



# SYSTEMS

*Engineering Associates, Inc.*

## **S3016-505 (MCOM505 AND NETCOM) User's Manual**

Systems Engineering Associates, Inc.  
14989 West 69th Avenue  
Arvada, Colorado 80007 U.S.A.  
Telephone: (303) 421-0484  
Fax: (303) 421-8108  
[www.sea-seg.com](http://www.sea-seg.com)

12/2007

---

**SYSTEMS Electronics Group**

14989 West 69<sup>th</sup> Avenue, Arvada, CO 80007 U.S.A. Phone: (303) 421-0484 FAX: (303) 421-8108

**S3016-505  
(MCOM505 AND NETCOM)  
User's Manual**

Copyright © 2004 - 2007 Systems Engineering Associates, Inc.

All Rights Reserved!

Revised: 11 December 2007

## **WARNING**

To ensure the equipment described by this User Manual, as well as the equipment connected to and used with it, operates satisfactorily and safely, all applicable local and national codes that apply to installing and operating the equipment must be followed. This includes the National Electric Code in the USA and other applicable legislation, regulations, and codes in practice elsewhere. Since codes can vary geographically and can change with time, it is the user's responsibility to determine which standards and codes apply, and to comply with them.

**FAILURE TO COMPLY WITH APPLICABLE CODES AND STANDARDS CAN RESULT IN DAMAGE TO EQUIPMENT AND/OR SERIOUS INJURY TO PERSONNEL.**

Persons supervising and performing installation or maintenance must be suitably qualified and competent in these duties, and should carefully study this User Manual and any other manuals referred to by it prior to installation and/or operation of the equipment.

---

The contents of the User Manual are believed to be correct at the time of printing; however, no responsibility is assumed for inaccuracies. In the interests of a commitment to a policy of continuous development and improvement, the manufacturer reserves the right to change the specification of the product or its performance or the contents of the User Manual without notice.

---

Copyright © 2004 - 2007 Systems Engineering Associates, Inc.

All Rights Reserved!



# CONTENTS

<b>1. General Description</b>	<b>1</b>
1.1 Features	1
1.2 S3000 Serial Network Operation	2
1.3 Communications Between TI505 and S3016-505	3
1.4 The MCOM505 Application Program	4
1.5 The Netcom Setup Program	5
<b>2. Installation</b>	<b>7</b>
2.1 Installing the S3016-505 in the Rack	7
2.2 S3000 Serial Network Installation	8
2.2.1 Wiring the Serial Network	8
2.2.2 Setting the Network Address in the Slave Nodes	10
2.3 I/O Configuration of TI Processors for S3016-505	11
2.4 Downloading "MCOM505" to the S3016-505	12
2.5 Installing "MCOM505" on the Hard Drive	12
<b>3. NETCOMM Windows Based Setup Program Reference</b>	<b>13</b>
3.1 General Description	14
3.2 The File Menu	17
3.2.1 The Set-up Data File	18
3.2.2 Download Program	20
3.2.3 Print Report	21
3.3 The Edit Menu	22
3.3.1 Enable Offline Editing	23
3.3.2 Setup Comm Port	23
3.3.3 TI Status Stack Address	24
3.4 The View Menu	26
3.4.1 View Online Data	27
3.4.2 View Offline Data	27
3.4.3 Target Board Interface	28
3.4.4 Communication Interface	29
3.5 The Main Display	30
3.5.1 The Comm Status Tab	31
3.5.2 The Configuration Tab	33
3.5.3 The Descriptions Tab	35

# CONTENTS

<b>4. NETCOM DOS Based Setup Program Reference</b>	<b>37</b>
4.1 Installing and Running NETCOM	37
4.2 "NETCOM" Shell	39
4.2.1 "NETCOM" Shell Menu	40
4.2.2 "NETCOM" Shell Commands	41
4.3 Main Menu	44
4.4 Edit Node Communications Parameters	45
4.4.1 Select Node Number	47
4.4.2 Edit Node Description	47
4.4.3 Edit Node Communications Parameters	47
4.4.4 Enable/Disable Communications to Node	48
4.5 Download Parameters to S3016-505	49
4.6 Upload Parameters from S3016-505	50
4.7 Print Node Communications Parameters	52
4.8 Set TI-505 Communications Status Stack Address	53
4.9 Monitor Communications Status (PROG Port)	54
<b>5. Specifications</b>	<b>57</b>

## LIST OF FIGURES

Figure 2.1 – Typical Network Wiring	9
Figure 2.2 – Alternative Serial Connector Wiring	9

## APPENDICES

MCOM505 Program Printout	Appendix A
RS-232/RS-422 Pin-outs/Cables	Appendix B

# SECTION 1

## GENERAL DESCRIPTION

The S3016-505 is a version of the S3016 CO-CPU communications board that resides on the Texas Instruments 505 bus structure. With the exception of that difference, the S3016-505 provides all the same features as the standard S3016. The S3016-505 is a communications CO-CPU board, which provides one S3000 serial network interface port and one RS-232/RS-422 USER PORT. The S3016-505 is a true CO-CPU with it's own processor and program/data memory which executes a user application program independent of the TI 505 main processor. The primary use of the S3016-505 is to allow communications between the TI 505 family of processors and the S3000/M4000 line of processors via the S3000 serial network. The S3016-505 can be installed in any I/O slot of the TI 505 rack. In addition, any number of S3016-505s may be installed in one TI 505 rack (up to the number of I/O slots available). This manual describes the features unique to the S3016-505, refer to the S3016 User's Manual for details on the S3016 in general.

---

### 1.1 FEATURES

- Communications CO-CPU Board with S3000 Serial Network Interface and RS-232/RS-422 USER PORT.
- Built in Processor executes User application program independently of main TI 505 processor.
- Resides on TI 505 Bus structure as a 3 word in / 5 word out Special Function module (similar to a TI Peerlink module).
- Provided with "MCOM505" S3016-505 program and "NETCOM" (DOS and Windows based) setup program which allows the TI 505 processor to communicate with up to 31 S3000/M4000/S3016-505 slave nodes with up to 120 words to and from each slave node.
- 24K Bytes User Program memory.
- 2K Bytes Data memory.
- Built in Real Time clock provides current time and date.

## SECTION 1

### GENERAL DESCRIPTION

- Fast program execution time (0.6 Milliseconds per 1K Bytes Program memory).
- Programmed with “SYSdev” program development software package. This allows the S3016-505 to be programmed in a combination of Ladder, High-level ('C') and Assembly (MCS-51).
- Interfaces with PC or Laptop computer via RS-232 for program download and on-line monitoring.
- Extensive internal diagnostics/fault detection including watchdog timer, communications fault detection, hardware confidence test, etc.
- Status LEDS on faceplate (RUN, SERIAL NETWORK COMM, and FAULT)
- Standard single width TI 505 module size.

---

#### 1.2 S3000 SERIAL NETWORK OPERATION

The S3000 serial network provides a means for the S3016-505 to communicate with other S3000/M4000/S3016-505 processors (nodes). The network operates in a master/slave topology. The S3016-505 is the master and controls all communications on the network. The remaining S3000/M4000/S3016-505 nodes act as slaves and simply respond to communications requests from the master. The master can send up to 120 consecutive words and receive up to 120 consecutive words from the slave in one command.

Up to 31 slave nodes can be connected to one S3016-505. Each node on the network is assigned a unique address from 1 to 32. The S3016-505 master node is automatically set to 1 when the “MCOM505” program is used, while the slave nodes are assigned addresses 2 through 32. The network address is used to specify which slave the master is communicating to. The network addresses in the S3000/M4000 nodes are set from the SYSdev Target Board Interface Menu and downloaded directly to the node from the PC or Laptop computer running SYSdev.



## SECTION 1

### GENERAL DESCRIPTION

Communications from the master is implemented using System function 13 (sfunc13). See the S3016 User's Manual for details on the system function 13 or the "MCOM505" program printout in appendix A for an example of using sfunc13.

---

#### 1.3 COMMUNICATIONS BETWEEN TI505 AND S3016-505

Communications between the TI 505 processors (545, etc.) is accomplished over the TI 505 back plane. The S3016-505 reads and writes to the V memory of the TI processor using the TI task codes. From the TI processor point of view, the V memory is read and written to transparently by the S3016-505. No special communications programming is implemented in the TI processor. The S3016-505 determines which V memory locations will be read and written to.

**Note:** Only the V memory is read and written to by the S3016-505. No other TI memory type reads and writes are supported by the S3016-505.

On the S3016-505 side, the algorithm for implementing the task code communication to the TI processor is embedded in the S3016-505 firmware. The S3016-505 user program initiates the communication by specifying which addresses will be read or written and how many words are to be transferred. Once initiated, the task code is executed transparently to the S3016-505 user program. See the "MCOM505" program printout in appendix A for an example of the task code communication to the TI processor.

## SECTION 1

### GENERAL DESCRIPTION

---

#### 1.4 THE MCOM505 APPLICATION PROGRAM

The primary purpose of the S3016-505 is to allow TI 505 based processors to communicate with other S3000/M4000/S3016-505 processors. The SYSdev “MCOM505” application program is loaded into the S3016-505 and allows a TI 505 processor to communicate with up to 31 slave nodes, reading and writing up to 120 words to each node. This program implements the task code communication to the TI 505 processor as well as the serial network communication to the slave nodes. No additional programming is required by the user.

The communication between the TI-505 and the S3016-505 is performed asynchronously, as is the communication between the S3016-505 to the slave nodes over the S3000 serial network.

The operation of the “MCOM505” program is as follows:

- 1) The “MCOM505” program reads the data from the TI-505 processor and stores this in an internal buffer (of the S3016-505). This data is then transmitted to a slave node.
- 2) Data is then read from a slave node (stored in an internal buffer of the S3016-505) and written to the TI-505 processor. This is performed in a sequential fashion for all sequences (0 to 30 or nodes 2 through 32) that are enabled for communications. The data that is passed for each node is defined in section 4.4.

**Note:** Task code communication will not occur for any disabled nodes (node number set to 0).

- 3) Once this data is updated (for all enabled nodes), the communications status is then written to the TI-505 processor (see section 4.9 for details on the status stack address data).
- 4) This process is continuously repeated, starting with the first enabled node.

**Note:** Communication between the TI-505 processor and the S3016-505, is occurring concurrently thus optimizing the total through-put of the system.

See section 2.4 for details on downloading the “MCOM505” program to the S3016-505.

---

### 1.5 THE NETCOM SETUP PROGRAM

The “NETCOM” setup program is a menu driven program (Windows or DOS based) and runs on a PC or Laptop computer. It is used to configure and monitor the communication to between the TI 505 processor and other S3000/N400/S3016-505 slave nodes. The combination of the “MCOM505” application program and the “NETCOM” setup program turns the S3016-505 into a communication board that allows the TI 505 processor to communicate to up 31 slave nodes.

“NETCOM” configures the V memory addresses and the number of “Words” used for reading data “From” and writing data “To” a slave node (W memory addresses). Once this information is downloaded to the S3016-505, the communication between the TI processor and the slave nodes is then performed automatically.

See section 3 for details on using the Windows based setup program.  
See section 4 for details on using the DOS based setup program.

## **SECTION 1**

### **GENERAL DESCRIPTION**

*(This Page Intentionally Left Blank)*

## SECTION 2 INSTALLATION

**Caution:** The internal components of the S3016-505 are susceptible to damage by static discharge. When handling the S3016-505, the board should be handled by the faceplate only and preferably in a static shielding bag.

---

### 2.1 INSTALLING THE S3016-505 IN THE RACK

The S3016-505 can be installed in any I/O slot of the TI 505 rack.

#### **Install the S3016-505 as follows:**

- 1) Turn power to the TI 505 rack “off”.
- 2) Install the S3016-505 in the rack by aligning the board with the card guides and sliding in until firmly seated. The board is held in the rack via captive screws located on the faceplate.
- 3) Connect the S3016-505 to the S3000 network by plugging the network field wiring connector into the network comm port, observing the proper keying of the connector.
- 4) Turn power to the TI 505 rack “on”.
- 5) Download “MCOM505” to the S3016-505 (see section 2.4).

#### **To remove the S3016-505 from the rack, perform the following:**

- 1) Turn power to the TI 505 rack “off”.
- 2) Remove the network field wiring connector from the “Comm” port.
- 3) Loosen the captive screws located on the faceplate and gently pull the board out of the rack using the handles located on the faceplate.

## SECTION 2 INSTALLATION

---

### 2.2 S3000 SERIAL NETWORK INSTALLATION

The S3000 serial network installation consists of wiring the network and setting each S3000/M4000/S3016-505 node on the network with a unique network address. Up to 31 slave nodes can be connected to the master S3016-505 on one network.

---

#### 2.2.1 WIRING THE SERIAL NETWORK

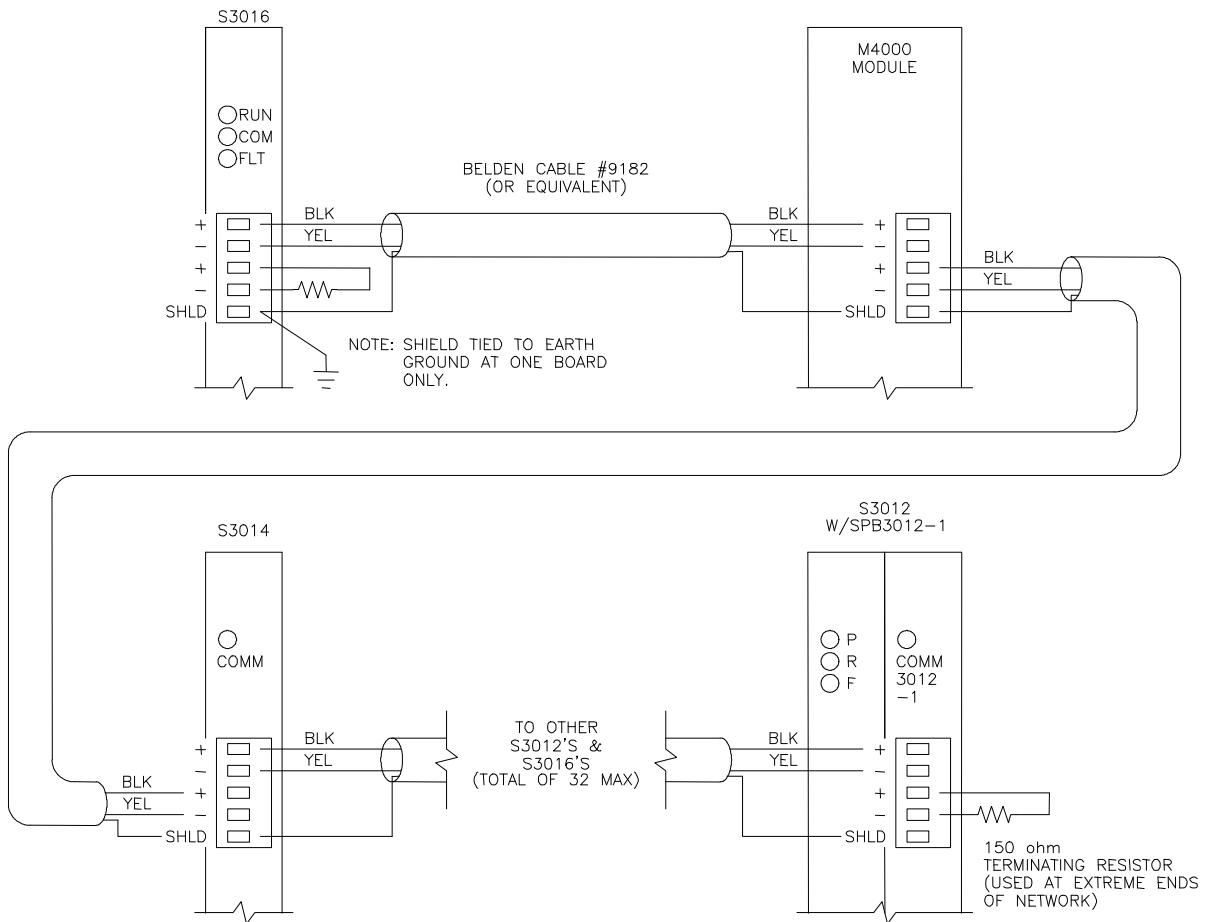
Refer to figure 2.1 for a typical schematic of the network and for the pin outs of the network interface connectors. When wiring the network, the following rules must be followed:

- 1) Wire the network using Belden #9182 single-shielded twisted pair cable or an equivalent data communications cable meeting the following spec:

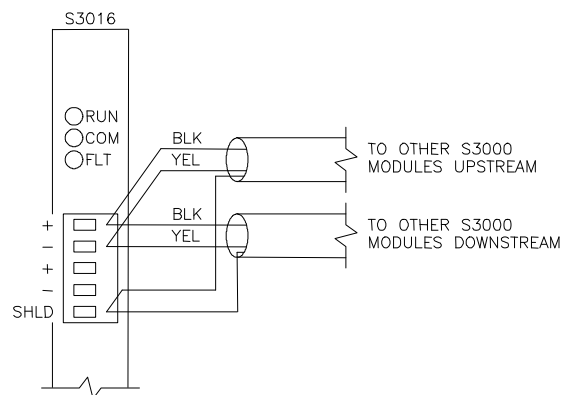
Wire gauge: \_\_\_\_\_ 22AWG  
Nom. impedance: \_\_\_\_\_ 150 ohms/ft.  
Nom. attenuation at 1 MHZ: \_\_\_\_ .004 db/ft.  
Twisted pair, single-shielded

- 2) The total wire length of the network cannot exceed 1,000 ft. if 344KBPS is selected, 2,000 ft. at 229KBPS, and 4,000 ft. at 106KBPS.
- 3) The shield of the cable should be carried through the entire network, using the shield tie points on the interface connectors to achieve this. The shield tie-points are strictly tie-points on the connectors are not internally tied to anything. One of these tie-points should then be tied to earth ground.
- 4) The two extreme ends of the network should be terminated with a 150 ohm resistors as shown in figure 2.1.
- 5) The network wiring should be isolated from other high voltage wiring by routing the network in a separate conduit dedicated to the network.

## SECTION 2 INSTALLATION



**Figure 2.1 – Typical Network Wiring**



**Figure 2.2 – Alternative Serial Connector Wiring**

## SECTION 2 INSTALLATION

- 6) The network should be wired directly to the network “Comm” port connectors. No intermediate terminations or splices should be used. The network should be wired in direct connect topology as shown, not in multi-drop or cluster topologies.

**Note:** The network “Comm” interface connectors contain two sets of + and - terminals. The two sets of terminals are tied together internally on the board (+ to +, - to -) and are provided as tie-points to ease wiring. Communications across the network will continue even if one of the nodes has failed provided all the connectors are installed in their respective board. However, if a connector is pulled from it's board, communications to the boards downstream will be lost (the internal tie-point will be broken). If it is desired, this situation can be avoided by wiring the connectors as shown in figure 2.2.

---

### 2.2.2 SETTING THE NETWORK ADDRESS IN THE SLAVE NODES

When using the “MCOM505” application program, the network address of the S3016-505 is automatically set to address 1. For this reason the S3016-505 network address does not have to be set by the user. Each of the slave S3000/M4000/S3016-505 nodes on the network must, however, be set by the user with a unique network address between 2 and 32. This is how the S3016-505 can distinguish one node from another.

To set the network address of a particular S3000/M4000 node, perform the following:

- 1) Connect the PC or Laptop computer running the Windows based NETCOM setup program, from the “COM” port on the PC to the “PROG PORT” on the respective S3000/M4000 node using a SYSdev RS-232 cable.
- 2) From the “View” menu, select Target Board Interface. A Windows Explorer type dialog box will appear, prompting the user to select a SYSdev program file.
- 3) Click the “Cancel” button. The “Target Board Interface” window will appear and the setup program will attempt to establish communications with the S3000/M4000 node.



## SECTION 2 INSTALLATION

- 4) Select the “Set Network Address” tab. The current network address will be displayed
- 5) To change the address, enter the new address in the field or use the increment/decrement control to change the address. The network address will be immediately changed.
- 6) Select “OK” to exit the Target Board Interface. The Setup program will remain online with the Target Board. Press the F4 key to cease online communications.

Repeat the above steps for all S3000/M4000 nodes on the network. This is true when the network is first installed, and when a new S3000/M4000 node is added or replaced on the network (that node must have the network address set in it).

**Note:** System function sfunc19(); is used to set the network node address and configure the S4516 for the M4500 processor. See the M4500 Program Development manual for more information on setting the Network Station Address using system function sfunc19().

---

### 2.3 I/O CONFIGURATION OF TI PROCESSORS FOR S3016-505

The S3016-505 can be installed in any I/O slot of the TI 505 rack. The S3016-505 is configured as a **3 word in/5 word out Special Function module**. In the TI I/O module definition table the S3016-505 should be defined as shown below:

<u>Slot</u>	<u>I/O Address</u>	<u>Number of BIT and WORD I/O</u>				<u>Special Function</u>
		<u>X</u>	<u>Y</u>	<u>WX</u>	<u>WY</u>	
XX	XXXX	00	00	03	05	YES

Where XX and XXXX under “SLOT” and “I/O ADDRESS” are the user defined slots and I/O addresses where the S3016-505 is located.

