message	48	30	0
Prompt Message	49	31	1
Disable Time and Date	50	32	2
Enable Time and Date	51	33	3
Send Diagnostic Message	52	34	4
Send Status Information	53	35	5
User Program Load	57	39	9
User Program Boot	58	3A	:
Reset	59	3B	;
Clear Diagnostic Count	60	3C	<
Turn Relay On	61	3D	=
Turn Relay Off	62	3E	>
Program Message	65 (or greater)	41 (or greater)	A

Sending System Message

The command 48 (hexadecimal 30) allows you to display a system message on a data collection terminal's screen. When the system message is sent, any data currently appearing on the data collection terminals' displays will blank out. The system message will then display for a specified period of time. After the specified time period has expired, the original screen will be displayed.

Command Parameter Value: 48 Decimal (30 hex)
Message Format: <TIME><DATA>

where:

TIME—Is a single byte (the first byte of the message), its ASCII character value represents the amount of time you want the message to remain on the display. Allowed values are 0 to 127 seconds (*NUL* to *DEL*). However, if a value greater than 63 seconds (*?*) is set, the message string will be continually recycled. The recycle period is determined by the value of

duration. For example, if you set a duration of 100 seconds, the message will recycle every 36 seconds (100 - 64 = 36).

Note: Use of recycled messages will disable the data collection terminal from further use until a reset is performed. Refer to "Resetting Starnode Device" on page 9-7 for more information.

DATA—Is a string of up to 125 ASCII characters representing the message you want displayed. The screen control characters discussed in the next section may be incorporated into the data string.

Sending Prompt Message

The command 49 (hexadecimal 31) allows you to display a prompt message on the bar code data collection terminal's screen. When this command is used, the prompt message will be written to the terminal's display starting at the current cursor position. The message will remain on the display until another message must be shown. Prompt messages are used in a non-programmed, real-time prompt response protocol.

Command Parameter Value—49 (31 hex)
Message Format—<DATA>

where:

DATA—Is a string of up to 126 ASCII characters that are to be written to the terminal's display. This string may contain screen control commands.

Disabling Time & Date Display

The time and date are displayed in the lower right-hand corner of the data collection terminal's display, if there is enough room. This is described in detail in the operator's manual provided with your data collection terminal. You can disable the time and date display by using the 50 (hexadecimal 32) command. Even if the display is being already being controlled locally by a resident user program, the 50 command will override any previously set screen attributes.

Command Parameter Value—50 (32 hex)
Message Format—None

Enabling Time & Date Display

The time and date are displayed in the lower right-hand corner of the data collection terminal's display. This is described in detail in the operator's manual provided with your data collection terminal. You can enable the time and date display by using the 51 (hexadecimal 33) command. Even if the display is being already being controlled locally by a resident user program, the 51 command will override any previously set screen attributes.

Command Parameter Value—51 (33 hex)
Message Format—None

Requesting Diagnostic String

Upon the receipt of a 52 (34 hexadecimal) command, the Starnode device will send the following string back to the PC:

SSXXX #NNNNN

where:

SSXXX—Is the model number of the Starnode device.

NNNNN—Is a number of Diagnostic String messages sent. This will be a value between 00000 and 65535.

This string can be used for diagnostic purposes. The message count is incriminated each time this message is requested. The count will be reset to zero when the Starnode device receives a Clear Diagnostics Count (Command 60) or when the Starnode device is reset. This can be performed from the Starnode PC using the 59 (3B hexadecimal) command. Refer to the operator's manual accompanying the unit or "Resetting Starnode Device" on page 9-7 for more information.

Command Parameter Value—52 (34 hex)
Message Format—None

Requesting Status String

When a 53 (35 hexadecimal) is received by the Starnode device, the Starnode device will respond by sending a string representing its current operating parameter setup. This string is different for each Starnode device. Refer to the Starnode device manual for information on the status string. The following is a typical format:

MMM VV S TTT CCC LLLLL FF GG HHH III JJJ KKK B

where:

MMM—Is the three-digit Starnode device model.

VV—Is the version number of the operating system program residing in the unit. It will be important to know this information when making a service call.

S—Represents the status of the Starnode board's receive buffers. A returned value of 0 indicates that the unit's receive buffers are empty, while a value of 1 indicates that the unit's receive buffers are full.

