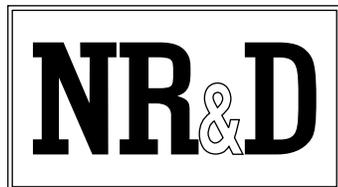


CAM500

Installation and Programming Manual

This manual describes the CAM500 SY/MAX[®] compatible memory module, its uses and setup. It also describes the use of the CAM500SW programming software.

Effective: 27 February 1996



Niobrara Research and Development Corporation
P.O. Box 3418 Joplin, MO 64803

Telephone: (800) 235-6723 or (417) 624-8918
Facsimile: (417) 624-8920
BBS: (417) 624-2028
Internet: <http://niobrara.com>

® SY/MAX and SY/NET are registered trademarks of Square D Company.

® MS/DOS, MS, and Microsoft are trademarks of Microsoft Corporation

© Niobrara Research & Development Corporation 1994, 1995, 1996. All Rights Reserved.

Contents

1 Introduction	7
2 Installation	11
3 Configuration and Initialization	15
Configuration Planning	15
Counting data storage registers	15
Planning file windows	15
Counting rack interface registers	16
Planning file sizes	16
Processing speed	16
Loading the Configuration	16
Manual configuration procedure	17
Access through the front panel port.....	17
Arbitration between Multiple Accessors	18
Rack Interface Data Direction Control	18
4 File Operations	21
Storage Operations	23
Store By Record Number	23
Store by Key	23
Store by Next Record Number	24
Retrieval Operations	24
Retrieve by Record Number	24
Retrieve by Key	25
Retrieve by Next Record Number	25
Retrieve by Previous Record Number	26
Multiple Record Sequential Retrieval	26
.....	26
Delete Operations	27
Delete by Record Number	27
Delete by Key	27
Delete All	27

5 Examples	29
Sequential Storage and Retrieval	29
Random Storage and Retrieval	31
Keyed Storage and Retrieval	32
Sequential Retrieval of Random or Keyed Storage	32
Read Only Applications	32
6 Tips & Tricks	33
7 CAM500SW	35
Main Menu	35
SETUP	36
Personal Computer COM: port	36
SY/LINK Connection	37
Command Line Parameters	38
Configuration Menu	38
Fetch configuration from module	38
Edit Configuration in PC memory	38
Print configuration data, Print Map of rack interface	39
Load configuration from disk to PC memory	39
Write from PC memory to disk	39
Send configuration to CAM500 module	39
Delete configuration file	40
Transfer Menu	40
Loading computer data into a module	40
Unloading module data to the computer	41
Converting the record format of module data	41
Deleting data files	42
Utilities Menu	42
Viewing and Editing module registers	42
Erase CAM500 file	42
File record length conversion	42
Backup Menu	43
Protect Menu	43
8 Local Registers	45
9 Connector Pinouts	49
RS-422 Port (DE9S with slide lock posts)	49
Glossary	51
Figures	
Figure 1-1 CAM500 Front Panel	9

Figure 2-1 CAM500SW Startup Screen	12
Figure 2-2 Utility Menu	12
Figure 2-3 View Module Registers Screen.....	13
Figure 4-1 Example Rack Interface Map	22
Figure 4-2 Retrieve by Key Ladder Example	22
Figure 5-1 Ladder Example	30
Figure 5-2 Ladder Example	31
Figure 7-1 Main Menu	36
Figure 7-2 Setup Screen	38
Figure 7-3 Configuration Sub-menu	39
Figure 7-4 Edit Configuration screen	39
Figure 7-5 Transfer Menu	40
Figure 7-6 Binary and Hex DOS files	41
Figure 7-7 Utility Menu	42
Figure 7-8 Register viewer screen	43
Figure 7-9 Erase Module Screen	43

Tables

Table 3-1 PLC I/O Operations Overview	19
Table 8-1 CAM500 Register Map	45
Table 8-2 File Register Block	46
Table 8-3 Window Register Block	46
Table 8-4 Baud Rate Values	47
Table 8-5 Command and Status Bit Listing	47

Introduction

The Niobrara CAM500 is a SY/MAX register rack compatible memory module. The CAM500 features 520,192 nonvolatile registers which can be stored, retrieved or deleted through the register backplane and/or through a SY/MAX compatible port on the front of the module. The CAM500 performs data storage, searching, and retrieval with minimum ladder programming. The module performs arbitration between multiple external accessors. By relieving the ladder programmer of these complex problems, the CAM500 greatly simplifies building robust applications involving data storage and retrieval.