## SECTION 2 INSTALLATION

---

### 2.4 DOWNLOADING “MCOM505” TO THE S3016-505

With the S3016-505 powered up, perform the following to download the “MCOM505” program to the S3016-505:

- 1) Connect the PC or Laptop computer, running the Windows based NETCOM setup program, from the “COM” port on the PC to the “PROG PORT” on the S3016-505 using a SYSdev RS-232 cable.
- 2) Invoke the Windows based NETCOM setup program.
- 3) From the “File” menu, select “Download Program”. Select the directory that contains the “MCOM505” application program.

**Note:** If “MCOM505” has not yet been loaded onto the hard drive of the computer, perform the steps in section 2.5 to load “MCOM505” onto the computer.

- 4) Select the “MCOM505” program.
- 5) Follow the instructions on the screen to start the download process. The current address being downloaded will be displayed while program download is in progress.
- 6) Once the download is complete, the setup program will be left in an “Online” edit mode.

See section 3 for complete details on using the Windows based NETCOMM setup program.

---

### 2.5 INSTALLING “MCOM505” ON THE HARD DRIVE

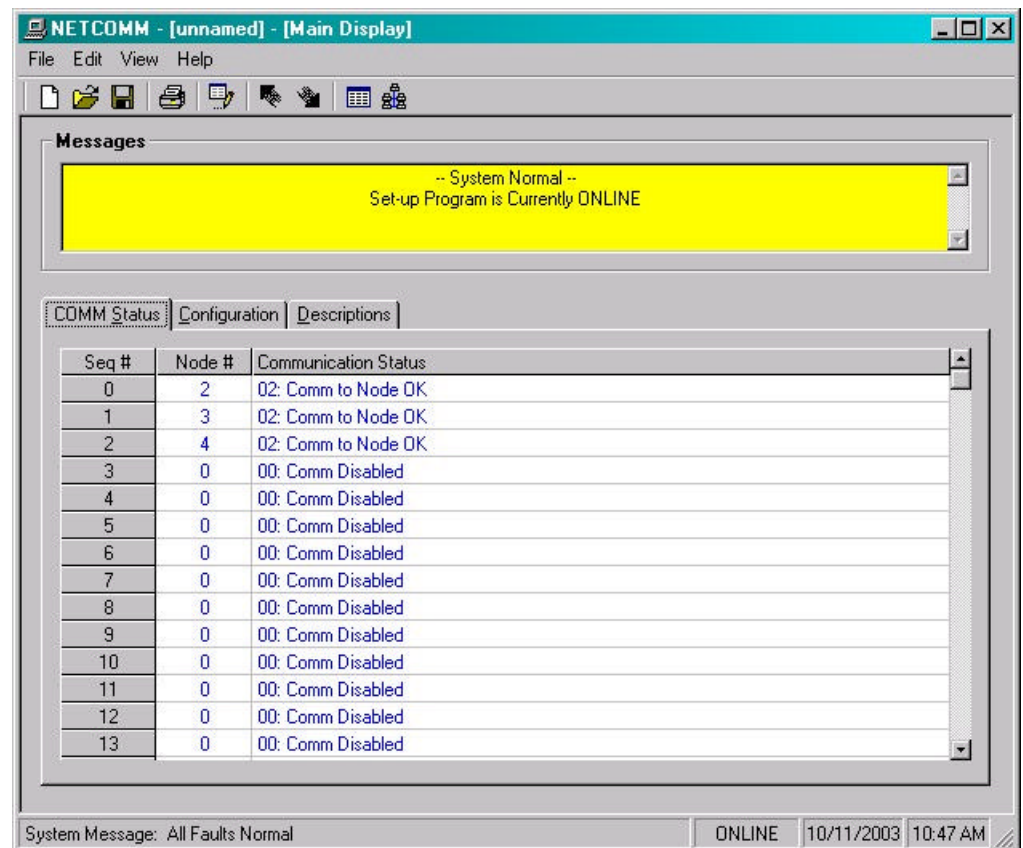
- 1) Create a directory off the root that will be used to store the “MCOM505” application program
- 2) Install the diskette that contains “MCOM505” into the floppy drive.
- 3) Copy all files from the program diskette to the directory created to store the “MCOM505” application program.

## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

The Windows based set-up program is menu driven, allowing the user to easily view communication status or alter communication parameters using a PC or Laptop computer running the Windows (95/98/ME/2000/XP/NT) operating system.

**Note:** The set-up program is an on-line communications program used to interface with the S3016-505. The data displayed and set in the windows is communicated directly to the module, while in the “Online” edit mode. Therefore, prior to going online with the processor, make sure an RS-232 cable is connected from the COM port on the computer to the "PROG" port on the S3016-505. The variables displayed while in the “Online” edit mode are read directly from the processor. Data is saved to a “Set-up Data” file (\*.sdt) whenever changes are made to a parameter or if the data is uploaded from the processor.



## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

---

#### 3.1 GENERAL DESCRIPTION

**Title Bar:** At the top of the window is the “Title Bar”. The title bar is used to display the name of the working “Set-up Data” file, as well as, the name of the active “Window”. The title bar is dark if the window is active and grayed if another window is active. The color depends on the settings of the Display Properties of the Control Panel.

**Tool Bar:** At the top of the window is a “Tool Bar”. The tool bar allows the user to quickly execute those functions most often used. These functions can also be executed from the drop down menus at the top of the window. The following functions can be executed from the tool bar:



**New...** This creates a completely new “Setup” file, loaded with the default variables and the word “[unnamed]” is displayed in the title bar. If any changes were made to the existing file, the user is prompted to save changes to the existing file.



**Open...** This displays a dialog box allowing the user to select an existing setup data file to open. The name of the file will be displayed in the title bar.

**Note:** Existing “.NCM” files created by the DOS based (NETCOM) setup program can be opened as well.



**Save** This displays a dialog box allowing the user to select a folder and enter a name for the file. The user will be notified if the file already exists and the extension “.sdt” will automatically be added to the file name. If this is a “New” file, the user will be prompted to enter a file name.



**Print** This allows the user to generate a “Report” printout of all the communication settings and node descriptions contained in the current setup data file.



**Offline Edit** This function allows the user to perform “Offline” editing on the current set-up data file. This allows the user the ability to make any necessary changes to the communication settings while not online with the processor.

## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE



**Upload (save) Settings** This allows the user to upload (save) communication settings from the S3016-505 to a “Set-up Data” file.



**Download (restore) Settings** This allows the user to download communication settings to the S3016-505. This can be the default setting or from a “Set-up Data” file.



**Target Board Interface** This allows the user to view fault codes, S3000 network communication error codes and review the current “Ident” and “Revision” of the application program loaded into the S3016-505.



**Communication Interface** This allows the user to view the data that is being passed to and from the TI processor and the slave node.

**Status Bar:** At the bottom of the window is the “Status Bar”. The status bar is used to display system messages, online or offline mode, as well as, the current time and date as set by the operating system. The system messages panel displays general information about operation of the system. The Online/Offline mode panel displays the status of the current set-up program mode of operation. The mode of operation can be changed by simply double clicking the online/offline mode panel.

**Hot Keys:** Hot keys are activated by holding down the “ALT” key and simultaneously pressing the underlined letter of the desired function. Almost every function can be activated by either pressing a series of hot keys or using the “TAB” key to move between fields.

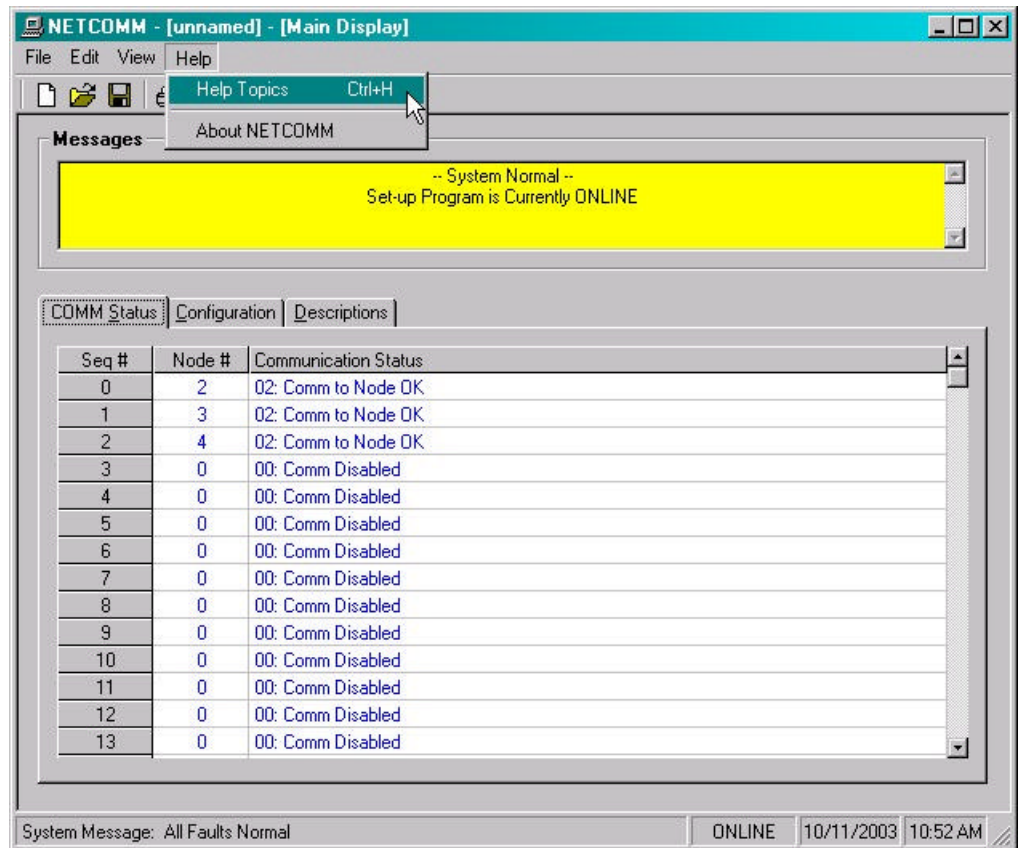
**Online/Offline Modes:** The set-up program allows the user to make changes while “Online” with the processor. The “Offline” mode is used to preset communication settings prior to download. All functions are available to the user while “Online”, however specific “Online” functions are disabled in the “Offline” edit mode.

**Note:** Offline changes can only be made by enabling “Offline Editing”, accessed under the “Edit” menu or selecting the “Edit” icon from the tool bar.

## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

**Getting Help:** Pressing “Ctrl+H” will display the help file window. Pressing the F1 key will display the contents file. Hot spots allow jumps to other topics to display additional information as desired. Selecting “About NETCOMM” from the Help menu will display a dialog box listing information about the current revision of the setup program and how to obtain technical support.



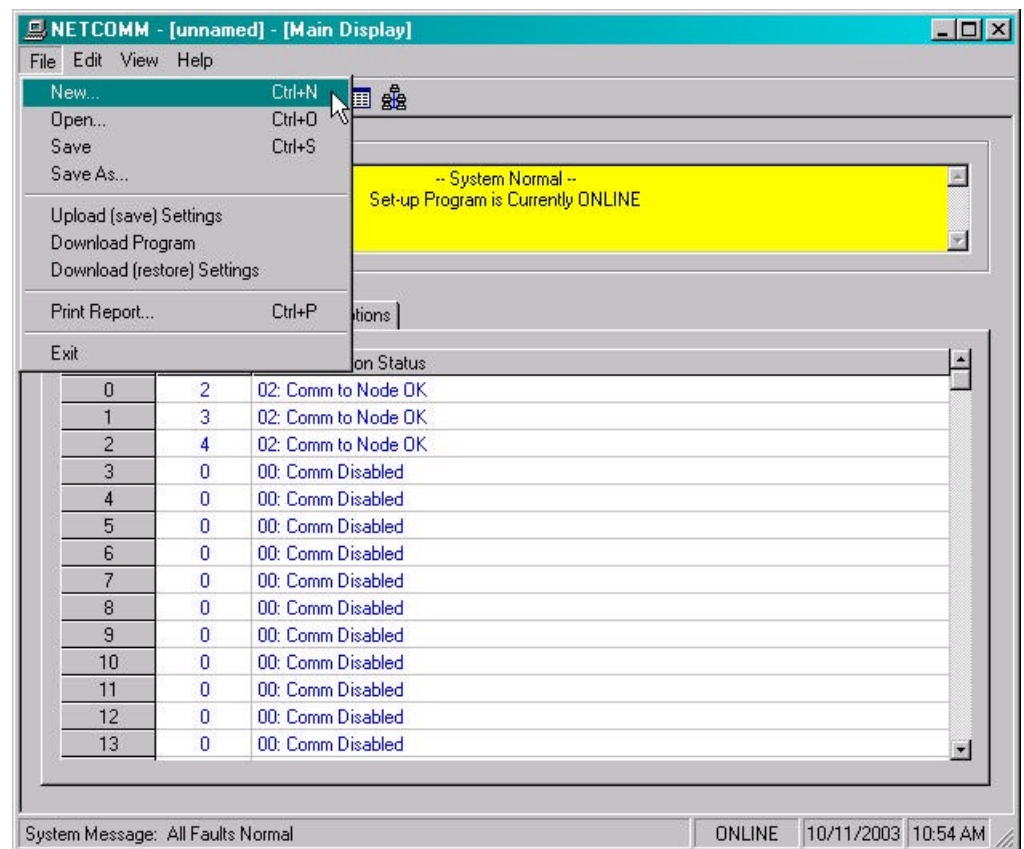
## SECTION 3

# NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

### 3.2 THE FILE MENU

The “File” menu allows the user to perform the following functions:

- Create a New “Set-up Data” file
- Open an existing “Data” (.sdt) or “Netcom” (.ncm) file
- Save any changes made to the communications settings to disk
- Upload (save) Settings from the Processor
- Download a SYSdev (.sdv) program to the S3016-505 (or any other Systems processor)
- Download (restore) Settings from a “Set-up Data” file to the processor
- Print a Report of the communication settings
- Exit the set-up program



## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

---

#### 3.2.1 THE SET-UP DATA FILE

The “Set-up Data” file (.sdt) is a binary access file, designed for fast file I/O operation. When the set-up program is first invoked, the default communication settings are loaded into memory. If changes are made to any of these settings (either online or offline), the user will be flagged to “Save Changes” upon exit of the program.

**Note:** Any windows based “Set-up” program can open a “Set-up Data” file, however, the data tables will not be properly aligned. The user will be alerted to the problem if a set-up data file has been created by either a different set-up program or a different revision of the software.

The “Set-up Data” file is similar to that of a word processing file. When the program is loaded, the default communication settings are loaded and the user is able to make any changes as desired. The set-up program is unaware of the settings and parameters that exist within the S3016-505. Therefore, to normalize the set-up program with the processor, the user should define or open an existing file, then upload “All” variables from the processor or click on the “Save” icon. This allows the user to either create a backup of the data or maintain an existing file. The user can even open a data file for another S3016-505, save the file to a new name, make the necessary changes and simply download the new settings to another S3016-505.

The following functions can be accessed any time:

**New:** To create a “New” data file, select “New” from the “File” menu, press “Ctrl + N” or click on the “New...” icon. This replaces the current settings with the default settings. If any changes were made to the existing settings or file, the user is prompted first to save changes to the existing file.

**Note:** The file name “[unnamed]” is displayed in the title bar when a new file is first created.



## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

**Open:** To “Open” and existing data (.sdt) or Netcom (.ncm) file, select “Open” from the “File” menu, press “Ctrl + O” or click on the “Open...” icon. This displays a dialog box allowing the user to select an existing file to open (default is “Setup Files”, \*.sdt”. The name of the file is then displayed in the title bar.

**Note:** If any changes were made to the existing settings/file, the user will be prompted to save any changes before opening a new file.

**Save:** To “Save” the current settings to disk, select “Save” from the “File” menu, press “Ctrl + S” or select the “Save” icon. This displays a dialog box allowing the user to select a folder and enter a name for the file.

**Note:** If this is a “New” file, the user will be prompted first to enter a file name. The user will be notified if the file already exists and the extension “.sdt” will be added automatically.

**Save As:** To save the settings to a “New” file name, select “Save As” from the “File” menu. This displays a dialog box allowing the user to select a folder and enter in a new name for the file. The user will be notified if the file exists and the extension “.sdt” will be added automatically to the file name.

**Upload (save) Settings:** The “Set-up” program allows the user to upload communication settings from the S3016-505. This function is accessed from the “File” menu or by selecting the “Upload” icon.

Once invoked, the user is prompted to confirm their choice. If the set-up program is in an “Offline” edit mode, the user is prompted to go online with the processor to continue the process. If communication can’t be established or the user cancels the operation, the settings will not be uploaded from the processor.

**Download (restore) Settings:** The set-up program allows the user to download “Set-up” communication settings to the S3016-505. This function is accessed from the “File” menu or by selecting the “Download” icon.

**Note:** Only the values loaded (from either a data or Netcom file) are used. If the validity of the communication settings is questionable, review the data in an “Offline” mode prior to download.

## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

---

#### 3.2.2 DOWNLOAD PROGRAM

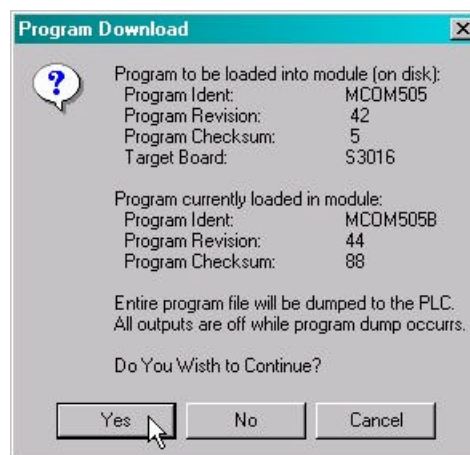
The “Set-up” program allows the user to “Download” any SYSdev program to the S3016-505.

**Note:** To “Download” a SYSdev program to the processor, the set-up program must be “Online” with the S3016-505. If “Online” mode cannot be achieved, program download will not be executed. If the program is currently “Offline”, the user will be prompted to first go “Online”.

Once selected, and the set-up program “Online” with the processor, a dialog box will be displayed, allowing the user to select the SYSdev file to download.

**Note:** Only the files with the “.sdv” file extension will be displayed. It is important to keep in mind that only a valid S3016-505 SYSdev file should be downloaded to the processor. Care must be taken when selecting a program to download.

A message box is then displayed informing the user of the current program, revision and checksum of the program loaded in the processor, as well as, that of the selected program.



The user must confirm their selection by clicking the “Yes” command button. Program download is initiated and the current program download address is displayed. When program download is complete, the user is alerted. Control is passed back to the main program and the set-up program will remain in an “Online” edit mode.

## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

---

#### 3.2.3 PRINT REPORT

The “Set-up” program allows the user to generate a “Report” printout of all the communication settings and node descriptions. This function is accessed from the “File” menu.

At the top of each page, the report displays the name of the set-up file being printed. At the bottom of each page is the date and time the document was printed, as well as, the page number.

From the “File” menu, select “Print Report”, press “Ctrl + P” or select the “Print” icon. This displays the “Print Setup” dialog box, allowing the user to select a printer, as well as, the paper size and orientation. Once the user selects “OK”, the report is generated and sent to the specified printer device. This function makes use of the windows print manager, which allows the user to continue with their work while the document is being printed.

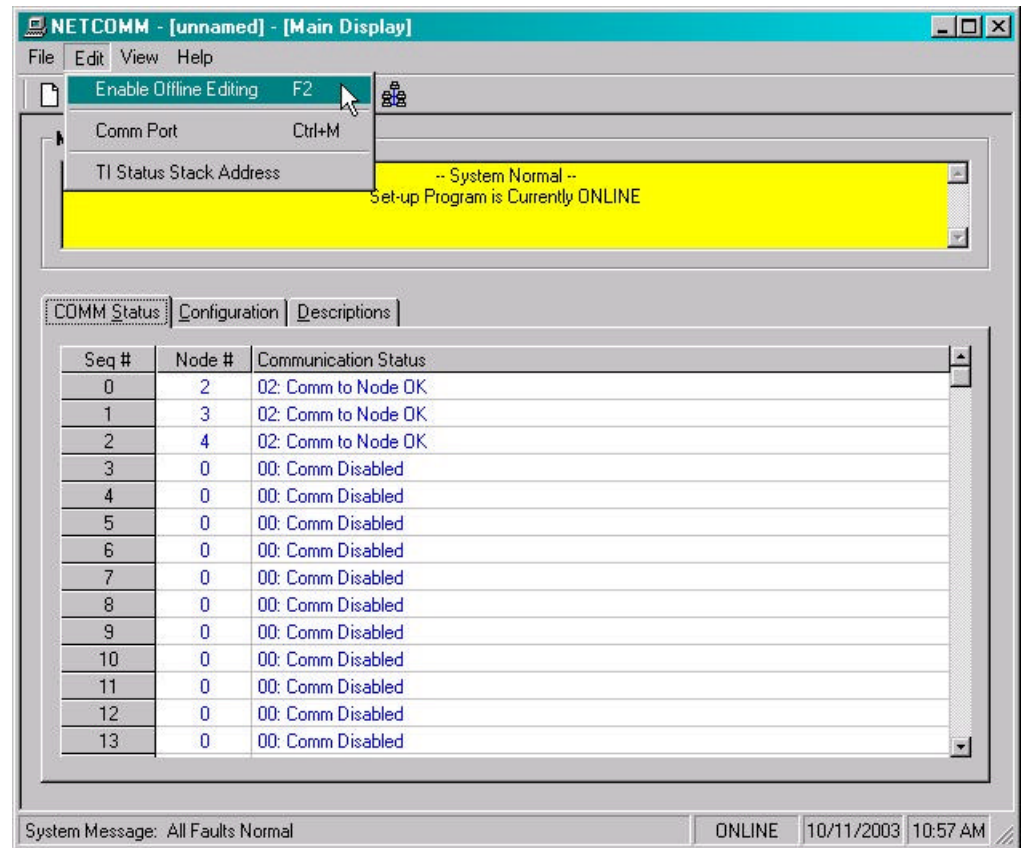
## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

#### 3.3 THE EDIT MENU

The “Edit” menu allows the user to perform the following functions:

- Enable/Disable Offline Editing
- Set-up the Comm Port
- Set the TI Status Stack Address



## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

---

#### 3.3.1 ENABLE OFFLINE EDITING

This function allows the user to perform “Offline” editing on the currently loaded communication settings or edit the node descriptions. This allows the user the ability to make any necessary changes to the communication settings while not online with the S3016-505.