TTT—Represents the unit's assigned terminal type. The terminal type will be a value between 000 and 127. Programs cannot be downloaded to a Starnode device assigned a value of 000. The terminal type value is assigned locally at the unit and can be temporarily changed within a user program.

CCC—Indicates if the clock display is enabled and, if so, at what level the display is *enabled*. The value returned is a bit-wise value, which may consist of the following component values:

001—Clock display *enabled* at the operating system level.

016—Clock display *enabled* at user program level.

064—Clock display temporarily *disabled*.

128—Unit is operating under user program control.

LLLLL—Indicates which types of bar codes are auto distinguished by the unit. The barcode scheme can be set either locally at the operating system level or temporarily overridden by a user program.

FF—Represents the number of characters that will be decoded in fixed length barcodes. Valid lengths are 0 to 50 characters.

GG—Specifies the minimum number of characters that a label must contain in order to be decoded. Valid minimum lengths are 00 to 50 characters.

HHH—Specifies if any check digits enabled. This value will be 000 if the option is *disabled*.

III—Indicates which UPC/EAN Title and Issue Extensions the unit decodes. This value will be 000 if the option is *disabled*.

JJJ—Indicates whether or not the Code 93 and/or Code 3-of-9 append functions will be decoded. This value will be 000, if the option is *disabled*.

KKK—Indicates the MSI check digit-decoding scheme. This value will be 000 if the option is *disabled*.

B—Represents the pitch and volume settings of the unit's beeper tone.

Command Parameter Value—53 (35 hex)
Message Format—None

Loading User Program

Message Format—None

To load a user program through the application program, use the 57 (hexadecimal 39) command. This command will allow the loading of a compiled user program.

Note: This command already appears as the first character on each download line of a user program's .LAN file.

Command Parameter Value—57 (39 hex)
Message Format—From .LAN File

Booting User Program

If you want to reboot a user program that has already been downloaded to the Starnode device, use the 58 (hexadecimal 3A) command.

If no user program has been downloaded to this terminal and this command is sent, an error will occur and the display will return to the power-up screen.

If a user program does reside at the Starnode device, program execution will be interrupted and the program restarted from the beginning.

Command Value—58 (3A hex)
Message Format—None

Resetting Starnode Device

Upon receipt of the 59 (hexadecimal 3B) command, the Starnode device will be reset to the operating parameters that were set up at the operating system level and power-up state. Refer to the unit's operator's manual for more specific information regarding the operating parameters.

Sending this command to a terminal is similar to turning its power *Off* and *On* again, except that the terminal will not go through start-up diagnostics.

Command Parameter Value: 59 (3B hex)
Message Format: None

Clearing Diagnostic Count

The value 60 (hexadecimal 3C) is used to clear the diagnostic count information from a terminal without having to restart the Starnode device. Refer to "Requesting Diagnostic String" on page 9-4 for more information.

This count may also be resent by sending a command value of 59, or by powering the terminal *Off* and *On*.

Command Parameter Value: 60 (3C hex)
Message Format: None

Enabling DC Control Output

Starnode devices may be equipped with DC control outputs, which may be used to control relays, conveyors, alarms, etc. These may be enabled from a PC application program using the 61 (hexadecimal 3D) command. Installation specifications for these DC control outputs are given in the operator's manual provided with each unit.

Command Parameter Value: 61 (3D hex)
Message Format: N

where:

N—Is a byte, single ASCII character, with an integer value between 1 and 256. The byte's value determines which DC control output is enabled. The values represent the following:

1—Turn on DC control output \#1

2—Turn on DC control output \#2

4—Turn on DC control output \#3

8—Turn on DC control output \#4

16—Turn on DC control output \#5

32—Turn on DC control output \#6

64—Turn on DC control output \#7

128—Turn on DC control output \#8

The above values may be summed to turn on multiple outputs simultaneously. This command only turns outputs *On*. If an output is already *On*, its state will not be affected by this command.

Disabling DC Control Output

Starnode devices may be equipped with DC control outputs, which may be used to control relays, conveyors, alarms, etc. These may be disabled from a PC application program using the 62 (hexadecimal 3E) command. Installation specifications for these DC control outputs are given in the operator's manual provided with each unit.