Store and Retrieve:

- Pick Lists
- Routing Tables
- Recipes
- Tables of Correspondence

Data Logging:

- Alarms
- Events
- Statistics
- Reports

Registers are grouped together into records. The number of registers in a group is user defined and called the record length. The CAM500 is designed for storing and retrieving data by record number and/or by key. The key is a user-defined identifier stored as the first part of the record that must be unique for each record.

Records are grouped together inside the CAM500 into files and files are accessed through register rack windows. Multiple windows can exist for each file and multiple files can exist on the CAM500 for storing and retrieving records of different lengths and/or key structures in the same module. The total number of records the CAM500 can store is determined by the number of files and the number and length of the records. The rack space required for the CAM500 is determined by the number of files, the number of windows for each file, and the length of the records.

The CAM500 storage area is flexible! Records can be accessed by record number or key, record lengths are user defined, records are grouped into files, multiple files can exist in the CAM500, multiple windows can exist for each file, multiple processes or areas of ladder code can access the same data

simultaneously.

The CAM500 is rack addressed like other register rack modules. The minimum register rack address space is 9 registers plus the user defined record length per file. The maximum available register rack address space in the CAM500 is 2,048 registers. The baud rate of the front port is set in a rack register from the processor or using the CAM500SW configuration software that is supplied with the unit.

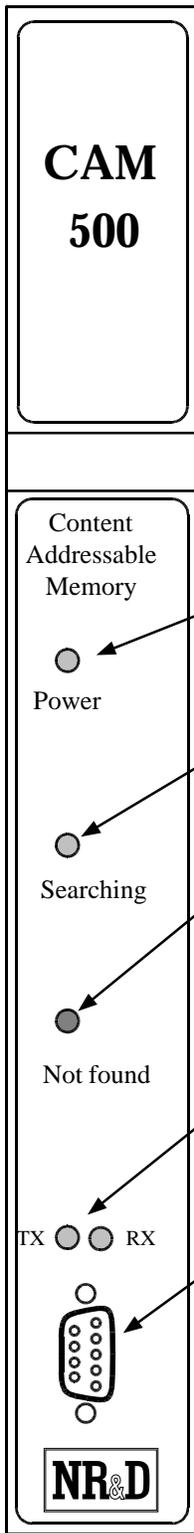
A file is set up in the register rack address space and consists of a record length register, the number of windows register, the number of records register, the length of the key register, a command register, a status register, the record number register and the registers where the record appears. See the CAM500 Rack Image Description below for the order of the registers and the valid values each register can have.

Multiple file windows can be opened into the CAM500 allowing an operator interface, computer, another PLC or other input device to update data through one file window while the ladder program accesses records using another file window. The ladder program can also have more than one file window open into a file in the CAM500.

The average time required to retrieve a record in the CAM500 is 0.4 milliseconds when the record has a 3 register key and a record length of 8 registers, and the CAM500 is storing 30,000 of these type records. The CAM500 uses a hashed search and retrieval time is optimized if less than 75% of the available records are in use.

Additional information on the registers within the module is available in Chapter 8.

Figure 1-1 shows the front panel of the CAM500 module.



Status Indicators (LEDs)

Power (Green) - This light is on when the unit has power and is running.

Searching (Green) - When on, the CAM 500 is performing a file operation.

Not found (Red) - When on for two seconds, the CAM 500 has attempted to search for a record not in the file. When on steady, indicates that the module has not been properly configured.

TX, RX (Yellow) - These lights flash on when the port is transmitting (TX) or receiving (RX) data.