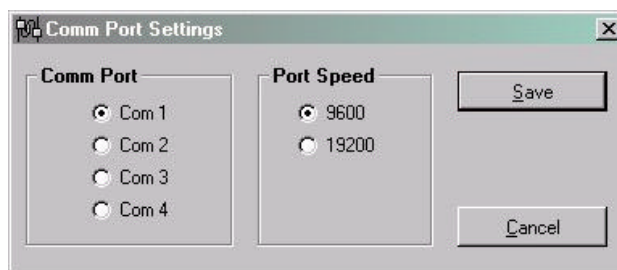
**Note:** If offline editing is not enabled, the user is only able to view the settings and descriptions. When the program is first invoked, offline editing is disabled.

The user will need to select “Enable Offline Editing” from the edit menu, press function key F2 or select the “Offline Edit” icon to enable/disable this feature.

---

#### 3.3.2 SETUP COMM PORT

This function allows the user to specify the serial communications port and baud rate to talk to the S3016-505. The programming port of the S3016-505 is set to 9600 baud.



Once selected, a dialog box requesting the user to select a “Comm Port” and “Baud Rate” will be displayed. The default setting is COM1 at 9600 baud.

In most cases, the user will only need to specify the communications port and leave the baud rate at 9600. If communication problems occur, make sure there is a secure connection from the PC to the PLC. Then check the Comm port. In most cases the user will only need to select a new Comm port. If communication problems persist, there may be another program causing a conflict with the port. Check the port configuration from the Windows “Settings” folder.

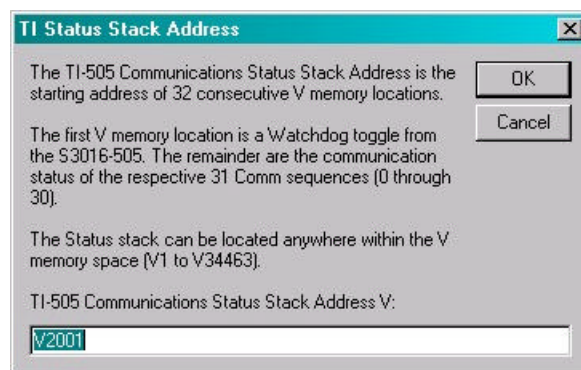
## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

---

#### 3.3.3 TI STATUS STACK ADDRESS

The TI-505 Communications Status stack is 32 consecutive V memory locations in the TI 505 processor, representing the communications status of each node in the S3016-505. Once selected, an “Input Box” is displayed, allowing the user to change the address. If an invalid address is entered, the new address will not be written to the processor and the user alerted to the error. This selection is accessed from the “Edit” menu.



**Note:** If “Offline” with the processor and “Offline Editing” is not enabled, this function is disabled.

The first V memory location is a watchdog toggle from the S3016-505. This is toggled between 0 and 1 once all the enabled nodes are updated in the S3016-505. This should be monitored for a change of state by the TI-505 program to verify that the S3016-505 is functioning. This is generally accomplished with a timer that is reset by a one shot fired every time this V memory location changes from either 0 to a 1 or a 1 to a 0.

**Note:** While the Node parameters are being downloaded to the S3016-505, that this location will not change state. The timer should be set to the maximum download time (approximately 5 seconds).

### **SECTION 3**

## **NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE**

The remaining “V” memory locations are the status of the respective 32 communication sequences (the second V memory location is for sequence 0, the third for sequence 1, etc.). The values loaded into these locations are numbers between 0 and 19 decimal. These are the same status codes as displayed in the “Comm Status” tab selection. See section 3.5.1 for definitions of these codes. These status registers should be monitored by the TI-505 program to verify communications to a specific node is occurring.

When this selection is made, the current TI-505 Communications Status stack address is displayed. To change this address, simply type in the desired V memory address and press <ENTER> otherwise, press <ESC>. The address is then written to S3016-505.

## SECTION 3

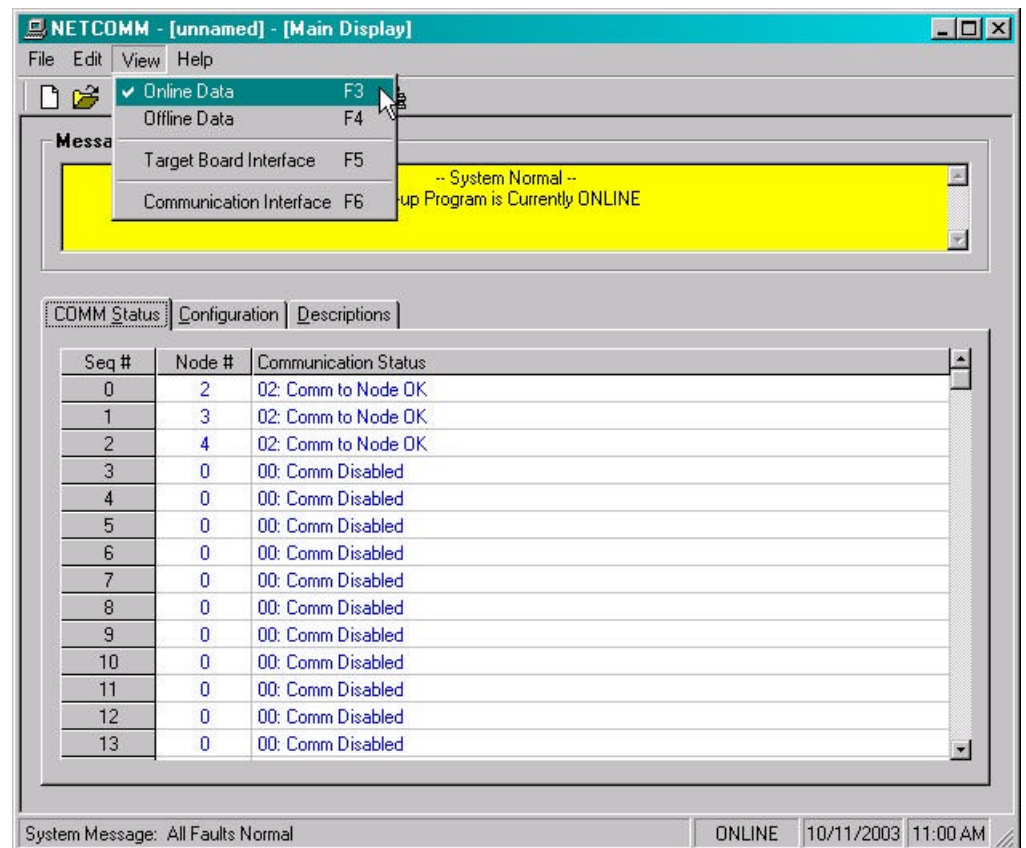
### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

---

#### 3.4 THE VIEW MENU

The “View” menu allows the user to perform the following functions:

- View “Online” Data
- View “Offline” Data
- View the “Target Board Interface”
- View the “Communications Interface”





## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

---

#### 3.4.1 VIEW ONLINE DATA

This function allows the user to place the set-up program in an “Online” mode with the processor. This is accessed from the “View” menu by selecting “Online Data” or by simply pressing the “F3” function key.

**Note:** The program can be toggled between “Offline” and “Online” modes by simply “Double Clicking” on the “Online” or “Offline” panel displayed in the status bar at the bottom of the window.

Once invoked, the set-up program will attempt to open the Comm port and communicate with the S3016-505. If the set-up program is unsuccessful, a warning message will be displayed prompting the user to either “Retry” or “Cancel” the operation. If the operation is canceled and communication with the processor cannot be established the system will be placed in an “Offline” edit mode.

**Note:** Anytime while the set-up program is “Online” with the processor and communication is interrupted, a warning message will be displayed, prompting the user to either “Retry” or “Cancel” the operation.

---

#### 3.4.2 VIEW OFFLINE DATA

This function allows the user to place the set-up program in an “Offline” mode. This is accessed by the “View” menu, by selecting “Offline Data” or by simply pressing the “F4” function key. This allows the user to perform “Offline” editing. All values displayed in “Offline” edit mode reflect the actual values loaded into memory from either a “Set-up Data” file or the default settings.

**Note:** The program can be toggled between “Online” and “Offline” by simply double clicking on the “Online” or “Offline” panel displayed in the status bar at the bottom of the window.

Once invoked, the set-up program will close the Comm port and cease communication with the S3016-505.

## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

---

#### 3.4.3 TARGET BOARD INTERFACE

This function allows the user to view fault codes, S3000 network communication error codes, review the current “Ident” and “Revision” of the application program, set the serial network node address and set the time and date of the S3016-505. This is accessed from the “View” menu or by clicking the Icon or by pressing the F5 key.

Once invoked, the set-up program will prompt the user to select a program to compare with the one existing in the processor. Whether a program is selected or the user cancels, the setup program will attempt to communicate with the S3016-505. If unsuccessful a warning message will be displayed, prompting the user to either “Retry” or “Cancel” the operation. If the operation is canceled and communication with the processor cannot be established the system will be placed in an “Offline” mode, however the “Target Board Interface” window will be displayed.

The screenshot shows the 'Target Board Interface' window with a tabbed interface. The 'Fault Codes / Status' tab is selected. It contains two main sections: 'Internal Fault Code' and 'Communications Network Error Codes'. Each section displays 'Current' and 'Last' error codes (all are 00H) and their descriptions (all are 'No Internal Fault has Occurred' or 'No Network Comm Error'). There is a 'Reset Faults' button in the internal fault section and an 'Ok' button at the bottom right.

Section	Field	Value
Internal Fault Code	Current Fault: Code =	00H
	Current Fault Description	No Internal Fault has Occurred.
	Last Fault: Code =	00H
	Last Fault Description	No Internal Fault has Occurred.
Communications Network Error Codes	Current Comm Error: Code =	00H
	Current Comm Error Description	No Network Comm Error.
	Last Comm Error: Code =	00H
	Last Comm Error Description	No Network Comm Error.

## SECTION 3

# NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

### 3.4.4 COMMUNICATION INTERFACE

This function allows the user to view the data as it's passed from the TI processor to any selected slave node. This is accessed from the "View" menu, by clicking the icon or pressing the F6 key.

**Note:** The entire "Send" and "Receive" buffer for each sequence is displayed and is only available when the setup program is "Online" with the S3016-505.

Once invoked, the "Communications Interface" window is displayed. The user can select which sequence number to display the "Send" and "Receive" buffer, as well as view the particular communication configuration. The data displayed is dynamic and reflects that which is being communicated between the TI processor and the slave node.

See section 3.5.2 for a description of the "Settings" fields.

The screenshot shows the "Communication Interface" window. At the top, it has a title bar with a close button. Below the title bar, there are fields for "Sequence Number" (set to 0), "Node Number" (set to 2), "Description" (Deco 1 Status Data (M4500)), and "Status" (02: Comm to Node OK). Below these fields is a "Settings" section with two tables. The first table is for "Send" settings and the second is for "Receive" settings. Both tables have columns for "Setting" and "Value". The "Send" table has values 120, V100, and W1000. The "Receive" table has values 120, W2000, and V220. Below the settings section is a "Comm Data" section with a table showing "Word", "Send Data", and "Receive Data". The table has 10 rows, with "Word" values from 1 to 10, "Send Data" values from 2100 to 2118, and "Receive Data" values from 2000 to 2018. At the bottom right of the window is an "Ok" button.

Send	Setting	Receive	Setting
# Words	120	# Words	120
From TI	V100	From Node	W2000
To Node	W1000	To TI	V220

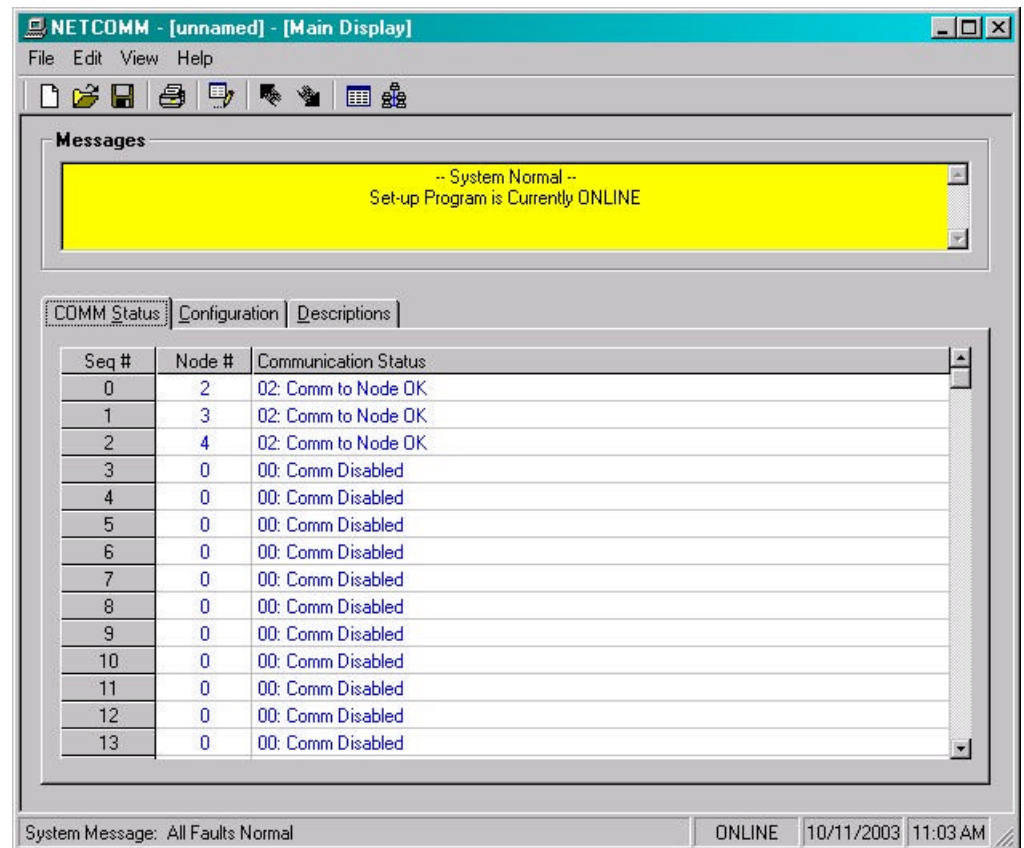
Word	Send Data	Receive Data
1	2100	2000
2	2102	2002
3	2104	2004
4	2106	2006
5	2108	2008
6	2110	2010
7	2112	2012
8	2114	2014
9	2116	2016
10	2118	2018

## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

#### 3.5 THE MAIN DISPLAY

The “Main Display” displays the communication status of the S3016-505, as well as allows the user to view/edit communication settings and descriptions for each communication sequence.



**Note:** Selecting a cell/parameter will allow the user to enter in a new value. The “COMM Status” table is “Read Only” and will only display the communication status when “Online” with the processor.

The variables are displayed in blue when either “Online” or when “Offline Edit” is enabled. Selecting a cell/parameter to edit will display an entry box, with the current variable selected for editing, displayed black.

Pressing the “ESC” key once will restore the original value, pressing it a second time will end the edit operation.

## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

Pressing the “Enter/Return” (or the up/down cursor) keys will validate the entry. If the entry is valid, the setting is updated. If the setup program is “Online” with the processor, the value is immediately written to the S3016-505 and focus is moved to the next cell. If an invalid entry is entered, a message box will be displayed indicating the error and the focus returned to the original cell/parameter.

The following is a list of the functions of the “Main Display”.

---

#### 3.5.1 THE COMM STATUS TAB

This selection is used to display the communications status with other nodes while “Online” with the S3016-505. The status of each communication sequence is displayed in a table with a status code and short description. The status codes are the same as those passed to the TI-505 processor in the “TI-505 communications status stack”.

The list of possible codes is shown below:

<b>Code (HEX)</b>	<b>Code (decimal)</b>	<b><u>Definition</u></b>
00H	00	Comm to Node Disabled
01H	01	-----
02H	02	Comm to Node OK
03H	03	More than one bus master
04H	04	Xmitt timeout - no response
05H	05	No slave response - timeout
06H	06	Invalid command from master
07H	07	Receive overflow
08H	08	Receive collision detected
09H	09	Receive alignment error
0AH	10	Receive CRC error
0BH	11	Unknown error
0CH	12	Xmitt no acknowledge
0DH	13	Xmitt under run error
0EH	14	Xmitt collision detected
0FH	15	Address range error
10H	16	Unexpected slave response
11H	17	TI-505/S3016 read error
12H	18	TI-505/S3016 write error
13H	19	Undefined error

## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

Status code 00H “Comm to Node Disabled” is set when node value of a communications sequence is set to “0”. This is the normal status code for any unused nodes on the network. Status code 02H “Comm to Node OK” is the normal status code when a node is enabled for communications (valid node number of 2 to 32) and communication to that node is successful. Status codes 03H through 10H are network communication error codes. In this case, communication to the node is enabled, however communications to that node was unsuccessful. The data returned to the TI-505 processor for that node would not be valid.

Refer to the S3016 User's Manual for a complete description of these error codes.

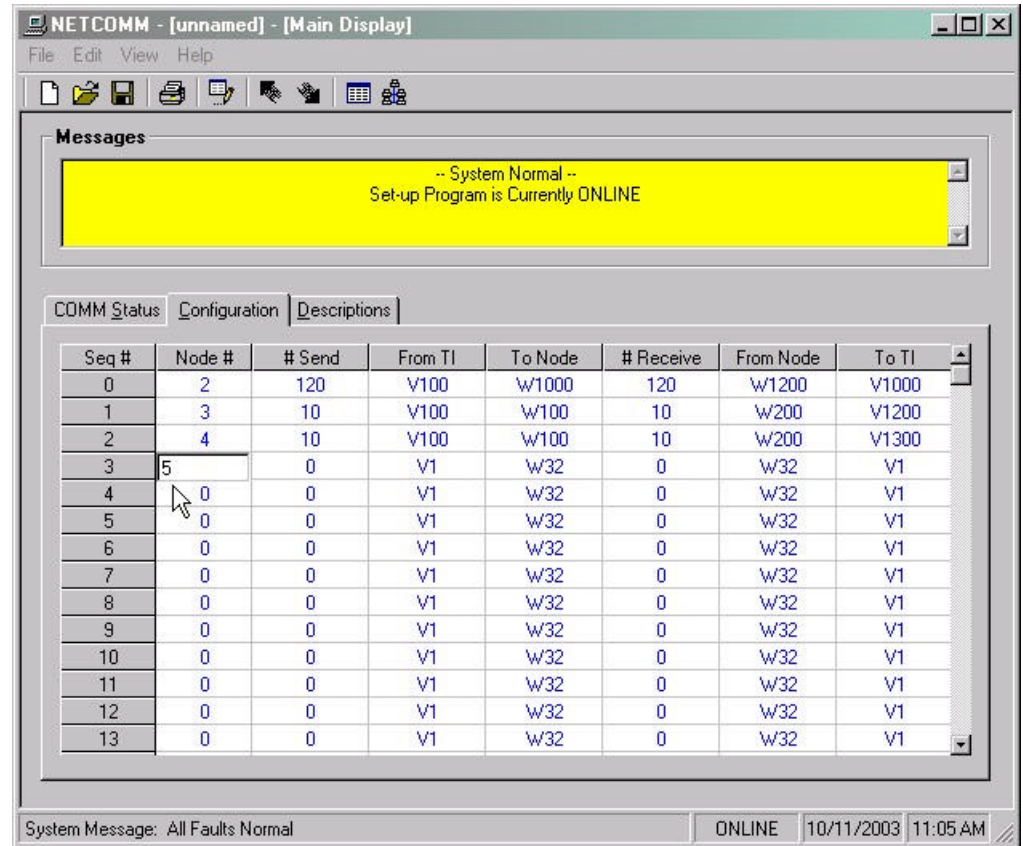
**Note:** This table is updated continuously such that any change in the communications status of any node is reflected in the table immediately. If the RS-232 communications from the computer to the S3016-505 cannot be established (cable not connected, etc.) or the setup program is in an “Offline” edit mode, the status table will display “--: -” for each communication sequence in the table. This indicates that the setup program is not “Online” with the S3016-505.