Command Parameter Value—62 (3E hex)
Message Format—N

where:

N—Is a byte (single ASCII character) with an integer value between 1 and 256. The byte's value determines which DC control output is disabled. The values represent the following:

1—Turn *off* DC control output \#1

2—Turn *off* DC control output \#2

4—Turn *off* DC control output \#3

8—Turn *off* DC control output \#4

16—Turn *off* DC control output \#5

32—Turn *off* DC control output \#6

64—Turn *off* DC control output \#7

128—Turn *off* DC control output \#8

These values may be summed to turn off multiple outputs simultaneously. This command only turns outputs *Off*. If an output is already *Off*, its state will not be affected by this command.

Forwarding String to Application Program

Any command value equal to or greater than 65 (hexadecimal 41) instructs the terminal to forward the message to the application program running in a Starnode device.

The message received by the program will have the command byte as the first byte of the message data, followed by up to 126 characters of the message. For example, a message of *TEST* sent to a Starnode device with a command byte of 65 would be received by a user program as a single message: ATEST.

Command Parameter Value—65 or greater (40 hex or greater)
Message Format—Any message (up to 126 characters) to be passed to the user application program in the Starnode device.

Screen Control Characters

The Screen Control Characters are used to manipulate a data collection terminal's display. To use the characters, embed them in the data that is to be sent to the display. This data may be shown by a user program or sent to the terminal from the PC using the SSEND, SRBDC, or SBRDT routines.

A short description of each character's is shown in Table 9–2.

TABLE 9–2. Screen Control Characters

Dec	Hex	Description
7	07	Bell (Single Beep)
8	08	Backspace
9	09	Horizontal Tab
10	0A	Line Feed (Down One Line)
13	0D	Carriage Return (Beginning of Current Line)

TABLE 9–2. Screen Control Characters (Continued)

Dec	Hex	Description
21	15	NAK (Clear Line)
22	16	Synchronous Idle (Error Beep)
24	18	Cancel (Clear Screen)
127	7F	Delete (Erase Character)

The following describe how a Starnode Terminal interprets Screen Control Characters.

- *Bell*—This character causes a Starnode terminal to emit a single, short beep.
- *Backspace*—This character causes the cursor to move backward, to the left, one space on the display, without deleting data. If the cursor is already in the home position when the *Backspace* character is received, the cursor will not move.
- *Horizontal Tab*—This character causes the cursor to move forward (to the right) one character position on the display. If a character is already being shown in that position when the *Horizontal Tab* character is received, the character is overwritten by a *Space* character.
- *Line Feed*—This character causes the cursor to move down one line on the display. For example, if the cursor is on the top line of the display, in the fourth character position, the *Line Feed* character results in the cursor moving to the second line of the display, still in the fourth character position.
- *Carriage Return*—Causes the cursor to move to the beginning, left-most character position, of the display. If the cursor is on the first line of the Starnode terminal's display when the *Carriage Return* character is received, the cursor remains on the first line but returns to the first, left-most, character position.
- *Synchronous Idle*—This character causes the Starnode terminal to emit an error tone, four 4 short beeps.
- *NAK*—This character causes the line that the cursor is currently on, as well as the following line, if any, to be cleared of all characters. The cursor is moved to the beginning of the current line.

- *Cancel*—This character causes the Starnode terminal to clear its display completely of all characters currently being shown. The cursor returns to the home position.
- *Delete*—This character causes the cursor to move backwards, to the left, one space. Any character that was being shown in the cursor's original position is cleared.

10

CHAPTER 10

Maintenance & Troubleshooting

This chapter provides general maintenance procedures for the Starnode PC data collection network.

Network Troubleshooting

Maintenance of the Starnode Data Collection consists of checking network cabling and device connections, ensuring the network is setup to specifications, and troubleshooting some common, minor network problems. A list of common network problems and their possible causes and solutions is provided in Table 10–1.