Communication port

RS-422 Serial Differential Port for connecting to programming and peripheral equipment. Provides +5VDC for powering SC406 Smart Cable. Does not support RTS/CTS handshaking.

Figure 1-1 CAM500 Front Panel

Installation

- 1 Mount the CAM500 module in an available slot in the register rack.
- 2 For ease of configuration, the CAM500 may be connected to an MS/DOS® personal computer COM: port. Niobrara's SC406 RS-232 to RS-422 converter cable can be used to make this connection. The if the SC406 is connected to directly to the CAM500, the external power supply for the SC406 need not be used as the port provides power for the smart cable.
- 3 Apply power to the rack. The green "Searching" light and the red "Not Found" light will flash briefly. The both lights should extinguish. If the red light stays illuminated, this indicates that the module needs to be configured.
- 4 The module must be rack addressed for the PLC to access its registers. Using Square D programming software, allocate rack address register space to the module. You may allocate as few as 9 or as many as 2048 registers depending on the needs of the application.

Note: The scan time of the processor is directly related to the number of externally addressed registers. Allocating more registers than needed can have an adverse impact on the speed of the system. See the manual for the processor for more information on optimizing scan speed.

- 5 Install the CAM500 configuration software. A directory name of "CAM500" is suggested. The installation may be accomplished by using the INSTALL.EXE program included with the distribution diskette. Simply insert the diskette into a floppy drive, and run INSTALL from that floppy drive. Select CAM500SW from the list of softwares to install and allow INSTALL to copy and expand the files to your hard drive.

Run the CAM500 software with the commands:

```
C:  
CD \CAM500  
CAM500SW
```

The screen should look like Figure 2-1.

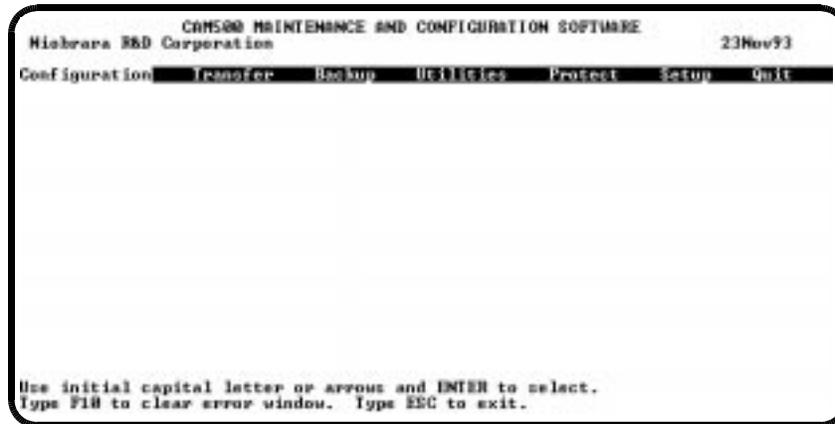


Figure 2-1 CAM500SW Startup Screen

- 6 Press the **S** key for Setup. A menu should appear as in Figure <Cam500 setup screen>. Adjust the settings to match your system. Pressing **Enter** will accept the present value for each item. Pressing the **ESC** key will abort all changes in the setup. For a direct connection from the personal computer to the CAM500, the default settings should be: 9600 baud, Route = NONE, and Front Panel Connection. For more information about CAM500SW see Chapter 7.
- 7 To check the communication link between the CAM500 and the personal computer, select the Utilities menu from the main menu. A sub-menu will appear such as in Figure 2-2.