## SECTION 3

# NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

### 3.5.2 THE CONFIGURATION TAB

The configuration tab is used to define the communications parameters for each sequence.



The following selections are available:

**Node Number:** This field is used to specify the slave node number for communications. When selected, enter a node number (0 or 2 through 32). Entering a “0” will disable the communications, entering any number from 2 through 32 will enable communications to the specified node. In addition, the next or previous node can be selected by pressing the “Up” or “Down” cursor keys respectively.

**Note:** When configuring communications settings “Online” it is best to set the “Node Number” last after setting the other parameters. When “Online”, parameters are written directly to the processor, immediately after being entered. Setting the “Node Number” last will then enable communication to slave.

## SECTION 3

### NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

**Sending (writing) Data to Slave Node:** The following three fields are the communication parameters, which define the words sent (written) from the TI-505 to the slave node:

**# Send - # of Words to Send to Slave Node:** This is the number of V memory words that will be transmitted to the slave node. This is a decimal value between 0 and 120.

**From TI - Starting Address in TI-505 of Words to Send:** This is the first address of a stack in the TI-505 processor that will be sent to the respective slave node. Valid variable type is V memory only within the V memory space of the TI 505 processor.

**To Node - Starting Address in Slave Node to Store Data:** This is the first address of a stack in the slave node where the words sent from the master will be stored. This is a 'W' word address in the slave node processor (see respective user's manual for slave node processor for valid 'W' addresses).

**Receiving (reading) Data from Slave Node:** The following three fields are the communication parameters, which define the words received (read) from the slave node and stored in the TI-505 processor:

**# Receive - # of Words to Read from Slave Node:** This is the number words that will be read from the slave node and stored in V memory locations of the TI-505 processor. This is a decimal value between 0 and 120.

**From Node - Starting Address in Slave Node to Read Data:** This is the first address of a stack in the slave node that will be read from the slave node and sent to the TI-505. This is a 'W' word address in the slave node (see respective user's manual for slave node processor for valid 'W' addresses and external memory (HEX) addresses).

**To TI - Starting Address in TI-505 to store Data:** This is the first address of a stack in the TI-505 where the words read from the slave node will be stored. This is a V memory location in the TI-505. Valid variable type is V memory only within the V memory space of the TI 505 processor.

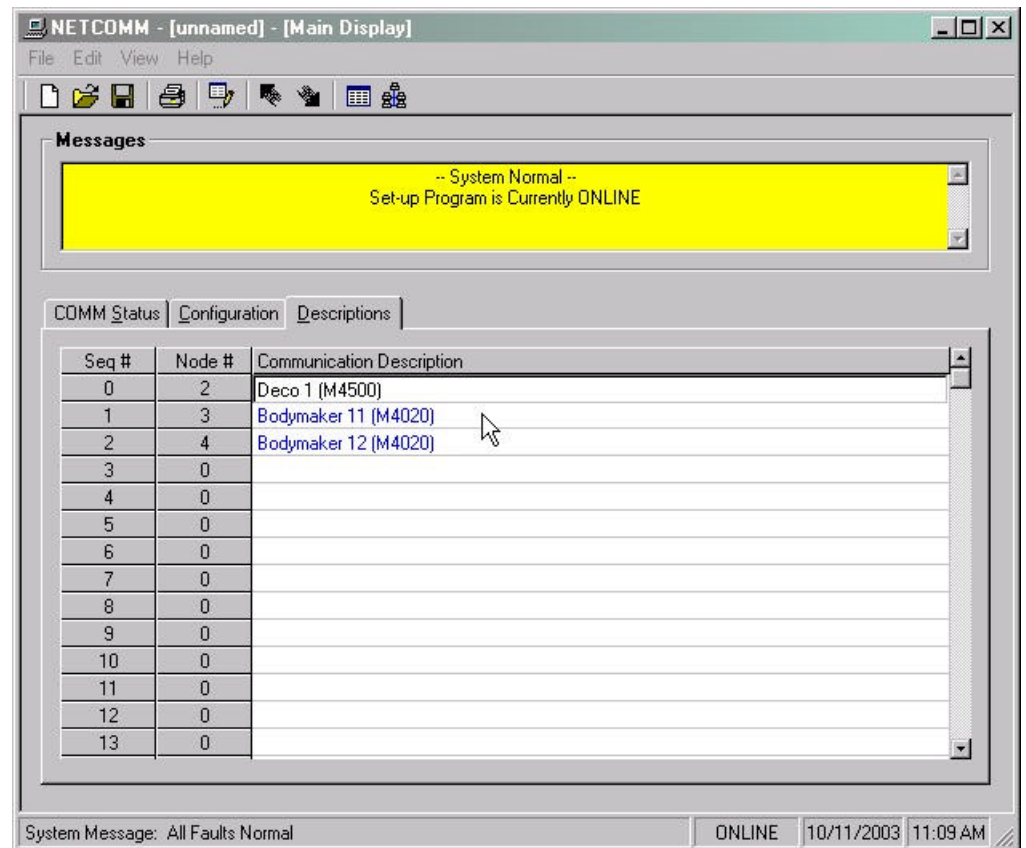


## SECTION 3

# NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE

### 3.5.2 THE DESCRIPTIONS TAB

This is the description for the node (i.e. type of machine, type of slave processor, etc., whatever is desired by the user to describe the node up to 50 characters).



### **SECTION 3**

## **NETCOMM WINDOWS BASED SETUP PROGRAM REFERENCE**

*(This Page Intentionally Left Blank)*

## **SECTION 4**

### **NETCOM DOS BASED SETUP PROGRAM REFERENCE**

The DOS based NETCOM setup program is used in conjunction with the “MCOM505” application program for the S3016-505. The combination of the “MCOM505” and “NETCOM” programs turn the S3016-505 into a communications board that allows the TI 505 processor to communicate with up to 31 S3000/M4000/S3016-505 slave nodes, reading and writing up to 120 words to each node.

NETCOM allows the user to perform the following:

- 1) Upload Node communication parameters from S3016-505.
- 2) Download Node communication parameters to S3016-505.
- 3) Save Node communication parameters in a file.
- 4) Document slave node descriptions.
- 5) Print Node communication parameters and descriptions.
- 6) Create multiple network configuration files with up to 32 nodes per file.

The “NETCOM” program consists of a shell (the menu displayed when “NETCOM” is first invoked), used for file and directory control. From this menu files, which will store the node parameters, are created, copied, backed up or restored.

When a file is selected or created, the Main menu is then invoked. The user can then edit the node parameters, download, upload or prints the communication parameters. The following sections describe the menus and features of the “NETCOM” setup program.

---

#### **4.1 INSTALLING AND RUNNING NETCOM**

To install the program on a PC or Laptop computer, perform the following:

- 1) Load the diskette into the floppy drive.
- 2) From a DOS command prompt, switch to the root directory and type “A:INSTALL”.
- 3) The install program will create a directory call “HSLSETUP” and copy the “NETCOM.EXE” program to this directory.

To execute the setup program, run the NETCOM.EXE file from the “HSLSETUP” directory. The program will be invoked and the “NETCOM” shell will be displayed.

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

If desired, the “NETCOM” shell can be bypassed by including the path (directory) and parameters file name in the command line when “NETCOM” is invoked. This allows a specific file to be automatically invoked without having to go through the “NETCOM” shell for file selection. The format of this is as follows:

```
HSLSETUP>NETCOM path filename<ENTER>
```

Where “path” is the directory\sub-directories that the file name is under and “filename” is file that contains the respective node parameters.

**Note:** There must be a space between “NETCOM” and “path” and a space between “path” and “filename” in the above command line. No extension is appended to “filename” in this command line as well.

**Example:**

```
C:\HSLSETUP\NETCOM.EXE \MCOM505\ NETTEST
```

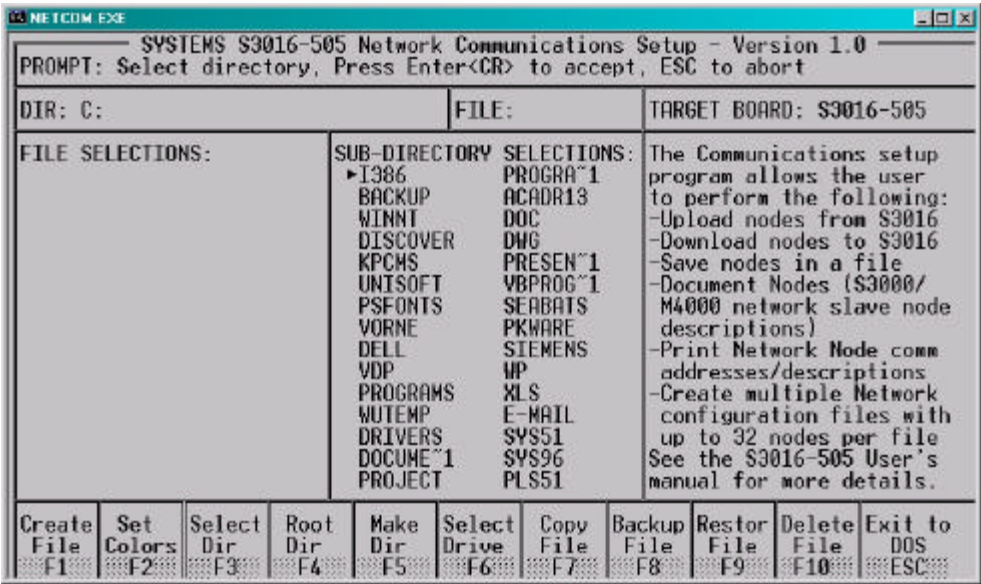
Where “MCOM505” is a directory off the root of the current drive that contains a file named “NETTEST”.

**Note:** Only an existing file can be accessed when adding the path and filename to the command line. New files must be created through the “NETCOM” shell by invoking “NETCOM” with no path and filename specified.

**SECTION 4**  
**NETCOM DOS BASED**  
**SETUP PROGRAM REFERENCE**

**4.2 “NETCOM” SHELL**

The “NETCOM” shell is used to create and edit user files and automatically invoke the main menu when a file is created or selected. Sub-directories should be used to store and organize the user files. These sub-directories can be created using the “F5: Make Dir” command in the “NETCOM” shell.



## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

---

#### 4.2.1 “NETCOM” SHELL MENU

The “NETCOM” shell consists of a menu which displays the selected file name, current directory, target board along with fields that display the existing file selections available for editing and available directories. The definitions of these fields are as follows:

**PROMPT:** This field displays various prompts to the user based on the selected command, informing the user what to do.

**DIR:** This field displays the currently selected drive and directory. This is set using the “F6: Select Drive:” and “F3: Select Dir” commands. When “NETCOM” is initially invoked, this is set to the root directory of the current drive.

**FILE:** This is the file currently pointed to by the selection arrow in the File Selections field of the menu. This field is also used to enter the name of the user file when the “F1: Create File” command is executed.

#### **FILE**

**SELECTIONS:** This field contains a list of the existing user files in the currently selected directory. The currently selected file is the file pointed to by the selection arrow. The selection arrow can be moved to any displayed file using the Left, Right, Up and Down arrow keys. When “NETCOM” is initially invoked, this displays all the “NETCOM” user files in the root directory of the current drive.

#### **SUB-DIRECTORY**

**SELECTIONS:** This field contains a list of the existing sub-directories in the currently selected directory. The “F3: Select Dir” command is used to select one of these directories as the current directory. When initially invoked, this displays all the sub-directories in the root of the current drive (whether they contain “NETCOM” user files or not).

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

---

#### 4.2.2 "NETCOM" SHELL COMMANDS

The shell contains numerous commands for creating and editing the user file, selecting the directory and drive the user files are stored in, and execution DOS commands such as Backup, Copy, Delete, etc. from within the shell. The definitions of these commands are as follows:

**F1:**

**Create File**

This command is used to create a new user file. When selected, the menu will prompt for the file name. Enter the new name in the "FILE:" field of the menu using the valid DOS file name character set.

**F2:**

**Set Colors**

This selection allows the user to select the foreground color (characters) and background color on PC's equipped with color monitors. Any of 16 foreground colors and any of 8 background colors can be selected.

**F3:**

**Select Dir**

This is used to select a directory. Only directories within the current sub-directory are displayed and are available for selection.

Pressing "F3" positions the selection arrow in the sub-directories menu. Use the cursor keys to position the selection arrow at the desired sub-directory and press ENTER. The "DIR:" field will reflect the selected directory, while the file selections menu will be updated to show the existing user files.

**F4:**

**Root Dir**

This selection is used to set "DIR:" to the root directory of the selected drive. This command is primarily used to back out of the selected directory. Pressing "F4" will set "DIR:" to the root directory and display the user files and directories in the root directory of the selected drive.

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

#### **F5:**

##### **Make Dir**

This selection creates a new sub-directory in the currently selected directory. When selected, the menu will prompt for the directory name. Enter the new name using the valid DOS directory name character set. The directory name can be a maximum of eight characters in length. Press ENTER to accept the directory name. The new directory will now be displayed in the sub-directories selection menu and can now be selected as the current directory using the “F3: Select Dir” command if desired.

#### **F6:**

##### **Select Drive**

This selection is used to change the currently selected drive. When selected, the shell prompts for the drive letter (A – Z). Enter the new drive and press ENTER. The “DIR:” field will be changed to the root directory of the new drive and the existing “NETCOM” user files and sub-directories will be displayed.

#### **F7:**

##### **Copy File**

This selection is used to copy the selected file to a new file name in the current directory. Enter the new name using the valid DOS file name character set. The file name can be a maximum of eight characters in length (no extension should be entered). Press ENTER to accept the file name that the selected file will be copied to. The new file will then be displayed in the file selections menu.

#### **F8:**

##### **Backup File**

This selection is used to backup the selected file to the root directory of a user specified diskette drive. When selected, the shell prompts for the drive (A or B) that the file will be backed up to. Enter the drive and press ENTER. The currently selected file will be copied to the root directory of the specified diskette.



## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

#### **F9:**

**Restore File** This selection is used to restore a previously backed up file from the root directory of a user specified drive to the currently selected directory and file. When selected, the shell prompts for the diskette drive (A or B) that the file will be copied from. Enter the drive and press ENTER. The file will be copied from the root directory of the specified drive to the currently selected directory.

#### **F10:**

**Delete File** This selection deletes the selected file from the current directory. When selected, the shell prompts one time to verify that the file is to be deleted, answer "Y" to delete, "N" to abort. If yes, the file is deleted from the file selections menu.

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

---

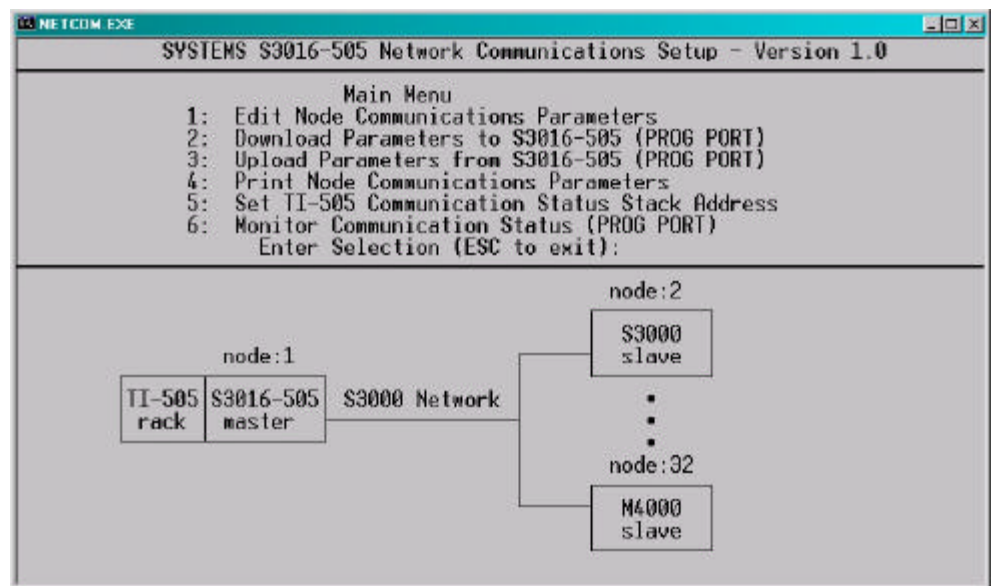
#### 4.3 MAIN MENU

The Main menu is invoked after a new file is created or when an existing file is selected from the shell.

The Main menu allows the user to edit the node communications parameters, download node parameters, upload node parameters, print node parameters, set the TI-505 status stack address and monitor the communications status on-line.

**Note:** When a file is created for the first time, the “TI-505 Communications Status Stack address” menu is displayed first (see section 4.8). This forces the user to set this address prior to editing the rest of the communications parameters.

The following describes these menu selections.

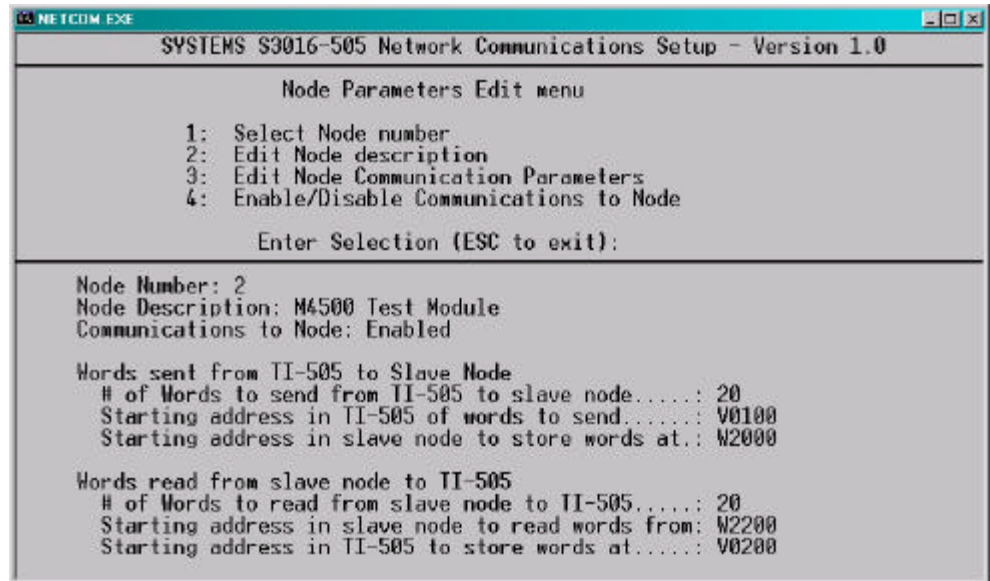


## SECTION 4

# NETCOM DOS BASED SETUP PROGRAM REFERENCE

### 4.4 EDIT NODE COMMUNICATIONS PARAMETERS

The Node Parameters Edit menu is used to define the communications parameters of each node as well as enable or disable communications to the node and enter descriptions for each node. The following selections are available on this menu:



In addition to the above selections, this menu contains the following fields:

**Node Number:** This is the currently selected node for editing.

**Node Description:** This is the description for the node (i.e. type of machine, type of slave processor, etc., whatever is desired by the user to describe the node up to 50 characters).

**Communications to Node:** Enabled/Disabled -This field indicates whether communications to the node is enabled or disabled. When enabled the S3016-505 communicates to the respective node using the parameters defined for that node (number of words, addresses to transfer, etc.). This information is then communicated to the TI processor. When disabled, no communications is performed to the respective node and no information is transferred to the TI processor. The communication parameter fields are also blanked when the comm is disabled.

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

**Words sent from TI-505 to Slave Node:** The following three fields are the communication parameters, which define the words sent from the TI-505 to the slave node:

**# of Words to send from TI-505 to slave node:** This is the number of V memory words that will be transmitted to the slave node. This is a decimal value between 0 and 120.

**Starting address in TI-505 of words to send:** This is the first address of a stack in the TI-505 processor that will be sent to the respective slave node. Valid variable type is V memory only within the V memory space of the TI 505 processor.

**Starting address in slave node to store words at:** This is the first address of a stack in the slave node where the words sent from the master will be stored. This is a 'W' word address in the slave node processor (see respective user's manual for slave node processor for valid 'W' addresses).

**Words read from slave node to TI-505:** The following three fields are the communication parameters, which define the words read from the slave node and stored in the TI-505 processor:

**# of Words to read from slave node to TI-505:** This is the number words that will be read from the slave node and stored in V memory locations of the TI-505 processor. This is a decimal value between 0 and 120.

**Starting address in slave node to read words from:** This is the first address of a stack in the slave node that will be sent to the TI-505. This is a 'W' word address in the slave node (see respective user's manual for slave node processor for valid 'W' addresses and external memory addresses).