TABLE 10–1. Common Network Problems

Description of Problem	Possible Cause	Possible Solution
Program running at Starnode device locks up at end of transaction. Any Starnode device on the Starnode network exhibits the same lock-up after one or two transactions have been entered.	The Starnode board's received message queue is full and the PC is no longer picking up messages from the Starnode board. There may be a bug in the application program or the program was halted.	De-bug or re-start your application program.
Time display on the Starnode terminal is frozen or a period (.) is shown to the right of the seconds position.	Network was stopped from PC by call to SSTOP routine.	Re-start your application program on the PC. Messages from Starnode devices may have been buffered in the terminals and transmission may be attempted again once the controller begins polling the network again.
	Hardware failure or a cable break on the network.	Refer to Problem—Waiting for Starnode.

TABLE 10–1. Common Network Problems (Continued)

Description of Problem	Possible Cause	Possible Solution
Prompt Waiting for Starnode remains on display.	Network cable may be disconnected from the Starnode device.	Reconnect the interface cable to the Starnode device.
	Network cable may have pulled loose from T-tap box.	Re-install the cable in the T-tap box. Check all connections using a continuity tester.
	Network cable is disconnected from Starnode board.	Re-connect network cable to Starnode board.
	If the Starnode devices not driven by the Repeater continue to display the correct time, then the Repeater between the PC and the down device has failed.	Call SICK Auto Ident, Inc.'s Return Authorization department for an R.A. number. Remove the circuit board from the Repeater and return it for repair. Replace the faulty board with a working board.
	A transceiver chip in a Starnode device or Repeater is shorting the network.	Disconnect and reconnect each terminal on the main branch circuit and each branch circuit off of each Repeater, one at a time, until the network resumes operating when the faulty unit or branch circuit is isolated.
Downloads occasionally fail and errors occur when sending data to a specific terminal ID.	Overlapping terminal IDs on gateway terminals.	Run STARNDEM diagnostics. These diagnostics will report all overlapping ID ranges. Change all affected terminal IDs to eliminate overlaps.

TABLE 10–1. Common Network Problems (Continued)

Description of Problem	Possible Cause	Possible Solution
When download is attempted, nothing happens.	Starnode board did not receive a download table for that terminal's type.	Verify that you did indeed download the program table to the Starnode board. Try using STARNDEM to download the program. If this works, re-check your software's logic. Ensure that the terminal type corresponds to what is expected by the software on the PC.
	Starnode board did not have room for this download table.	Reduce the size of your programs or download the programs to terminals individually (bypass the Starnode board entirely - call SICK Auto Ident, Inc. Application Support Department for instructions on how to do this).
PC is not receiving messages from terminal.	This is symptomatic of a PC that is too fast for the Starnode board.	Increase the delay time after each communication with the Starnode board by calling the SDLAY routine before your first call to SINIT. Try delay values of between 2 and 50.
PC periodically receives messages from terminal ID that is not on the network. STARNDEM diagnostics reports non-zero statuses for several IDs that do not exist.	The above problems are symptomatic of a PC that is too fast for the Starnode board.	Increase the delay time after each communication with the Starnode board by calling the SDLAY routine before your first call to SINIT. Try delay values of between 2 and 50.
	Discontinuous cable shielding.	Check shield continuity through all T-tap boxes, or have a network verification done by Field Service.
PC fails to boot properly or PC has problem locating hard disk.	Starnode board I/O port address conflicts with other I/O port addresses in machine.	Change JH4 jumpers on Starnode board to another I/O port address. If PC boots properly, run STARNDEM and test the new address.

TABLE 10–1. Common Network Problems (Continued)

Description of Problem	Possible Cause	Possible Solution
Terminals receive date and time only intermittently. Terminals fail to recognize that Starnode is up, or messages take a long time to be picked up.	Discontinuous cable shielding.	Check shield continuity through all T-tap boxes, or have a network verification done by Field Service.
	A *Y* in the branch circuit.	Re-wire the branch circuit to be one long wire with a series of short drops. Install a U in the branch circuit to pick up the terminals that were previously on one branch of the Y.
Network operated until another Starnode device was added, or network operates normally except for devices connected to one branch circuit off the same Repeater.	Multiple terminating resistors.	Verify that only the last T-tap box on a branch circuit has a terminating resistor installed in it. Also verify that the T-tap closest to the Starnode board does not have a terminating resistor.
Network operates properly until Repeater is connected.	The interface cables are too short near the PC or Repeater.	Make sure there is at least 50 feet of branch circuit cable between the PC/Repeater and the Starnode device drop cable. If you have less than 50 feet, replace short cable with 50 feet of Starnode cable.
Starnode device does not receive a download table. Messages received are from a terminal type of 0.	Terminal type has not been setup - it has a type of 0.	Set Terminal Type of Starnode device.
Data transmissions between the terminal and PC are filled with error characters (default character is @) or incorrect data is received by receiving devices.	The baud rate or parity is not set correctly.	Check the baud rate and parity selected on both devices and reset them if necessary.