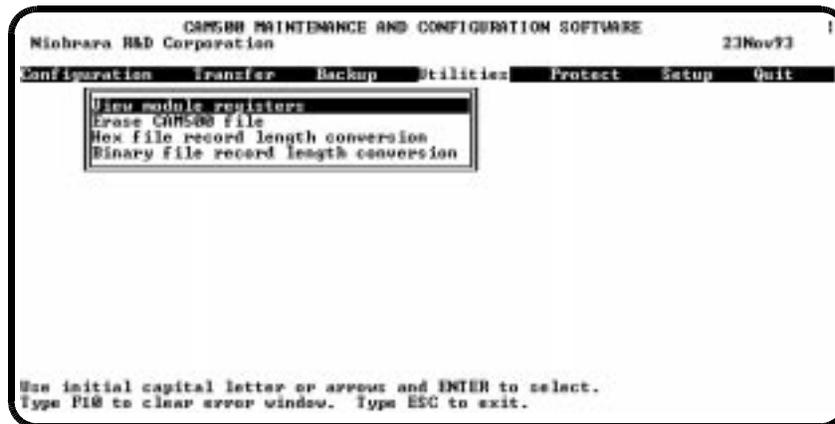


Figure 2-2 Utility Menu

- 8 Press **V** for View Module Registers. This will cause a new screen to appear to display a group of 20 registers within the Module similar to Figure 2-3. The left column is the register number. The second column is the data of that register in hexadecimal format. The third column is the data of that register in unsigned decimal format. The fourth column is the data of that register in signed decimal format. The last column is the status field for that register. A status of E000 indicates that register is currently a PLC input. A status of A000 indicates that register is currently a PLC output. If an error window appears, check the cabling and setup parameters of CAM500SW

Once a good connection to the CAM500 is obtained, the module is ready to be configured for operation.. It is highly recommended to finish reading this manual before configuring the module.

```

CAM500 MAINTENANCE AND CONFIGURATION SOFTWARE
Nichears R&D Corporation                               23Nov93

REGSTR  HEX  UNSIGN  SIGNED  STAT
1 0001      1      1  E000
2 0000     13     13  A000
3 0001      1      1  E000
4 0002      2      2  E000
5 03E8    1000    1000  E000
6 0001      1      1  E000
7 0000      0      0  E000
8 0000      0      0  A000
9 0000      0      0  A000
10 0000      0      0  A000
11 0000      0      0  E000
12 0000      0      0  A000
13 0000      0      0  A000
14 0000      0      0  A000
15 0000      0      0  A000
16 0000      0      0  A000
17 0000      0      0  A000
18 0000      0      0  A000
19 0000      0      0  A000
20 0000      0      0  A000

CAM500 Sys/Max Register Viewer
Up and Down arrows to select register.
Page Up and Page Down to change by 10.
Left and Right arrows to select node.
0..9, A..F to enter new value.
Enter to update without moving.
P10 to acknowledge error.
Escape to exit.

```

Figure 2-3 View Module Registers Screen

Configuration and Initialization

Configuration Planning

Planning the optimum allocation of CAM500 resources is essential to application success. CAM500 resources include data storage registers, rack interface registers, file and record size limits, and processing speed. Read this manual all the way through before attempting to plan the implementation of the CAM500 in your application.

Counting data storage registers

The number of CAM500 data storage registers required by your application is the total of the number consumed by all of the files. The number of data storage registers consumed by a file may be computed as follows:

$$\text{Number of Registers} = (\text{Maximum Record Number} + 1) \times (\text{Record Size} + 1)$$

The additional register per slot is used internally by the CAM500 to track whether a slot is in use or empty and in the key searching process. The total number of data storage registers needed by all files must not exceed 523,115. If you have excess registers, allocate extra records to files that will be accessed by key. The hashing algorithms used by the CAM500 generally work faster with more free slots available.

Planning file windows

A file window should be allocated for each concurrent process in the application system. In addition, the fact that no window may be used from both the front port and the rack may impose the requirement of an additional window for some files. If it is desired to use the Load, Unload, or Backup functions of CAM500SW through the front panel port, one window per file must be available. This may be the same window that is used for other external computer access provided that the two operations do not need to proceed concurrently. Load, Unload, and Backup operations performed through the PLC port do not require an extra window except to allow other operations to proceed concurrently. The CAM500SW backup program uses the last window (highest register address) allocated for each file. In order for Backup to read the configuration through the PLC, the file definition registers must accurately reflect the configuration of the module. If the CAM500 was configured through the PLC port, these values were written at that time. If the configuration is locked, these registers will automatically reflect the correct values.