**Starting address in TI-505 to store words at:** This is the first address of a stack in the TI-505 where the words read from the slave node will be stored. This is a V memory location in the TI-505. Valid variable type is V memory only within the V memory space of the TI 505 processor.

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

---

#### 4.4.1 SELECT NODE NUMBER

This selection is used to select the slave node number for editing. When selected, enter the node number (2 through 32) to be edited. In addition, the next or previous node can be selected by pressing the “PgDn” or “PgUp” keys respectively.

---

#### 4.4.2 EDIT NODE DESCRIPTION

This selection is used to edit the “Node Description” field. This field can be used to describe the slave node (type of machine, type of processor, etc.). When selected, the cursor is placed at the beginning of this field. Then enter the desired node description (up to 50 characters).

---

#### 4.4.3 EDIT NODE COMMUNICATIONS PARAMETERS

This selection is used to modify the communication parameters. This includes:

- # of Words to send from TI-505 to slave node
- Starting address in TI-505 of words to send
- Starting address in slave node to store words at
- # of Words to read from slave node to TI-505
- Starting address in slave node to read words from
- Starting address in TI-505 to store words at

If the node is not already enabled for communications when this selection is made, the user is prompted to enable communications. If communications is not enabled, the communication parameters cannot be set and the cursor will return to the Enter selection prompt.

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

With the communications enabled, the following sequence will occur:

- 1) The cursor will be located in the “# of Words to send” field. Enter the number of words to be sent and press <ENTER>.
- 2) The cursor will move to the “Starting address in TI-505” field. Enter the starting V memory address of the send stack (the “V” is automatically prefixed to the beginning of the address) and press <ENTER>.
- 3) The cursor will now move to the “Starting address in the slave node” field. Enter the starting address in the slave that the words sent will be stored at as either a valid “W” address in the form “Wxxxx” where xxxx is the address or as a valid external memory HEX address in the form “xxxxH” where xxxx is the HEX address and press <ENTER>.
- 4) The cursor will now advance to the next three fields of the “Words read from slave node to TI-505”. Enter these in the same way that the three previous fields were entered.

---

#### 4.4.4 ENABLE/DISABLE COMMUNICATIONS TO NODE

This selection is used to enable or disable communications to the node. When the node file is first created, by default, communications to all nodes is disabled. The user then enables communications only to the respective nodes that are connected to the network. For instance, if the S3016-505 is connected to only one other slave node, communications to that node only should be enabled and all other nodes should be disabled. If the S3016-505 is connected to two slave nodes, communications to those two nodes should be enabled and all other nodes disabled and so on.

This selection simply toggles the “Enable” or “Disable” state of the node. If the node is disabled, pressing “4: Enable/Disable” will enable the node. If the node is disabled, pressing “4: Enable/Disable” will disable the node.

**Note:** When the node is enabled, the communication parameters values are shown in the parameter fields. When disabled, the fields are blanked, even though the parameter values in those fields are still retained in memory and are shown when the communications is enabled again.

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

When any of the descriptions or parameters are modified for any of the nodes, the data is saved in the selected file when the “Node Parameters Edit” menu is exited. The S3016-505 is not updated with any changes until the “Download Parameters” is performed to the S3016-505.

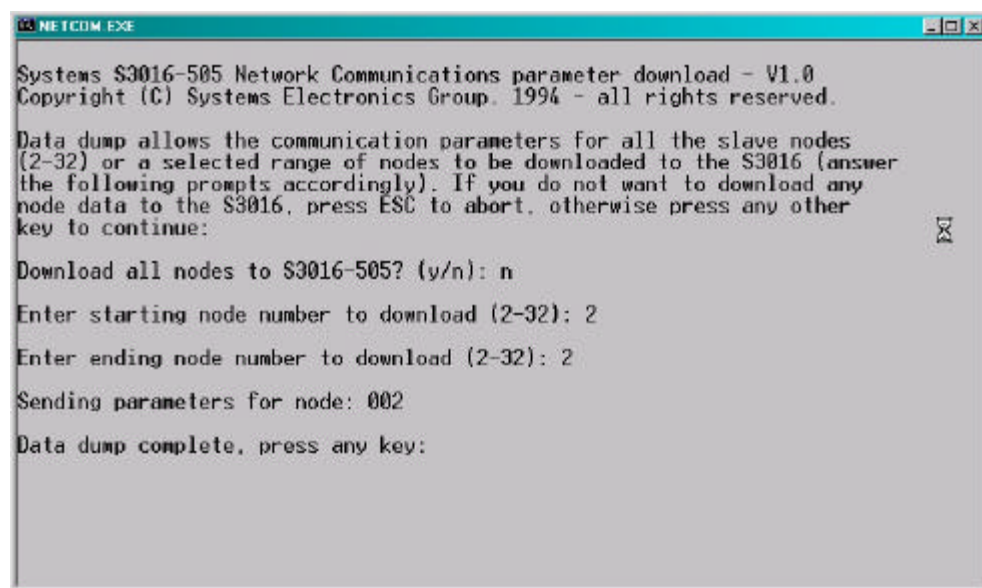
---

#### 4.5 DOWNLOAD PARAMETERS TO S3016-505

This selection is used to download the node communications parameters for the slave nodes to the S3016-505 from the computer.

**Note:** The node descriptions are not downloaded to the S3016-505, these are saved in the file on disk only. The download feature allows the user to either download all the nodes (2 to 32) or just a range of nodes (“starting” node and “ending” node) as specified by the user. This allows just a certain number of nodes (i.e. nodes 2 to 4) to be downloaded without having to download all the parameters for all the nodes.

To download the node parameters, connect an RS-232 cable from the COM port on the computer to the “PROG” port on the S3016-505 and press this selection. The user is then prompted to abort the download (by pressing “ESC”) or to continue (by pressing any other key). If any key other than “ESC” is pressed, the following prompt is displayed:



```
NETCOM EXE
Systems S3016-505 Network Communications parameter download - V1.0
Copyright (C) Systems Electronics Group. 1994 - all rights reserved.

Data dump allows the communication parameters for all the slave nodes
(2-32) or a selected range of nodes to be downloaded to the S3016 (answer
the following prompts accordingly). If you do not want to download any
node data to the S3016, press ESC to abort, otherwise press any other
key to continue:

Download all nodes to S3016-505? (y/n): n
Enter starting node number to download (2-32): 2
Enter ending node number to download (2-32): 2
Sending parameters for node: 002
Data dump complete, press any key:
```

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

If “n” is answered, the user is prompted for the starting node number. Enter the starting node number and press Enter. The user is then prompted for the ending node. Enter the ending node number and press Enter. If, for instance, the parameters for nodes 10 through 20 are to be downloaded, Enter 10 at the starting node prompt and enter 20 at the ending node prompt.

In both cases, once the node download is initiated, the current node number being downloaded is displayed on the computer.

**Note:** Communication on the S3000 network, as well as between the TI processor and S3016-505 is halted while the download is in progress. Communication will automatically resume as soon as the download is complete.

---

#### 4.6 UPLOAD PARAMETERS FROM S3016-505

This selection is used to upload the node communications parameters for the slave nodes from the S3016-505.

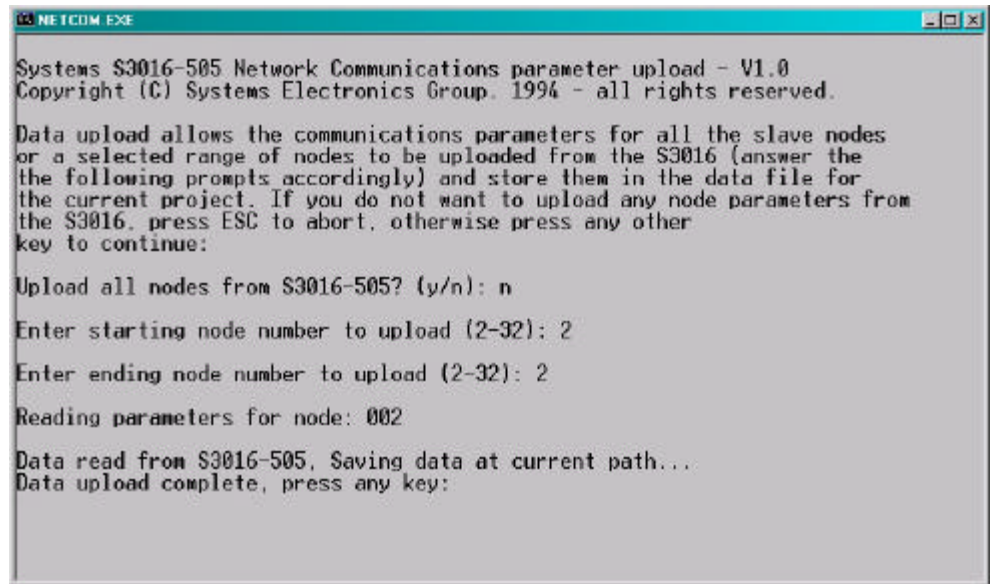
**Note:** The node descriptions are not uploaded these are saved in the file on disk only. The upload feature allows the user to either upload all the nodes (2 to 32) or just a range of nodes (“starting” node and “ending” node) as specified by the user. This allows just a certain number of nodes (i.e. nodes 2 to 4) to be uploaded without having to upload parameters for all the nodes.

To upload the node parameters, connect an RS-232 cable from the COM port on the computer to the “PROG” port on the S3016-505 and press this selection. The user is then prompted to abort the upload (by pressing “ESC”) or to continue (by pressing any other key). If any key other than “ESC” is pressed, the following prompt is displayed:



## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE



```
NETCOM EXE

Systems S3016-505 Network Communications parameter upload - V1.0
Copyright (C) Systems Electronics Group. 1994 - all rights reserved.

Data upload allows the communications parameters for all the slave nodes
or a selected range of nodes to be uploaded from the S3016 (answer the
the following prompts accordingly) and store them in the data file for
the current project. If you do not want to upload any node parameters from
the S3016, press ESC to abort, otherwise press any other
key to continue:

Upload all nodes from S3016-505? (y/n): n
Enter starting node number to upload (2-32): 2
Enter ending node number to upload (2-32): 2
Reading parameters for node: 002
Data read from S3016-505, Saving data at current path...
Data upload complete, press any key:
```

If “n” is answered, the user is then prompted for the starting node number. Enter the starting node number to be uploaded and press Enter. The user is then prompted for the ending node. Enter the ending node number and press Enter. If, for instance, the parameters for nodes 10 through 20 are to be uploaded, Enter 10 at the starting node prompt and enter 20 at the ending node prompt.

In both cases, once the upload is initiated, the current node number being uploaded is displayed on the computer.

**Note:** Communication on the S3000 network, as well as between the TI processor and S3016-505 is halted while the upload is in progress. Communication will automatically resume as soon as the upload is complete.

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

---

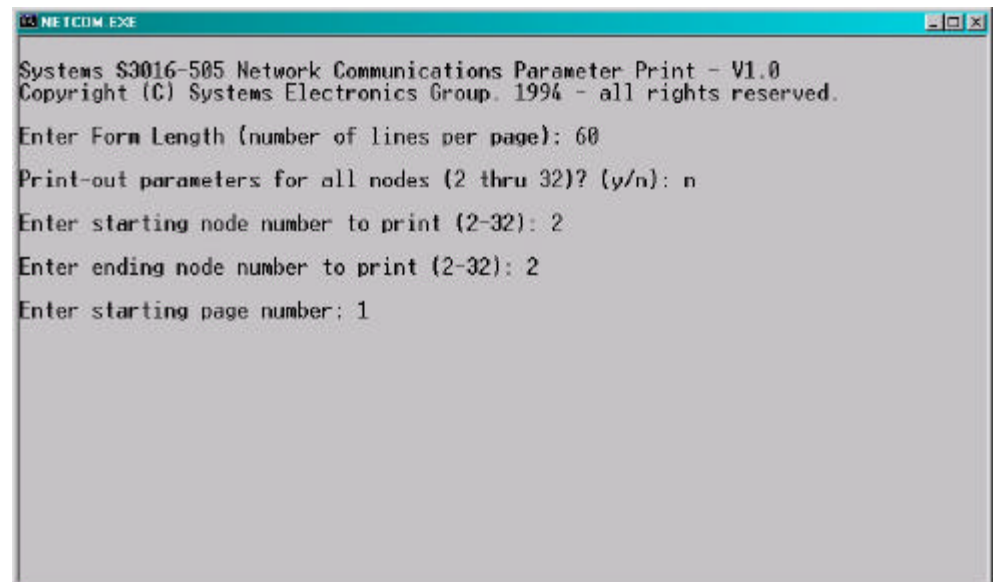
#### 4.7 PRINT NODE COMMUNICATIONS PARAMETERS

This selection prints all the information on the “Edit Node Communications Parameters” menu:

- The Node Number
- Node Description
- Communications Enabled/Disabled
- Words Sent From Ti-505 To Slave Node
- Words Read From Slave Node To Ti-505

If communications to a specific node is disabled, the Words sent and read fields will not be printed. Similar to the download and upload, the user can print the parameters for all the nodes (2-32) or just a range of nodes.

The user will first be prompted for the form length. This is the number of lines printed for each page and is typically 60. Enter the number of lines per page and press Enter. Next the user is prompted “Print-out parameters for all nodes (2 thru 32)? (y/n)”. If “n” is answered the user is then prompted for the starting node number to print. Enter the starting node number and press Enter. Next the ending node number is prompted for. Enter the ending node number and press enter. For instance, if nodes 10 thru 20 are to be printed, enter 10 for the starting node and 20 for the ending node.



## SECTION 4

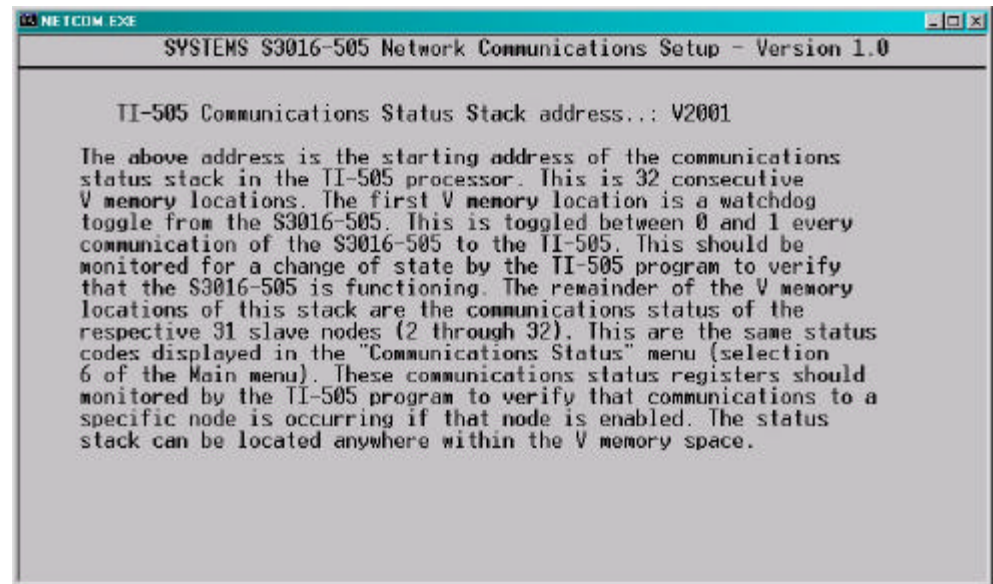
# NETCOM DOS BASED SETUP PROGRAM REFERENCE

Once the above prompts are answered, the print-out will be initiated. Pressing “ESC” or any other key once the print-out is initiated will abort the print-out.

---

### 4.8 SET TI-505 COMMUNICATIONS STATUS STACK ADDRESS

The TI-505 Communications Status stack is 32 consecutive V memory locations in the TI 505 processor which represent the communications status of each node in the S3016-505.



The first V memory location is a watchdog toggle from the S3016-505. This is toggled between 0 and 1 once all the enabled nodes are updated in the S3016-505. This should be monitored for a change of state by the TI-505 program to verify that the S3016-505 is functioning. This is generally accomplished with a timer that is reset by a one shot fired every time this V memory location changes from either 0 to a 1 or a 1 to a 0.

**Note:** While the Node parameters are being downloaded to the S3016-505, that this location will not change state. Thus the timer should be set to the maximum download time (approximately 5 seconds).

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

The remainder of the V memory locations are the communication status of the respective 31 slave nodes (the second V memory location is for node 2, the third for node 3, etc.). The values loaded into these locations are numbers between 0 and 19 decimal. These are the same status codes as displayed in the “Communications Status” menu (selection 6 of the Main menu). See section 4.9 for definitions of these codes. These status registers should be monitored by the TI-505 program to verify that communications to a specific node is occurring.

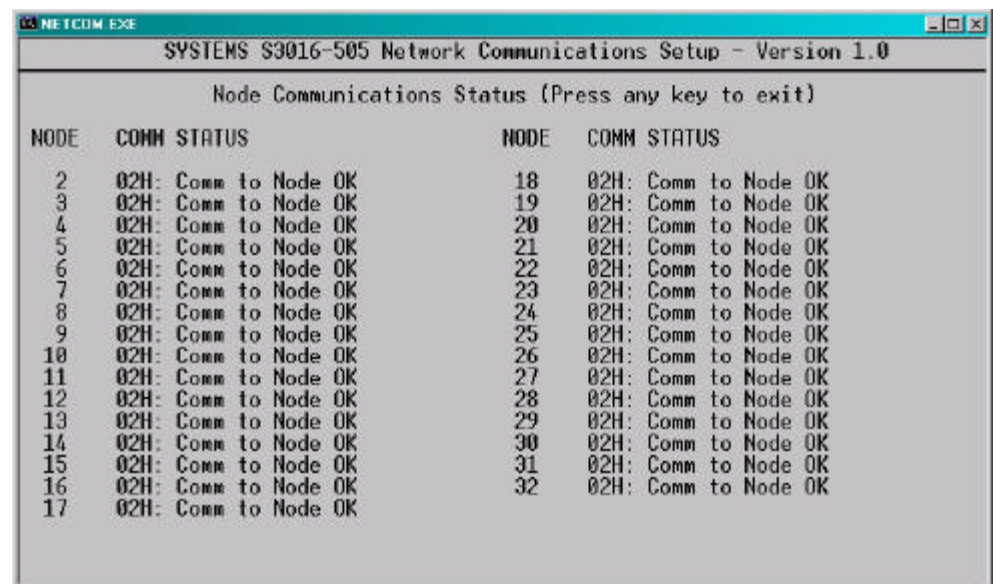
When this selection is made, the current TI-505 Communications Status stack address is displayed. To change this address, simply type in the desired V memory address and press <ENTER> otherwise, press <ESC>.

**Note:** This address is only loaded into the S3016-505 when the “Download Parameters to S3016-505” selection is performed. If this address is changed, the parameters must be download to update the S3016-505.

---

#### 4.9 MONITOR COMMUNICATIONS STATUS (PROG PORT)

This selection is used to monitor the communications status on-line. The status of each node is displayed in a table with a status code and short description of each code. They are the same status codes passed to the TI-505 processor in the “TI-505 communications status stack”.



The screenshot shows a DOS window titled "NETCOM EXE" with a subtitle "SYSTEMS S3016-505 Network Communications Setup - Version 1.0". The main content is a table titled "Node Communications Status (Press any key to exit)". The table has two columns: "NODE" and "COMM STATUS". It lists 32 nodes, each with a status code of "02H: Comm to Node OK".

NODE	COMM STATUS	NODE	COMM STATUS
2	02H: Comm to Node OK	18	02H: Comm to Node OK
3	02H: Comm to Node OK	19	02H: Comm to Node OK
4	02H: Comm to Node OK	20	02H: Comm to Node OK
5	02H: Comm to Node OK	21	02H: Comm to Node OK
6	02H: Comm to Node OK	22	02H: Comm to Node OK
7	02H: Comm to Node OK	23	02H: Comm to Node OK
8	02H: Comm to Node OK	24	02H: Comm to Node OK
9	02H: Comm to Node OK	25	02H: Comm to Node OK
10	02H: Comm to Node OK	26	02H: Comm to Node OK
11	02H: Comm to Node OK	27	02H: Comm to Node OK
12	02H: Comm to Node OK	28	02H: Comm to Node OK
13	02H: Comm to Node OK	29	02H: Comm to Node OK
14	02H: Comm to Node OK	30	02H: Comm to Node OK
15	02H: Comm to Node OK	31	02H: Comm to Node OK
16	02H: Comm to Node OK	32	02H: Comm to Node OK
17	02H: Comm to Node OK		

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

The list of possible codes is shown below:

<b><u>Code (HEX)</u></b>	<b><u>Code (decimal)</u></b>	<b><u>Definition</u></b>
00H	00	Comm to Node Disabled
01H	01	-----
02H	02	Comm to Node OK
03H	03	More than one bus master
04H	04	Xmitt timeout - no response
05H	05	No slave response - timeout
06H	06	Invalid command from master
07H	07	Receive overflow
08H	08	Receive collision detected
09H	09	Receive alignment error
0AH	10	Receive CRC error
0BH	11	Unknown error
0CH	12	Xmitt no acknowledge
0DH	13	Xmitt under run error
0EH	14	Xmitt collision detected
0FH	15	Address range error
10H	16	Unexpected slave response
11H	17	TI-505/S3016 read error
12H	18	TI-505/S3016 write error
13H	19	Undefined error

Status code 00H “Comm to Node Disabled” is set when communications to the respective node is disabled. This is the normal status code for any unused nodes on the network. Status code 02H “Comm to Node OK” is the normal status code when a node is enabled for communications and communications to that node is successful. Status codes 03H through 10H are network communication error codes. In this case, comm to the node was enabled but communications to that node was not successful. The data returned to the TI-505 processor for that node would not be valid.