TABLE 10–1. Common Network Problems (Continued)

Description of Problem	Possible Cause	Possible Solution
Starnode device sends transactions to PC, occasionally with a type of 0, even though all terminals on the network have been assigned a non-zero type.	The particular Starnode device had a message sitting in the Starnode buffer when the Starnode network was initialized. Messages from terminals do not include the terminal type; this information is supplied by the Starnode board before the message is passed to the PC. In this instance, the Starnode board received the waiting message before it received the startup message containing the terminal's type.	Program your Starnode device not to send a message without checking the LAN status. If the LAN is not operating, do not do a PUT LAN.
When connecting a Starnode device serial port to another device using RS-232C communications, the other device hangs, fails to operate, or behaves erratically.	Ground loop problem exists.	Use the minimum number of conductors possible in the RS-232C cable. Usually pins 2, 3, and 7 are adequate for communications.
	Too many conductors are in the cable linking the two devices.	Use the minimum number of conductors possible in the RS-232C cable. You may need to use only pins 1, 2, 3, 4, 5, and 7 (if you are using RTS/CTS), pins 1, 2, 3, and 7, or pins 2, 3, and 7.
When connecting a terminal to the Starnode network through a gateway terminal, the PC receives messages one character at a time.	The gateway terminal does not have its port set up in Block Mode.	Change the port parameters by entering the appropriate setup string. This may be done in a user program, or local setup of device.
Gateway terminal does not seem to receive anything when Block Mode is enabled.	The expected prefix character is not being received (if enabled), or suffix character is not being received.	Set up the prefix and suffix characters again for the serial port, double-checking that the device is sending them.

TABLE 10–1. Common Network Problems (Continued)

Description of Problem	Possible Cause	Possible Solution
Serial port errors occur randomly on the Starnode device.	The error character is being received as part of the data sent from the device.	At the operating system level, or in your user setup string for the serial port (whichever is in effect at the time), change or disable the error character.
Starnode network does not seem to work.	Any of the causes listed in this section.	Run the STARNDEM program verifying that it does not report any errors when initializing the Starnode board. Turn on several data collection terminals and scan the LA label. Run diagnostics under STARNDEM. If diagnostics report several terminals on the network, and run to completion without reporting errors, the network works. The problem may be in your application program.
When keyboard on Starnode device is pressed, random characters are generated or all keys generate the same character.	A bad key or keyboard encoder chip.	Call SICK Auto Ident, Inc. to have the terminal repaired.
Terminal randomly resets to power-on state, with no error messages shown.	Intermittent power or improper grounding.	Check the reliability of your power supply by using a power line monitor to record spikes and brownouts. If power problems exist, install power line filters or some sort of filtering uninterruptible power supply system. Verify that all Starnode-related electrical outlets are properly grounded to a common earth ground.

APPENDIX A *ASCII Equivalence*

This appendix contains the ASCII character set with decimal and hex equivalents for each character, and the keyboard strokes required to generate ASCII control characters.

ASCII Equivalence Table

TABLE A–1. ASCII Equivalence Table

ASCII Character	Decimal	Hex	Keyboard
NULL	000	00	CTRL @
SOH (START OF HEADING)	001	01	CTRL A
STX (START OF TEXT)	002	02	CTRL B
ETX (END OF TEXT	003	03	CTRL C
EOT (END OF TRANSMISSION)	004	04	CTRL D
ENQ (ENQUIRY)	005	05	CTRL E
ACK (ACKNOWLEDGE)	006	06	CTRL F
BEL (BELL)	007	07	CTRL G
BS (BACKSPACE)	008	08	CTRL H
HT (HORIZONTAL TAB)	009	09	CTRL I
LF (LINE FEED)	010	0A	CTRL J
VT (VERTICAL TAB)	011	0B	CTRL K
FF (FORM FEED)	012	0C	CTRL L