Refer to the S3016 User's Manual for a complete description of these error codes.

## SECTION 4

### NETCOM DOS BASED SETUP PROGRAM REFERENCE

To view the communications status, connect an RS-232 cable from the COM port on the computer to the “PROG” port on the S3016-505 and press this selection. The status of each node will then be displayed in the status table.

**Note:** This table is updated continuously such that any change in the communications status of any node is reflected in the table immediately. If the RS-232 communications from the computer to the S3016-505 cannot be established (cable not connected, etc.), the status table will display “---: -----” for each node in the table. This indicates that the computer is not communicating to the S3016-505.

## SECTION 5 SPECIFICATIONS

Board Size:	
Length:	9.15"
Height:	6.30"
Width:	0.80"
Processor Memory:	
Program:	24K bytes battery backed CMOS RAM
Data:	
non-volatile:	2K bytes battery backed CMOS RAM
volatile:	
Flags (F):	104 bits
Bytes (B):	185 bytes
Words (W):	93 words
Interface Ports:	
PROG PORT:	
Type:	RS-232
Comm Rate:	9600 BAUD
USER PORT:	
Type:	RS-232/RS-422
Comm Rate:	300,600,1200,2400,4800,9600 BAUD
Start bits:	1
Data bits:	8
Stop bits:	1 or 2
Parity:	NONE, ODD, or EVEN
Serial Network:	
S3000-N1:	
Type:	RS-485
Comm Rate:	344KBPS, 229KBPS, or 106KBPS
# of nodes (max):	32
Isolation:	2000 VRMS
Distance:	1000 ft., 2000 ft., or 4000ft.
Protocol:	Proprietary
Power Requirements:	
Icc (+5VDC):	1.00 amps (MAX)

## SECTION 5

### SPECIFICATIONS

Temperature Range:	
Storage:	0 to 70 degrees C
Operating:	0 to 60 degrees C
Relative Humidity:	5 to 95% (non-condensing)



## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

#### System Configuration

-----

Target board: S3016 Communications co-processor I/O board

Network baud rate: 344KBPS

USER PORT baud rate: 9600

USER PORT parity: NONE

USER PORT stop bits: 1

Co-cpu communications interrupt enabled: no

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

```
*****  
block:  1 - High-level
```

```
0:sfunc07(1900H,B90);      /* restore TI-505 com stack address */  
1:sfunc07(1901H,B91);  
2:  
3:sfunc08(1ff4H,1);        /* set node address to 1 (master) */  
4:B105 = 2;                /* start in download mode until TI545 resets */  
5:F10 = 0;                 /* TI505 resetting */  
6:  
7:B59 = 12;                /* TI Task Wait */  
8:
```

F010	(TIreset)	TI505	reset	comp
B059	(TIWait )	TI Task	Code	Wait
B090	(Verrstk)	Vmem	error	stack
B091	(Verrstk)	Vmem	error	stack
B105	(dwnload)	down-	load	mode

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

```
*****  
block:  2 - Assembly  
  
0:          clr    EX1    ;disable comm interrupt from TI505  
1:
```

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

```
*****
block:  1 - High-level
```

```
** Select state **
```

This block searches for enabled slave nodes (1st address of respective slave buffer set to the slave number when enabled) and then initiates the comm cycle with the TI545. The data is read from the TI545 and stored in the slave xmit buffer and then the data in the slave rcve buffer is written to the TI545. Once all slave nodes (2-32) have been updated, the network comm (sfunc13) error stack is written to the TI545.

```
0:if (F0 == 1)                /* select state? */
1:  {
2:    F8 = 0;                  /* reset done bit */
3:    for (B42 = B61 + 1; B42 < 33 && F8 == 0; ++B42)
4:      {
5:        /* search for enabled slave nodes and comm with TI545 */
6:        W84 = (B42*512)+3c00H; /* point to nth node */
7:        sfunc07(W84,B87);
8:        if (B87 == B42)      /* slave node enabled? */
9:          {
10:             W84 = W84 + 2;    /* load comm parameters */
11:             sfunc07(W84,B62); /* # to send */
12:             W84 = W84 + 2;
13:             sfunc07(W84,B64); ++W84; /* VSRCE address */
14:             sfunc07(W84,B65); ++W84;
15:             sfunc07(W84,B66); ++W84; /* sdest address */
16:             sfunc07(W84,B67); ++W84;
17:             sfunc07(W84,B63); ++W84; /* # to receive */
18:             ++W84;
19:             sfunc07(W84,B68); ++W84; /* ssrce address */
20:             sfunc07(W84,B69); ++W84;
21:             sfunc07(W84,B70); ++W84; /* VDEST address */
22:             sfunc07(W84,B71);
23:
24:             W84 = W84 + 3;    /* point to xmit buffer */
25:             B61 = B42;      /* set slave address */
26:             F8 = 1;         /* done */
27:           }
28:      }
29:
30:  if (F8 == 0)                /* all nodes checked? */
31:    {
32:      B61 = 1;                /* start with first slave node again */
33:      B63 = 32;               /* write comm error status to TI545 */
34:      W70 = W90;              /* VDEST = VERROR_STACK */
35:      W84 = 1b00H;            /* point to comm error stack */
36:      sfunc07(1b00H,B87); /* toggle S3016-505 (TI545) watchdog */
37:      B87.0 = ~B87.0;
38:      sfunc08(1b00H,B87);
39:      F4 = 1;                 /* write comm error status */
40:    }
```

- block continued on next page -

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 1 - High-level

- block originated on prev page -

```

41:  else
42:      F1 = 1;                /* initiate read state */
43:  }
44:
45:F0 = 0;                    /* exit select state */
46:

```

F000	(select )	select	slave	state
F001	(read TI)	read	TI545	state
F004	(writeTI)	write	TI545	state
F008	( done )			done
B042	( (i) )		loop	(i)
B061	( slave )	network	slave	number
B062	( #send )	# of	words	to send
B063	( #rcve )	# of	words	to rcve
B064	( Vsrce )	Vmem	source	address
B065	( Vsrce )	Vmem	source	address
B066	( sdest )	slave	dest	address
B067	( sdest )	slave	dest	address
B068	( ssrce )	slave	source	address
B069	( ssrce )	slave	source	address
B070	( Vdest )	Vmem	dest	address
B071	( Vdest )	Vmem	dest	address
B087	(tempreg)		temp	registr
B087.0	(tregbt0)	temp	reg	bit0
W070	( Vdest )	Vmem	dest	address
W084	(slveptr)	slave	buffer	pointer
W090	(Verrstk)	Vmem	error	stack

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

```
*****
block:  2 - High-level
```

```
** Read state **
```

The next two blocks perform the first step of the communication cycle to the currently selected slave (read TI-545). In this state, the TI-545 Vmemory is read (the number of words specified in "#sent" starting at "Vsrce") and loaded into the respective slave xmit buffer (addresses 4000H to 7c00H). This is the data that will be sent to the respective slave node.

```
0:if (F1 == 1)      /* read state? */
1:  {
2:    if (F2 == 0)    /* read from 545 not yet initiated? */
3:    {
4:      B86 = &B182;    /* initiate pointer to task code read buffer */
5:      W88 = 1a00H;    /* initiate pointer to TI545 buffer */
6:      if (B62%15 == 0) /* even number of 15 word blocks to read? */
7:        B35 = B62/15; /* yes, calc # of blocks to read */
8:      else
9:        B35 = B62/15 + 1; /* calc # of blocks to read */
10:     W36 = W64-1;    /* initiate Vaddr in 545 to read from */
11:     F2 = 1;        /* read initiated */
12:     B39 = 6;        /* max number of error retries = 6 */
13:   }
14: }
15:
```

F001	(read TI)	read	TI545	state
F002	(rd init)	read	state	initiat
B035	(numblks)	# of	blocks	TI task
B039	(err cnt)	TI task	retry	count
B062	( #send )	# of	words	to send
B086	( pntr1 )	task	buffer	pointer
B182	( rbuff )	tskcode	read	buffer
W036	( Vaddr )	TI545	Vmem	address
W064	( Vsrce )	Vmem	source	address
W088	( pntr2 )	network	buffer	pointer

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 3 - High-level

```
0:if (F1 == 1)          /* read state? */
1:  {
2:    if (F103 == 0)      /* task code with 545 complete? */
3:    {
4:      if (F102 == 1)    /* task code comm failure with 545? */
5:      {
6:        --B39;          /* dec error count */
7:        if (B39 == 0)   /* error retries count out? */
8:        {
9:          B40 = 11H;    /* comm error = TI_READ_ERROR */
10:         W106 = 1b00H + ((B61-1)<<1);
11:         sfunc08(W106,B40); ++W106; /* save fault code in */
12:         sfunc08(W106,0);          /* TI-505 comm stat stack */
13:         F1 = 0, F2 = 0, F0 = 1;   /* exit read state */
14:       }
15:     }
16:     F103 = 1; /* retry task code comm with 545 */
17:     F102 = 0; /* reset task code comm fail bit */
18:   }
19:   else if (F9 == 1) /* task code comm done? */
20:   {
21:     for (B42 = 0; B42 < 15; ++B42) /* copy block read from 545 */
22:     {                               /* to network xmit buffer */
23:       sfunc08(W88,*B86); ++B86, ++W88; /* copy nth word */
24:       sfunc08(W88,*B86); ++B86, ++W88;
25:     }
26:     B86 = &B182; /* reset task code read buffer */
27:     F9 = 0;      /* block read done */
28:   }
29:   else if (B35 != 0) /* more blocks to read? */
30:   {
31:     B56 = 50H; /* block read task code */
32:     B57 = 4;   /* # of bytes to xmit in task code req */
33:     B58 = 32;  /* # of bytes to rcve in task code resp */
34:     W150 = W36; /* starting V mem address to read */
35:     W36 = W36 + 15; /* next block address to read */
36:     --B35; /* dec # of blocks to read */
37:     F103 = 1; /* initiate task code comm */
38:     F9 = 1; /* preset block read done */
39:   }
40:   else
41:   {
42:     W88 = 1a00H; /* copy data read from TI545 to */
43:     while (B62 != 0) /* respective slave xmit buffer */
44:     {
45:       sfunc07(W88,B87);
46:       sfunc08(W84,B87); ++W84, ++W88;
47:       sfunc07(W88,B87);
48:       sfunc08(W84,B87); ++W84, ++W88;
49:       --B62;
50:     }
```

- block continued on next page -

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 3 - High-level

- block originated on prev page -

```

51:          F1 = 0, F2 = 0, F4 = 1;      /* enter write state */
52:          W84 = (B61*512)+3c00H+256; /* point to respective rcve buffer */
53:          }
54:      }
55:  }
56:

```

```

F000 (select )      select  slave  state
F001 (read TI)      read    TI545  state
F002 (rd init)      read    state  initiat
F004 (writeTI)      write   TI545  state
F009 (TI done)      TI545   task    done
F102 (tskfail)      TI545   task    fail
F103 (TIcomm )      TI545   task    in prog
B035 (numblks)      # of    blocks  TI task
B039 (err cnt)      TI task  retry   count
B040 (commerr)      comm    error   code
B042 ( (i) )        loop    (i)
B056 (tskcode)      TI545   task    code
B057 (#xmitt )      task    code    #xmitt
B058 ( #rcve )      task    code    #rcve
B061 ( slave )      network  slave   number
B062 ( #send )      # of    words  to send
B086 ( pntrl )      task    buffer  pointer
B087 (tempreg)      temp    registr
B182 ( rbuff )      tskcode  read    buffer
W036 ( Vaddr )      TI545   Vmem    address
W084 (slveptr)      slave   buffer  pointer
W088 ( pntr2 )      network  buffer  pointer
W106 (statptr)      comm    status  pointer
W150 (Vstack )      Vmem    read    stack

```



# APPENDIX A

## MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 4 - High-level

\*\* Write state \*\*

The next two blocks write the data that was read from the slave to the TI-545 Vmemory starting at "Vdest" (the number of words written is "#rcve").

```

0:if (F4 == 1)          /* write state? */
1:  {
2:    if (F5 == 0)      /* write state not yet initiated? */
3:    {
4:      W88 = 1a00H;    /* copy data to be written to TI545 from */
5:      for (B42 = B63; B42 > 0; --B42) /* respective slave rcve buffer */
6:      {
7:        sfunc07(W84,B87);
8:        sfunc08(W88,B87);
9:        ++W84, ++W88;
10:       sfunc07(W84,B87);
11:       sfunc08(W88,B87);
12:       ++W84, ++W88;
13:     }
14:     B86 = &B152;    /* initiate pointer to task code write buffer */
15:     W88 = 1a00H;    /* initiate pointer to TI545 buffer */
16:     B35 = B63/14;    /* number of blocks to to write */
17:     B43 = B63%14;    /* number of words remaining to write */
18:     W36 = W70-1;    /* initiate Vmem dest address */
19:     F5 = 1;         /* write state initiated */
20:     B39 = 6;        /* maximum # of error retries = 6 */
21:
22:     /* reset TI_READ_ERROR */
23:     W106 = 1b00H + ((B61-1)<<1);
24:     sfunc07(W106,B108);
25:     if (B108 == 11H)
26:       sfunc08(W106,2);
27:   }
28: }
29:

```

F004	(writeTI)	write	TI545	state
F005	(wr init)	write	state	initiat
B035	(numblks)	# of	blocks	TI task
B039	(err cnt)	TI task	retry	count
B042	( (i) )		loop	(i)
B043	(#words)	# of	words	remain
B061	( slave )	network	slave	number
B063	( #rcve )	# of	words	to rcve
B086	( pntrl )	task	buffer	pointer
B087	(tempreg)		temp	registr
B108	(comstat)		comm	status
B152	( xbuff )	tskcode	write	buffer
W036	( Vaddr )	TI545	Vmem	address
W070	( Vdest )	Vmem	dest	address

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 4 - High-level

W084	(slveptr)	slave	buffer	pointer
W088	(pntr2)	network	buffer	pointer
W106	(statptr)	comm	status	pointer

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 5 - High-level

```
0:if (F4 == 1)          /* write state? */
1:  {
2:    if (F103 == 0)      /* task code with TI 545 complete? */
3:    {                  /* yes */
4:      if (F102 == 1)    /* task code comm failure with 545? */
5:      {
6:        --B39;          /* dec error count */
7:        if (B39 == 0)   /* maximum number of retries done? */
8:        {
9:          B40 = 12H;    /* comm error = TI_WRITE_ERROR */
10:         W106 = 1b00H + ((B61-1)<<1);
11:         sfunc08(W106,B40); ++W106;    /* save fault code in */
12:         sfunc08(W106,0);              /* TI-505 stat stack */
13:         F4 = 0, F5 = 0, F0 = 1; /* exit write state */
14:       }
15:     } else
16:       F103 = 1;    /* retry task code with 545 */
17:     F102 = 0;
18:   }
19:   else if (B35 != 0)    /* more blocks to write? */
20:   {
21:     B56 = 51H;          /* block write task code */
22:     B57 = 32;           /* number of task code bytes to write */
23:     B58 = 2;            /* number of task code bytes to read */
24:     W150 = W36;          /* Vmem start address */
25:     W36 = W36 + 14;      /* next block start address to write */
26:     --B35;              /* dec # of blocks to write */
27:     B86 = &B152;        /* point to task code write buffer */
28:     for (B42 = 0; B42 < 14; ++B42)
29:     {
30:       sfunc07(W88,*B86); ++W88, ++B86; /* copy from comm buffer */
31:       sfunc07(W88,*B86); ++W88, ++B86; /* to task code buffer */
32:     }
33:     F103 = 1;           /* initiate task code comm with 545 */
34:   }
35:   else if (B43 != 0)    /* write remaining words to 545? */
36:   {
37:     B56 = 51H;          /* block write task code */
38:     B57 = (B43 << 1) + 4; /* # of bytes to write */
39:     B58 = 2;            /* # of bytes to read in task code */
40:     W150 = W36;          /* Vmem start address */
41:     B86 = &B152;        /* point to task code write buffer */
42:     for (B42 = 0; B42 < B43; ++B42)
43:     {
44:       sfunc07(W88,*B86); /* copy nth word from comm buffer */
45:       ++W88, ++B86;      /* to task code write buffer */
46:       sfunc07(W88,*B86);
47:       ++W88, ++B86;
48:     }
49:     B43 = 0;            /* reset # remaining */
50:     F103 = 1;           /* initiate task code comm */
```

- block continued on next page -

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 5 - High-level

- block originated on prev page -

```

51:         }
52:     else
53:         F4 = 0, F5 = 0, F6 = 1;    /* enter done state */
54:     }
55: }
56:

```

```

F000 (select )    select  slave  state
F004 (writeTI)   write   TI545  state
F005 (wr init)   write   state  initiat
F006 (comdone)   TI545   comm   done
F102 (tskfail)   TI545   task   fail
F103 (TIcomm )   TI545   task   in prog
B035 (numblks)   # of    blocks  TI task
B039 (err cnt)   TI task  retry   count
B040 (commerr)   comm     error   code
B042 ( (i) )     loop     (i)
B043 (#words)    # of     words   remain
B056 (tskcode)   TI545   task    code
B057 (#xmitt)    task     code    #xmitt
B058 (#rcve)     task     code    #rcve
B061 ( slave )   network  slave   number
B086 ( pntr1 )   task     buffer  pointer
B152 ( xbuff )   tskcode  write   buffer
W036 ( Vaddr )   TI545   Vmem    address
W088 ( pntr2 )   network  buffer  pointer
W106 (statptr)   comm     status  pointer
W150 (Vstack)    Vmem     read    stack

```

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 6 - High-level

\*\* Comm network node select state \*\*

This block searches for enabled nodes and initiates the parameters and communications with any enabled nodes.

```
0:if (F3 == 0 && B105 == 0)          /* select comm node? */
1:  {
2:    F8 = 0;                          /* reset done bit */
3:    for (B42 = B96 + 1; B42 < 33 && F8 == 0; ++B42)
4:    {
5:      /* search for enabled nodes and initiate comm */
6:      W92 = (B42*512)+3c00H;          /* point to nth node */
7:      sfunc07(W92,B87);
8:      if (B87 == B42)                /* nth node enabled? */
9:      {
10:        B96 = B42;                  /* initiate comm with node */
11:        F8 = 1;
12:      }
13:    else
14:    {
15:      W106 = 1b00H + ((B42-1)<<1);
16:      sfunc08(W106,0); ++W106; /* set stat to "disabled" */
17:      sfunc08(W106,0); /* in TI-505 stat stack is not used */
18:    }
19:  }
20:
21:  if (F8 == 0)                      /* all nodes checked? */
22:    B96 = 1;                        /* yes, start again with first slave */
23:  else
24:  {
25:    W92 = W92 + 2;                  /* get parameters for nth node */
26:    sfunc07(W92,B97);              /* # to xmit */
27:    W92 = W92+4;
28:    sfunc07(W92,B66); ++W92; /* sdest */
29:    sfunc07(W92,B67); ++W92;
30:    sfunc07(W92,B98);              /* # to rcve */
31:    W92 = W92+2;
32:    sfunc07(W92,B68); ++W92; /* ssrce */
33:    sfunc07(W92,B69);
34:    W92 = W92+5;                    /* point to nth node xmit buffer */
35:    W94 = 7e0aH;                    /* point to sfunc13 buffer */
36:    for (B42 = 0; B42 < B97; ++B42)
37:    {
38:      /* copy nth node xmit buffer to sfunc13 buffer */
39:      sfunc07(W92,B87);
40:      sfunc08(W94,B87);
41:      ++W92, ++W94;
42:      sfunc07(W92,B87);
43:      sfunc08(W94,B87);
44:      ++W92, ++W94;
45:    }
```

- block continued on next page -

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 6 - High-level

- block originated on prev page -

```

46:      F3 = 1;      /* initiate comm state */
47:      }
48:  }
49:

```

F003	(netcomm)	network	comm	state
F008	( done )			done
B042	( (i) )		loop	(i)
B066	( sdest )	slave	dest	address
B067	( sdest )	slave	dest	address
B068	( ssrce )	slave	source	address
B069	( ssrce )	slave	source	address
B087	(tempreg)		temp	registr
B096	( node )	network	node	number
B097	( #xmit )	# of	words	to xmit
B098	( #rcve )	# of	words	to rcve
B105	(download)	down-	load	mode
W092	(nodeptr)	node	buffer	pointer
W094	( pnt3 )		buffer	pointer
W106	(statptr)	comm	status	pointer

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 7 - High-level

\*\* network comm state \*\*

This block performs the communications with the nth node, xmitting the previously loaded sfunc13 buffer and storing the data read from the node at the node rcve buffer.

```

0:if (F3 == 1)          /* comm state */
1:  {
2:    W94 = 0;           /* comm with nth node */
3:    B38 = sfunc13(B96,B97,*W94,*W66,B98,*W68,*W94);
4:    if (B38 == 2)      /* comm done? */
5:      {
6:        W92 = (B96*512)+3d00H; /* point to nth node rcve buffer */
7:        W94 = 7e0aH;         /* point to sfunc13 buffer */
8:        while (B98 != 0)     /* copy data rcved to node buffer */
9:          {
10:           sfunc07(W94,B87);
11:           sfunc08(W92,B87);
12:           ++W92, ++W94;
13:           sfunc07(W94,B87);
14:           sfunc08(W92,B87);
15:           ++W92, ++W94;
16:           --B98;
17:          }
18:      }
19:    if (B38 >= 2)      /* comm not busy? */
20:      {
21:        W94 = ((B96-1)<<1) + 1b00H; /* save error code */
22:        sfunc08(W94,B38); ++W94;
23:        sfunc08(W94,0);
24:        if (B38 != 2)    /* comm error? */
25:          B40 = B38;     /* save error code */
26:        W106 = 1b00H + ((B96-1)<<1); /* save comm return value in */
27:        sfunc07(W106,B108); /* test for TI_READ/WRITE error */
28:        if (B108 < 11H)
29:          {
30:            sfunc08(W106,B38); ++W106; /* TI-505 stat stack */
31:            sfunc08(W106,0);
32:          }
33:        F3 = 0;         /* exit comm state */
34:      }
35:  }
36:

```

F003	(netcomm)	network	comm	state
B038	(sf13ret)	sfunc13	return	value
B040	(commerr)	comm	error	code
B087	(tempreg)		temp	registr
B096	( node )	network	node	number
B097	( #xmit )	# of	words	to xmit
B098	( #rcve )	# of	words	to rcve

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 7 - High-level

B108	(comstat)		comm	status
W066	( sdest )	slave	dest	address
W068	( ssrce )	slave	source	address
W092	(nodeptr)	node	buffer	pointer
W094	( pptr3 )		buffer	pointer
W106	(statptr)	comm	status	pointer



## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 8 - High-level

```

0:/* test for download mode and wait until complete */
1:if (F6 == 1)
2:  {
3:    if (B105 == 0)          /* not in download mode? */
4:    {
5:      F0 = 1;              /* start next comm cycle with TI */
6:      F6 = 0;
7:    }
8:
9:    /* reset TI_WRITE_ERROR */
10:   W106 = 1b00H + ((B61-1)<<1);
11:   sfunc07(W106,B108);
12:   if (B108 == 12H)
13:     sfunc08(W106,2);
14:   }
15:
16:if (B105 != 0)             /* download mode? */
17:  {
18:    sfunc07(1900H,B90);    /* restore TI-505 comm status stack address */
19:    sfunc07(1901H,B91);
20:  }
21:
22:sfunc08(1ff4H,1);         /* set node address to 1 (master) */
23:

```

F000	(select )	select	slave	state
F006	(comdone)	TI545	comm	done
B061	( slave )	network	slave	number
B090	(Verrstk)	Vmem	error	stack
B091	(Verrstk)	Vmem	error	stack
B105	(dwnload)	down-	load	mode
B108	(comstat)		comm	status
W106	(statptr)	comm	status	pointer

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

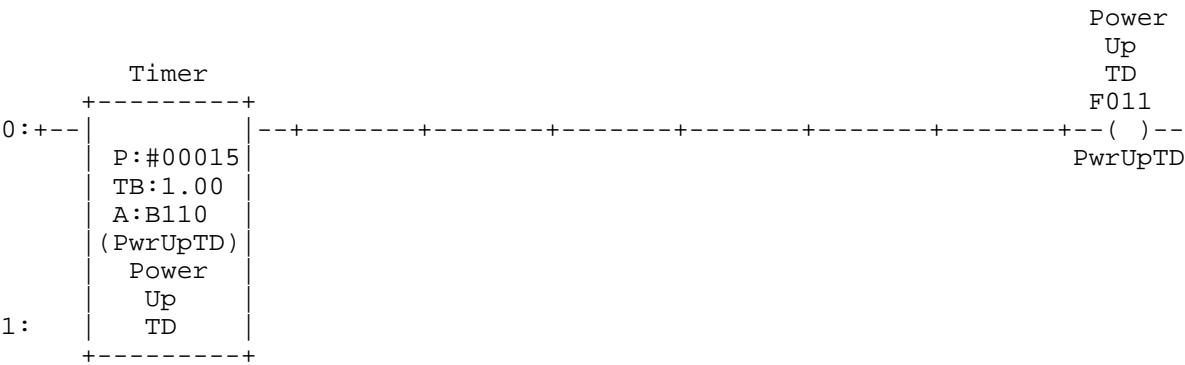
```
*****
block:  9 - Assembly
```

```
0:          jnb    F11,com.enb      ; power up TD complete?
1:          jnb    F10,com.enb      ; TI505 reset complete?
2:          anl    h'c0,#h'fb      ; reset TI bus complete latch
3:          orl    h'c0,#h'4
4:          jnb    P3.3,com.enb     ; TI505 still resetting?
5:          clr    IT1              ; level trigger on TI bus intrpt
6:          setb   EX1              ; enable TI505 bus intrpt
7:          setb   F10              ; TI505 reset complete
8:          mov    B105,#d'0        ; enable task code an network comm
9:          setb   F000              ; enable task code comm
10:;
11:com.enb:
12:;
13:
```

```
F000 (select )    select  slave  state
B105 (download)   down-   load   mode
```

APPENDIX A  
MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*  
block: 10 - Ladder



## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

```
*****
block:  1 - High-level

0:sfunc07(1900H,B90);      /* restore TI-505 com stack address */
1:sfunc07(1901H,B91);
2:
3:sfunc08(1ff4H,1);        /* set node address to 1 (master) */
4:B105 = 2;                /* start in download mode until TI545 resets */
5:F10 = 0;                 /* TI505 resetting */
6:
7:B59 = 12;                /* TI Task Wait */
8:

F010  (TIreset)    TI505  reset   comp
B059  (TIWait )    TI Task Code   Wait
B090  (Verrstk)    Vmem   error   stack
B091  (Verrstk)    Vmem   error   stack
B105  (dwnload)    down-  load    mode
```

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

```
*****  
block:  2 - Assembly  
  
0:          clr    EX1    ;disable comm interrupt from TI505  
1:
```

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

```
*****
block:  1 - High-level
```

```
** Select state **
```

This block searches for enabled slave nodes (1st address of respective slave buffer set to the slave number when enabled) and then initiates the comm cycle with the TI545. The data is read from the TI545 and stored in the slave xmit buffer and then the data in the slave rcve buffer is written to the TI545. Once all slave nodes (2-32) have been updated, the network comm (sfunc13) error stack is written to the TI545.

```
0:if (F0 == 1)                /* select state? */
1:  {
2:    F8 = 0;                  /* reset done bit */
3:    for (B42 = B61 + 1; B42 < 33 && F8 == 0; ++B42)
4:      {
5:        /* search for enabled slave nodes and comm with TI545 */
6:        W84 = (B42*512)+3c00H; /* point to nth node */
7:        sfunc07(W84,B87);
8:        if (B87 == B42)      /* slave node enabled? */
9:          {
10:             W84 = W84 + 2;    /* load comm parameters */
11:             sfunc07(W84,B62); /* # to send */
12:             W84 = W84 + 2;
13:             sfunc07(W84,B64); ++W84; /* VSRCE address */
14:             sfunc07(W84,B65); ++W84;
15:             sfunc07(W84,B66); ++W84; /* sdest address */
16:             sfunc07(W84,B67); ++W84;
17:             sfunc07(W84,B63); ++W84; /* # to receive */
18:             ++W84;
19:             sfunc07(W84,B68); ++W84; /* ssrce address */
20:             sfunc07(W84,B69); ++W84;
21:             sfunc07(W84,B70); ++W84; /* VDEST address */
22:             sfunc07(W84,B71);
23:
24:             W84 = W84 + 3;    /* point to xmit buffer */
25:             B61 = B42;      /* set slave address */
26:             F8 = 1;         /* done */
27:           }
28:      }
29:
30:  if (F8 == 0)                /* all nodes checked? */
31:    {
32:      B61 = 1;                /* start with first slave node again */
33:      B63 = 32;               /* write comm error status to TI545 */
34:      W70 = W90;              /* VDEST = VERROR_STACK */
35:      W84 = 1b00H;            /* point to comm error stack */
36:      sfunc07(1b00H,B87); /* toggle S3016-505 (TI545) watchdog */
37:      B87.0 = ~B87.0;
38:      sfunc08(1b00H,B87);
39:      F4 = 1;                 /* write comm error status */
40:    }
```

- block continued on next page -

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 1 - High-level

- block originated on prev page -

```

41:  else
42:      F1 = 1;                /* initiate read state */
43:  }
44:
45:F0 = 0;                    /* exit select state */
46:

```

F000	(select )	select	slave	state
F001	(read TI)	read	TI545	state
F004	(writeTI)	write	TI545	state
F008	( done )			done
B042	( (i) )		loop	(i)
B061	( slave )	network	slave	number
B062	( #send )	# of	words	to send
B063	( #rcve )	# of	words	to rcve
B064	( Vsrce )	Vmem	source	address
B065	( Vsrce )	Vmem	source	address
B066	( sdest )	slave	dest	address
B067	( sdest )	slave	dest	address
B068	( ssrce )	slave	source	address
B069	( ssrce )	slave	source	address
B070	( Vdest )	Vmem	dest	address
B071	( Vdest )	Vmem	dest	address
B087	(tempreg)		temp	registr
B087.0	(tregbt0)	temp	reg	bit0
W070	( Vdest )	Vmem	dest	address
W084	(slveptr)	slave	buffer	pointer
W090	(Verrstk)	Vmem	error	stack

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

```
*****
block:  2 - High-level
```

```
** Read state **
```

The next two blocks perform the first step of the communication cycle to the currently selected slave (read TI-545). In this state, the TI-545 Vmemory is read (the number of words specified in "#sent" starting at "Vsrce") and loaded into the respective slave xmit buffer (addresses 4000H to 7c00H). This is the data that will be sent to the respective slave node.

```
0:if (F1 == 1)      /* read state? */
1:  {
2:    if (F2 == 0)    /* read from 545 not yet initiated? */
3:    {
4:      B86 = &B182;    /* initiate pointer to task code read buffer */
5:      W88 = 1a00H;    /* initiate pointer to TI545 buffer */
6:      if (B62%15 == 0) /* even number of 15 word blocks to read? */
7:        B35 = B62/15; /* yes, calc # of blocks to read */
8:      else
9:        B35 = B62/15 + 1; /* calc # of blocks to read */
10:     W36 = W64-1;    /* initiate Vaddr in 545 to read from */
11:     F2 = 1;        /* read initiated */
12:     B39 = 6;        /* max number of error retries = 6 */
13:   }
14: }
15:
```

F001	(read TI)	read	TI545	state
F002	(rd init)	read	state	initiat
B035	(numblks)	# of	blocks	TI task
B039	(err cnt)	TI task	retry	count
B062	( #send )	# of	words	to send
B086	( pntr1 )	task	buffer	pointer
B182	( rbuff )	tskcode	read	buffer
W036	( Vaddr )	TI545	Vmem	address
W064	( Vsrce )	Vmem	source	address
W088	( pntr2 )	network	buffer	pointer



## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 3 - High-level

```
0:if (F1 == 1)          /* read state? */
1:  {
2:    if (F103 == 0)      /* task code with 545 complete? */
3:    {
4:      if (F102 == 1)    /* task code comm failure with 545? */
5:      {
6:        --B39;          /* dec error count */
7:        if (B39 == 0)   /* error retries count out? */
8:        {
9:          B40 = 11H;    /* comm error = TI_READ_ERROR */
10:         W106 = 1b00H + ((B61-1)<<1);
11:         sfunc08(W106,B40); ++W106; /* save fault code in */
12:         sfunc08(W106,0);          /* TI-505 comm stat stack */
13:         F1 = 0, F2 = 0, F0 = 1;   /* exit read state */
14:       }
15:     }
16:     F103 = 1; /* retry task code comm with 545 */
17:     F102 = 0; /* reset task code comm fail bit */
18:   }
19:   else if (F9 == 1) /* task code comm done? */
20:   {
21:     for (B42 = 0; B42 < 15; ++B42) /* copy block read from 545 */
22:     { /* to network xmit buffer */
23:       sfunc08(W88,*B86); ++B86, ++W88; /* copy nth word */
24:       sfunc08(W88,*B86); ++B86, ++W88;
25:     }
26:     B86 = &B182; /* reset task code read buffer */
27:     F9 = 0; /* block read done */
28:   }
29:   else if (B35 != 0) /* more blocks to read? */
30:   {
31:     B56 = 50H; /* block read task code */
32:     B57 = 4; /* # of bytes to xmit in task code req */
33:     B58 = 32; /* # of bytes to rcve in task code resp */
34:     W150 = W36; /* starting V mem address to read */
35:     W36 = W36 + 15; /* next block address to read */
36:     --B35; /* dec # of blocks to read */
37:     F103 = 1; /* initiate task code comm */
38:     F9 = 1; /* preset block read done */
39:   }
40:   else
41:   {
42:     W88 = 1a00H; /* copy data read from TI545 to */
43:     while (B62 != 0) /* respective slave xmit buffer */
44:     {
45:       sfunc07(W88,B87);
46:       sfunc08(W84,B87); ++W84, ++W88;
47:       sfunc07(W88,B87);
48:       sfunc08(W84,B87); ++W84, ++W88;
49:       --B62;
50:     }
```

- block continued on next page -

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 3 - High-level

- block originated on prev page -

```

51:          F1 = 0, F2 = 0, F4 = 1;    /* enter write state */
52:          W84 = (B61*512)+3c00H+256; /* point to respective rcve buffer */
53:          }
54:      }
55:  }
56:

```

```

F000 (select ) select slave state
F001 (read TI) read TI545 state
F002 (rd init) read state initiat
F004 (writeTI) write TI545 state
F009 (TI done) TI545 task done
F102 (tskfail) TI545 task fail
F103 (TIcomm ) TI545 task in prog
B035 (numblks) # of blocks TI task
B039 (err cnt) TI task retry count
B040 (commerr) comm error code
B042 ( (i) ) loop (i)
B056 (tskcode) TI545 task code
B057 (#xmitt ) task code #xmitt
B058 ( #rcve ) task code #rcve
B061 ( slave ) network slave number
B062 ( #send ) # of words to send
B086 ( pntrl ) task buffer pointer
B087 (tempreg) temp registr
B182 ( rbuff ) tskcode read buffer
W036 ( Vaddr ) TI545 Vmem address
W084 (slveptr) slave buffer pointer
W088 ( pntr2 ) network buffer pointer
W106 (statptr) comm status pointer
W150 (Vstack ) Vmem read stack

```

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 4 - High-level

\*\* Write state \*\*

The next two blocks write the data that was read from the slave to the TI-545 Vmemory starting at "Vdest" (the number of words written is "#rcve").

```

0:if (F4 == 1)          /* write state? */
1:  {
2:    if (F5 == 0)      /* write state not yet initiated? */
3:    {
4:      W88 = 1a00H;    /* copy data to be written to TI545 from */
5:      for (B42 = B63; B42 > 0; --B42) /* respective slave rcve buffer */
6:      {
7:        sfunc07(W84,B87);
8:        sfunc08(W88,B87);
9:        ++W84, ++W88;
10:       sfunc07(W84,B87);
11:       sfunc08(W88,B87);
12:       ++W84, ++W88;
13:     }
14:     B86 = &B152;     /* initiate pointer to task code write buffer */
15:     W88 = 1a00H;     /* initiate pointer to TI545 buffer */
16:     B35 = B63/14;    /* number of blocks to to write */
17:     B43 = B63%14;    /* number of words remaining to write */
18:     W36 = W70-1;     /* initiate Vmem dest address */
19:     F5 = 1;          /* write state initiated */
20:     B39 = 6;         /* maximum # of error retries = 6 */
21:
22:     /* reset TI_READ_ERROR */
23:     W106 = 1b00H + ((B61-1)<<1);
24:     sfunc07(W106,B108);
25:     if (B108 == 11H)
26:       sfunc08(W106,2);
27:   }
28: }
29:

```

F004	(writeTI)	write	TI545	state
F005	(wr init)	write	state	initiat
B035	(numblks)	# of	blocks	TI task
B039	(err cnt)	TI task	retry	count
B042	( (i) )		loop	(i)
B043	(#words)	# of	words	remain
B061	( slave )	network	slave	number
B063	( #rcve )	# of	words	to rcve
B086	( pntrl )	task	buffer	pointer
B087	(tempreg)		temp	registr
B108	(comstat)		comm	status
B152	( xbuff )	tskcode	write	buffer
W036	( Vaddr )	TI545	Vmem	address
W070	( Vdest )	Vmem	dest	address

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 4 - High-level

W084	(slveptr)	slave	buffer	pointer
W088	(pntr2)	network	buffer	pointer
W106	(statptr)	comm	status	pointer

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 5 - High-level

```
0:if (F4 == 1)          /* write state? */
1:  {
2:    if (F103 == 0)      /* task code with TI 545 complete? */
3:    {
4:      if (F102 == 1)    /* task code comm failure with 545? */
5:      {
6:        --B39;          /* dec error count */
7:        if (B39 == 0)   /* maximum number of retries done? */
8:        {
9:          B40 = 12H;    /* comm error = TI_WRITE_ERROR */
10:         W106 = 1b00H + ((B61-1)<<1);
11:         sfunc08(W106,B40); ++W106; /* save fault code in */
12:         sfunc08(W106,0);          /* TI-505 stat stack */
13:         F4 = 0, F5 = 0, F0 = 1; /* exit write state */
14:       }
15:     }
16:     F103 = 1; /* retry task code with 545 */
17:     F102 = 0;
18:   }
19:   else if (B35 != 0) /* more blocks to write? */
20:   {
21:     B56 = 51H; /* block write task code */
22:     B57 = 32; /* number of task code bytes to write */
23:     B58 = 2; /* number of task code bytes to read */
24:     W150 = W36; /* Vmem start address */
25:     W36 = W36 + 14; /* next block start address to write */
26:     --B35; /* dec # of blocks to write */
27:     B86 = &B152; /* point to task code write buffer */
28:     for (B42 = 0; B42 < 14; ++B42)
29:     {
30:       sfunc07(W88,*B86); ++W88, ++B86; /* copy from comm buffer */
31:       sfunc07(W88,*B86); ++W88, ++B86; /* to task code buffer */
32:     }
33:     F103 = 1; /* initiate task code comm with 545 */
34:   }
35:   else if (B43 != 0) /* write remaining words to 545? */
36:   {
37:     B56 = 51H; /* block write task code */
38:     B57 = (B43 << 1) + 4; /* # of bytes to write */
39:     B58 = 2; /* # of bytes to read in task code */
40:     W150 = W36; /* Vmem start address */
41:     B86 = &B152; /* point to task code write buffer */
42:     for (B42 = 0; B42 < B43; ++B42)
43:     {
44:       sfunc07(W88,*B86); /* copy nth word from comm buffer */
45:       ++W88, ++B86; /* to task code write buffer */
46:       sfunc07(W88,*B86);
47:       ++W88, ++B86;
48:     }
49:     B43 = 0; /* reset # remaining */
50:     F103 = 1; /* initiate task code comm */
```

- block continued on next page -

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 5 - High-level

- block originated on prev page -

```

51:      }
52:      else
53:          F4 = 0, F5 = 0, F6 = 1;    /* enter done state */
54:      }
55:  }
56:

```

F000	(select )	select	slave	state
F004	(writeTI)	write	TI545	state
F005	(wr init)	write	state	initiat
F006	(comdone)	TI545	comm	done
F102	(tskfail)	TI545	task	fail
F103	(TIcomm )	TI545	task	in prog
B035	(numblks)	# of	blocks	TI task
B039	(err cnt)	TI task	retry	count
B040	(commerr)	comm	error	code
B042	( (i) )		loop	(i)
B043	(#words )	# of	words	remain
B056	(tskcode)	TI545	task	code
B057	(#xmitt )	task	code	#xmitt
B058	( #rcve )	task	code	#rcve
B061	( slave )	network	slave	number
B086	( pntr1 )	task	buffer	pointer
B152	( xbuff )	tskcode	write	buffer
W036	( Vaddr )	TI545	Vmem	address
W088	( pntr2 )	network	buffer	pointer
W106	(statptr)	comm	status	pointer
W150	(Vstack )	Vmem	read	stack