TABLE A–1. ASCII Equivalence Table (Continued)

ASCII Character	Decimal	Hex	Keyboard
CR (CARRIAGE RETURN)	013	0D	CTRL M
SO (SHIFT OUT)	014	0E	CTRL N
SI (SHIFT IN)	015	0F	CTRL O
DLE (DATA LINK ESC)	016	10	CTRL P
DC1 (DATA CONTROL 1)	017	11	CTRL Q
DC2 (DATA CONTROL 2)	018	12	CTRL R
DC3 (DATA CONTROL 3)	019	13	CTRL S
DC4 (DATA CONTROL 4)	020	14	CTRL T
NAK (N0 ACKNOWLEDGE)	021	15	CTRL U
SYN (SYNCHRONOUS IDLE)	022	16	CTRL V
ETB (END OF TRANS. BLOCK)	023	17	CTRL W
CAN (CANCEL)	024	18	CTRL X
EM (END OF MEDIUM)	025	19	CTRL Y
SUB (SUBSTITUTE)	026	1A	CTRL Z
ESC (ESCAPE)	027	1B	CTRL [
FS (FILE SEPARATOR)	028	1C	CTRL \
GS (GROUP SEPARATOR)	029	1D	CTRL]
RS (RECORD SEPARATOR)	030	1E	CTRL ^
US (UNIT SEPARATOR)	031	1F	CTRL _
SP (SPACE)	032	20	
! (EXCLAMATION POINT)	033	21	
" (QUOTE MARK)	034	22	
# (POUND SIGN)	035	23	
$ (DOLLAR SIGN)	036	24	
% (PERCENT SIGN)	037	25	
& (AMPERSAND)	038	26	
' (APOSTROPHE)	039	27	
((LEFT PAREN)	040	28	
) (RIGHT PAREN)	041	29	
* (ASTERISK)	042	2A	
+ (PLUS SIGN)	043	2B	

TABLE A–1. ASCII Equivalence Table (Continued)

ASCII Character	Decimal	Hex	Keyboard
,(COMMA)	044	2C	
- (MINUS SIGN)	045	2D	
. (PERIOD)	046	2E	
/ (SLASH)	047	2F	
0	048	30	
1	049	31	
2	050	32	
3	051	33	
4	052	34	
5	053	35	
6	054	36	
7	055	37	
8	056	38	
9	057	39	
: (COLON)	058	3A	
; (SEMI-COLON)	059	3B	
<	060	3C	
=	061	3D	
>	062	3E	
?	063	3F	
@	064	40	
A	065	41	
B	066	42	
C	067	43	
D	068	44	
E	069	45	
F	070	46	
G	071	47	
H	072	48	
I	073	49	
J	074	4A	

TABLE A–1. ASCII Equivalence Table (Continued)

ASCII Character	Decimal	Hex	Keyboard
K	075	4B	
L	076	4C	
M	077	4D	
N	078	4E	
O	079	4F	
P	080	50	
Q	081	51	
R	082	52	
S	083	53	
T	084	54	
U	085	55	
V	086	56	
W	087	57	
X	088	58	
Y	089	59	
Z	090	5A	
[(LEFT BRACKET)	091	5B	
\ (BACKSLASH)	092	5C	
] (RIGHT BRACKET)	093	5D	
^ (UP ARROW)	094	5E	
_ (UNDERLINE)	095	5F	
' (ACCENT,GRAVE)	096	60	
a	097	61	
b	098	62	
c	099	63	
d	100	64	
e	101	65	
f	102	66	
g	104	67	
h	104	68	
i	105	69	

TABLE A–1. ASCII Equivalence Table (Continued)

ASCII Character	Decimal	Hex	Keyboard
j	106	6A	
k	107	6B	
l	108	6C	
m	109	6D	
n	110	6E	
o	111	6F	
p	112	70	
q	113	71	
r	114	72	
s	115	73	
t	116	74	
u	117	75	
v	118	76	
w	119	77	
x	120	78	
y	121	79	
z	122	7A	
{	123	7B	
| (VERTICAL BAR)	124	7C	
}	125	7D	
~ (TILDE)	126	7E	
DEL (DELETE)	127	7F	

Index