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 6 - High-level

\*\* Comm network node select state \*\*

This block searches for enabled nodes and initiates the parameters and communications with any enabled nodes.

```
0:if (F3 == 0 && B105 == 0)          /* select comm node? */
1:  {
2:    F8 = 0;                          /* reset done bit */
3:    for (B42 = B96 + 1; B42 < 33 && F8 == 0; ++B42)
4:      {
5:        /* search for enabled nodes and initiate comm */
6:        W92 = (B42*512)+3c00H;        /* point to nth node */
7:        sfunc07(W92,B87);
8:        if (B87 == B42)              /* nth node enabled? */
9:          {
10:           B96 = B42;                /* initiate comm with node */
11:           F8 = 1;
12:          }
13:        else
14:          {
15:           W106 = 1b00H + ((B42-1)<<1);
16:           sfunc08(W106,0); ++W106; /* set stat to "disabled" */
17:           sfunc08(W106,0); /* in TI-505 stat stack is not used */
18:          }
19:      }
20:
21:  if (F8 == 0)                      /* all nodes checked? */
22:    B96 = 1;                        /* yes, start again with first slave */
23:  else
24:    {
25:      W92 = W92 + 2;                /* get parameters for nth node */
26:      sfunc07(W92,B97);             /* # to xmit */
27:      W92 = W92+4;
28:      sfunc07(W92,B66); ++W92; /* sdest */
29:      sfunc07(W92,B67); ++W92;
30:      sfunc07(W92,B98);             /* # to rcve */
31:      W92 = W92+2;
32:      sfunc07(W92,B68); ++W92; /* ssrce */
33:      sfunc07(W92,B69);
34:      W92 = W92+5;                  /* point to nth node xmit buffer */
35:      W94 = 7e0aH;                  /* point to sfunc13 buffer */
36:      for (B42 = 0; B42 < B97; ++B42)
37:        {
38:          /* copy nth node xmit buffer to sfunc13 buffer */
39:          sfunc07(W92,B87);
40:          sfunc08(W94,B87);
41:          ++W92, ++W94;
42:          sfunc07(W92,B87);
43:          sfunc08(W94,B87);
44:          ++W92, ++W94;
45:        }
```

- block continued on next page -

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 6 - High-level

- block originated on prev page -

```
46:      F3 = 1;      /* initiate comm state */
47:      }
48:  }
49:
```

F003	(netcomm)	network	comm	state
F008	( done )			done
B042	( (i) )		loop	(i)
B066	( sdest )	slave	dest	address
B067	( sdest )	slave	dest	address
B068	( ssrce )	slave	source	address
B069	( ssrce )	slave	source	address
B087	(tempreg)		temp	registr
B096	( node )	network	node	number
B097	( #xmit )	# of	words	to xmit
B098	( #rcve )	# of	words	to rcve
B105	(download)	down-	load	mode
W092	(nodeptr)	node	buffer	pointer
W094	( pnt3 )		buffer	pointer
W106	(statptr)	comm	status	pointer



## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 7 - High-level

\*\* network comm state \*\*

This block performs the communications with the nth node, xmitting the previously loaded sfunc13 buffer and storing the data read from the node at the node rcve buffer.

```

0:if (F3 == 1)          /* comm state */
1:  {
2:    W94 = 0;          /* comm with nth node */
3:    B38 = sfunc13(B96,B97,*W94,*W66,B98,*W68,*W94);
4:    if (B38 == 2)      /* comm done? */
5:      {
6:        W92 = (B96*512)+3d00H; /* point to nth node rcve buffer */
7:        W94 = 7e0aH;         /* point to sfunc13 buffer */
8:        while (B98 != 0)     /* copy data rcved to node buffer */
9:          {
10:           sfunc07(W94,B87);
11:           sfunc08(W92,B87);
12:           ++W92, ++W94;
13:           sfunc07(W94,B87);
14:           sfunc08(W92,B87);
15:           ++W92, ++W94;
16:           --B98;
17:          }
18:      }
19:    if (B38 >= 2)      /* comm not busy? */
20:      {
21:        W94 = ((B96-1)<<1) + 1b00H; /* save error code */
22:        sfunc08(W94,B38); ++W94;
23:        sfunc08(W94,0);
24:        if (B38 != 2)    /* comm error? */
25:          B40 = B38;    /* save error code */
26:        W106 = 1b00H + ((B96-1)<<1); /* save comm return value in */
27:        sfunc07(W106,B108); /* test for TI_READ/WRITE error */
28:        if (B108 < 11H)
29:          {
30:            sfunc08(W106,B38); ++W106; /* TI-505 stat stack */
31:            sfunc08(W106,0);
32:          }
33:        F3 = 0;          /* exit comm state */
34:      }
35:  }
36:

```

F003	(netcomm)	network	comm	state
B038	(sf13ret)	sfunc13	return	value
B040	(commerr)	comm	error	code
B087	(tempreg)		temp	registr
B096	( node )	network	node	number
B097	( #xmit )	# of	words	to xmit
B098	( #rcve )	# of	words	to rcve

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

block: 7 - High-level

B108	(comstat)		comm	status
W066	( sdest )	slave	dest	address
W068	( ssrce )	slave	source	address
W092	(nodeptr)	node	buffer	pointer
W094	( pptr3 )		buffer	pointer
W106	(statptr)	comm	status	pointer

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

block: 8 - High-level

```

0:/* test for download mode and wait until complete */
1:if (F6 == 1)
2:  {
3:    if (B105 == 0)          /* not in download mode? */
4:    {
5:      F0 = 1;              /* start next comm cycle with TI */
6:      F6 = 0;
7:    }
8:
9:    /* reset TI_WRITE_ERROR */
10:   W106 = 1b00H + ((B61-1)<<1);
11:   sfunc07(W106,B108);
12:   if (B108 == 12H)
13:     sfunc08(W106,2);
14:   }
15:
16:if (B105 != 0)             /* download mode? */
17:  {
18:    sfunc07(1900H,B90);    /* restore TI-505 comm status stack address */
19:    sfunc07(1901H,B91);
20:  }
21:
22:sfunc08(1ff4H,1);         /* set node address to 1 (master) */
23:

```

F000	(select )	select	slave	state
F006	(comdone)	TI545	comm	done
B061	( slave )	network	slave	number
B090	(Verrstk)	Vmem	error	stack
B091	(Verrstk)	Vmem	error	stack
B105	(dwnload)	down-	load	mode
B108	(comstat)		comm	status
W106	(statptr)	comm	status	pointer

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*

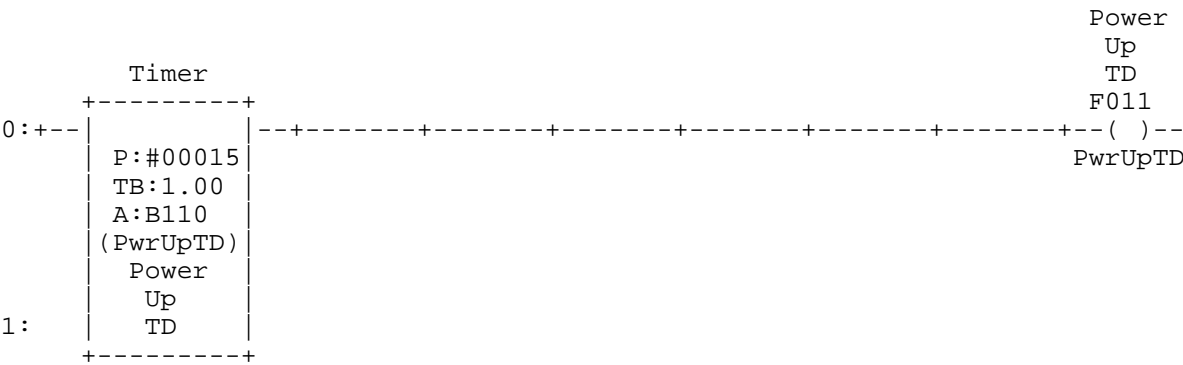
block: 9 - Assembly

```
0:          jnb    F11,com.enb      ; power up TD complete?
1:          jnb    F10,com.enb      ; TI505 reset complete?
2:          anl    h'c0,#h'fb      ; reset TI bus complete latch
3:          orl    h'c0,#h'4
4:          jnb    P3.3,com.enb     ; TI505 still resetting?
5:          clr    IT1              ; level trigger on TI bus intrpt
6:          setb   EX1              ; enable TI505 bus intrpt
7:          setb   F10              ; TI505 reset complete
8:          mov    B105,#d'0        ; enable task code an network comm
9:          setb   F000              ; enable task code comm
10:;
11:com.enb:
12:;
13:
```

```
F000 (select )    select  slave  state
B105 (download)   down-   load   mode
```

APPENDIX A  
MCOM505 PROGRAM PRINTOUT

\*\*\*\*\*  
block: 10 - Ladder



## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

Typical Cross reference format for variable:

file:block-line(usage)

where: file = file where variable is referenced.  
 block = block number in file.  
 line = line number in block.  
 usage = sense that variable is referenced.

file key: INI = initialization file.  
 MNF = main program file.  
 TMD = Timed interrupt file.  
 COM = co-cpu com interrupt file.  
 Uxx = User function file (xx = ufunc num).

Usage key: \* = variable value altered at referenced location.  
 (output coil, timer accumulator, set equal, etc.)

+ = variable used in Assembly block.  
 (usage in Assembly block unknown)

@ = variable used as pointer in High-level block.

if \*, +, or @ is not associated with variable,  
 the variable is not altered at location.  
 (contact, timer preset, tested for value, etc.)

Addr (nickname)	description			location used			
F000 (select )	select	slave	state	MNF: 1-0 MNF: 8-5*	1-45* 9-9+	3-13*	5-13*
F001 (read TI)	read	TI545	state	MNF: 1-42* MNF: 3-51*	2-0	3-0	3-13*
F002 (rd init)	read	state	initiat	MNF: 2-2	2-11*	3-13*	3-51*
F003 (netcomm)	network	comm	state	MNF: 6-0	6-46*	7-0	7-33*
F004 (writeTI)	write	TI545	state	MNF: 1-39* MNF: 5-13*	3-51* 5-53*	4-0	5-0
F005 (wr init)	write	state	initiat	MNF: 4-2	4-19*	5-13*	5-53*
F006 (comdone)	TI545	comm	done	MNF: 5-53*	8-1	8-6*	
F008 ( done )			done	MNF: 1-2* MNF: 6-2*	1-3 6-3	1-26* 6-11*	1-30 6-21
F009 (TI done)	TI545	task	done	MNF: 3-19	3-27*	3-38*	
F010 (TIreset)	TI505	reset	comp	INI: 1-5*			

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

Addr (nickname)		description			location used			
F011	(PwrUpTD)	Power	Up	TD	MNF:10-0*			
F102	(tskfail)	TI545	task	fail	MNF: 3-4	3-17*	5-4	5-17*
F103	(TIcomm )	TI545	task	in prog	MNF: 3-2	3-16*	3-37*	5-2
					MNF: 5-16*	5-33*	5-50*	
B035	(numblks)	# of	blocks	TI task	MNF: 2-7*	2-9*	3-29	3-36*
					MNF: 4-16*	5-19	5-26*	
B038	(sf13ret)	sfunc13	return	value	MNF: 7-3*	7-4	7-19	7-22
					MNF: 7-24	7-25	7-30	
B039	(err cnt)	TI task	retry	count	MNF: 2-12*	3-6*	3-7	4-20*
					MNF: 5-6*	5-7		
B040	(commerr)	comm	error	code	MNF: 3-9*	3-11	5-9*	5-11
					MNF: 7-25*			
B042	( (i) )		loop	(i)	MNF: 1-3*	1-3	1-6	1-8
					MNF: 1-25	1-3*	3-21*	3-21
					MNF: 3-21*	4-5*	4-5	4-5*
					MNF: 5-28*	5-28	5-28*	5-42*
					MNF: 5-42	5-42*	6-3*	6-3
					MNF: 6-6	6-8	6-10	6-15
					MNF: 6-3*	6-36*	6-36	6-36*
B043	(#words )	# of	words	remain	MNF: 4-17*	5-35	5-38	5-42
					MNF: 5-49*			
B056	(tskcode)	TI545	task	code	MNF: 3-31*	5-21*	5-37*	
B057	(#xmitt )	task	code	#xmitt	MNF: 3-32*	5-22*	5-38*	
B058	( #rcve )	task	code	#rcve	MNF: 3-33*	5-23*	5-39*	
B059	(TIWait )	TI Task	Code	Wait	INI: 1-7*			
B061	( slave )	network	slave	number	MNF: 1-3	1-25*	1-32*	3-10
					MNF: 3-52	4-23	5-10	8-10
B062	( #send )	# of	words	to send	MNF: 1-11*	2-6	2-7	2-9
					MNF: 3-43	3-49*		
B063	( #rcve )	# of	words	to rcve	MNF: 1-17*	1-33*	4-5	4-16
					MNF: 4-17			
B064	( Vsrce )	Vmem	source	address	MNF: 1-13*			
B065	( Vsrce )	Vmem	source	address	MNF: 1-14*			
B066	( sdest )	slave	dest	address	MNF: 1-15*	6-28*		

## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

Addr (nickname)	description			location used			
B067 ( sdest )	slave	dest	address	MNF: 1-16*	6-29*		
B068 ( ssrce )	slave	source	address	MNF: 1-19*	6-32*		
B069 ( ssrce )	slave	source	address	MNF: 1-20*	6-33*		
B070 ( Vdest )	Vmem	dest	address	MNF: 1-21*			
B071 ( Vdest )	Vmem	dest	address	MNF: 1-22*			
B086 ( pntrl )	task	buffer	pointer	MNF: 2-4* MNF: 3-24* MNF: 5-30@ MNF: 5-41* MNF: 5-47*	3-23@ 3-26* 5-30* 5-44@ 5-45*	3-23* 4-14* 5-31@ 5-45* 5-46@	3-24@ 5-27* 5-31* 5-46@
B087 ( tempreg )		temp	registr	MNF: 1-7* MNF: 3-45* MNF: 4-7* MNF: 6-7* MNF: 6-42* MNF: 7-13*	1-8 3-46 4-8 6-8 6-43 7-14	1-36* 3-47* 4-10* 6-39* 7-10* 7-11	1-38 3-48 4-11 6-40 7-11
B087.0 ( tregbt0 )	temp	reg	bit0	MNF: 1-37*	1-37		
B090 ( Verrstk )	Vmem	error	stack	INI: 1-0* MNF: 8-18*			
B091 ( Verrstk )	Vmem	error	stack	INI: 1-1* MNF: 8-19*			
B096 ( node )	network	node	number	MNF: 6-3 MNF: 7-6	6-10* 7-21	6-22* 7-26	7-3
B097 ( #xmit )	# of	words	to xmit	MNF: 6-26*	6-36	7-3	
B098 ( #rcve )	# of	words	to rcve	MNF: 6-30*	7-3	7-8	7-16*
B105 ( dwnload )	down-	load	mode	INI: 1-4* MNF: 6-0	8-3	8-16	9-8+
B108 ( comstat )		comm	status	MNF: 4-24* MNF: 8-11*	4-25 8-12	7-27*	7-28
B110 ( PwrUpTD )	Power	Up	TD	MNF: 10-0*			
B152 ( xbuff )	tskcode	write	buffer	MNF: 4-14	5-27	5-41	
B182 ( rbuff )	tskcode	read	buffer	MNF: 2-4	3-26		
W036 ( Vaddr )	TI545	Vmem	address	MNF: 2-10* MNF: 4-18* MNF: 5-40	3-34 5-24	3-35* 5-25*	3-35 5-25



## APPENDIX A

### MCOM505 PROGRAM PRINTOUT

Addr (nickname)	description			location used			
W064 ( Vsrce )	Vmem	source	address	MNF: 2-10			
W066 ( sdest )	slave	dest	address	MNF: 7-3@			
W068 ( ssrce )	slave	source	address	MNF: 7-3@			
W070 ( Vdest )	Vmem	dest	address	MNF: 1-34* 4-18			
W084 (slveptr)	slave	buffer	pointer	MNF: 1-6*	1-7	1-10*	1-10
				MNF: 1-11	1-12*	1-12	1-13
				MNF: 1-13*	1-14	1-14*	1-15
				MNF: 1-15*	1-16	1-16*	1-17
				MNF: 1-17*	1-18*	1-19	1-19*
				MNF: 1-20	1-20*	1-21	1-21*
				MNF: 1-22	1-24*	1-24	1-35*
				MNF: 3-46	3-46*	3-48	3-48*
				MNF: 3-52*	4-7	4-9*	4-10
W088 ( pntr2 )	network	buffer	pointer	MNF: 2-5*	3-23	3-23*	3-24
				MNF: 3-24*	3-42*	3-45	3-46*
				MNF: 3-47	3-48*	4-4*	4-8
				MNF: 4-9*	4-11	4-12*	4-15*
				MNF: 5-30	5-30*	5-31	5-31*
				MNF: 5-44	5-45*	5-46	5-47*
W090 (Verrstk)	Vmem	error	stack	MNF: 1-34			
W092 (nodeptr)	node	buffer	pointer	MNF: 6-6*	6-7	6-25*	6-25
				MNF: 6-26	6-27*	6-27	6-28
				MNF: 6-28*	6-29	6-29*	6-30
				MNF: 6-31*	6-31	6-32	6-32*
				MNF: 6-33	6-34*	6-34	6-39
				MNF: 6-41*	6-42	6-44*	7-6*
				MNF: 7-11	7-12*	7-14	7-15*
W094 ( pntr3 )		buffer	pointer	MNF: 6-35*	6-40	6-41*	6-43
				MNF: 6-44*	7-2*	7-3@	7-3@
				MNF: 7-7*	7-10	7-12*	7-13
				MNF: 7-15*	7-21*	7-22	7-22*
				MNF: 7-23			
W106 (statpctr)	comm	status	pointer	MNF: 3-10*	3-11	3-11*	3-12
				MNF: 4-23*	4-24	4-26	5-10*
				MNF: 5-11	5-11*	5-12	6-15*
				MNF: 6-16	6-16*	6-17	7-26*
				MNF: 7-27	7-30	7-30*	7-31
				MNF: 8-10*	8-11	8-13	
W150 (Vstack )	Vmem	read	stack	MNF: 3-34* 5-24* 5-40*			

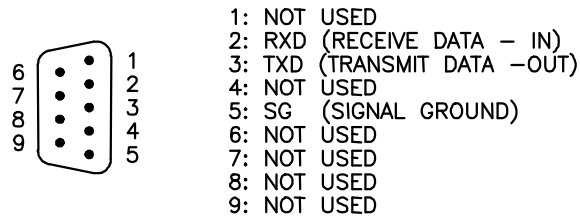
## **APPENDIX A**

### **MCOM505 PROGRAM PRINTOUT**

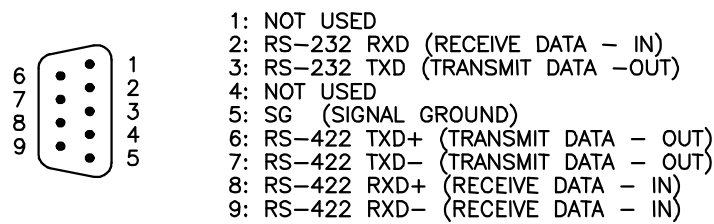
*(This Page Intentionally Left Blank)*

## APPENDIX B

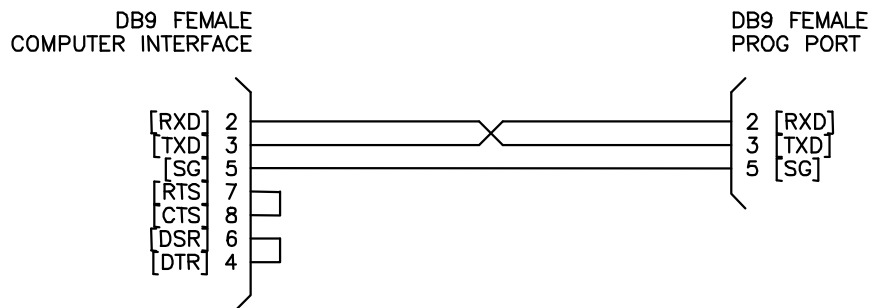
### RS-232/RS-422 PIN OUTS/ CABLES



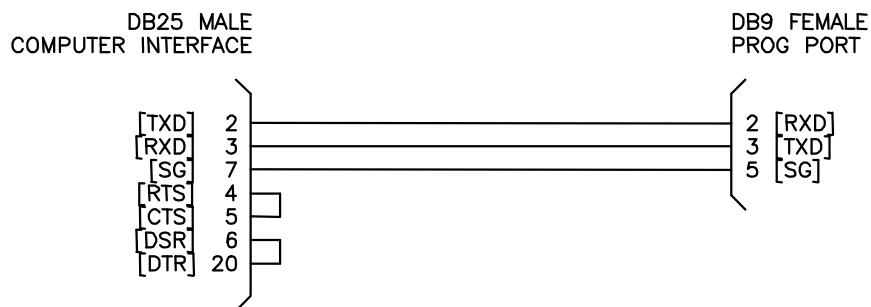
#### PROG Port Pin Out



#### USER Port Pin Out



#### DB9 (com1) to PROG Port Cable



#### DB25 (com1) to PROG Port Cable