Counting rack interface registers

The number of rack interface registers required is two (for the Number of Files and Baud Rate registers) plus the total required by all files. The number required for a file is:

$$\text{Registers Required} = 4 + \text{Number of Windows} \times (3 + \text{Record Size})$$

Extra rack interface registers beyond the total used for the first two and all files are initially configured as outputs and may be left out of the rack address map or be used in conjunction with the CAM500 front panel port for auxiliary communication between the PLC and an external device such as an operator interface or display. Interface registers used solely for communication between the CAM500 and an external device through the front panel port need not appear in the PLC's rack address table. The less registers that are included in the rack address table, the faster will be the PLC scan time. If the Backup function of CAM500SW will be used through the PLC port, all defined windows must be accessible through the rack address table. CAM500SW can print a report from a given CAM500 configuration listing the PLC rack addresses of the various CAM500 registers and their function.

Planning file sizes

The first criterion for file size (Maximum Record Number) is to allocate enough space to hold all of your data. You may wish to allocate additional slots to allow for growth of your application (without the need to reconfigure the module) or to improve the response time of keyed files. A more subtle effect is caused by the fact that the CAM500 modifies its key searching algorithms based on file and record size. If performance is critical, experiment with different file sizes to determine which gives optimum performance with your application. In general the more slots (with a given number of records) the quicker they can be searched but this is not an absolute rule. In particular, a file with Maximum Record Number of 65,516 or less uses a completely different algorithm than a larger one and can usually be searched faster.

If 65,536 slots is not enough records for your application, consider breaking your data up into two or more files and using logic within the ladder program to predetermine which file to access. Be sure the same logic is applied for both storage and retrieval!

Processing speed

The CAM500's internal processor is very fast but some applications may tax the available throughput. If more than one window has a command in process at a time, the processor must be shared between the two windows. In other words, it takes just as long to perform two searches simultaneously through two windows as it would to search through one window twice. In fact, it may take slightly longer due to the overhead of the arbitration process. The first window of each file is higher priority than the others and the first file in the module has the highest priority. If multiple files and/or windows are used, assign the lower numbered ones to the most time critical accessors. It is theoretically possible (but highly unlikely) to keep a high priority window so busy with successive operations that lower priority windows never get any processing at all. Response time also depends on the total number of windows configured, each additional window marginally slows operation. If a time critical application uses multiple files, consider using multiple CAM500 modules and putting just one file in a module. Storage, retrieval, and deletion by record number are very fast, always occurring in a millisecond or less. Key search operations and Next and Previous Record Number operations can vary in speed from a fraction of a millisecond to several seconds. The worst case search time depends on the Maximum Record Number and (for a typical file size) is approximately:

$$\text{Worst Case Search (seconds)} = \text{Maximum Record Number} / 40,000$$

Worst case search times will almost never be experienced. File operation responsiveness can also be adversely affected by communications traffic through the front panel port.

Loading the Configuration

The simplest way to configure the module is by using CAM500SW on an MS/DOS computer attached to the front panel port. If the CAM500 software is used, you may skip this section and refer to the section titled CAM500SW.

The division of the memory of the CAM500 into files and of files into slots and of records into registers is defined by invoking the Configure Module operation by setting bit 16 in the Number of Files register and then writing (through the rack or the serial port) configuration information. The following sequence of operations should be used to configure the module. The register modifications listed may be made through the front panel port with CAM500SW or other software, through the rack by the ladder program, or through the rack interface via a PLC communications port. In the following discussion, a module register number is the number of the interface register as seen through the port. To calculate the rack address of a module register, add the register number of the first register assigned to the module in the PLC's rack address table to the module register number and subtract one.

For example, if the first address of the module in the PLC rack address is 3,065 then the address register of the status bits of the first window of the first file is 3,071 (3065+7-1).

Manual configuration procedure

- 1 If the Configuration Lock (module register 8192, bit 1) has previously been set through the front panel port, reset it. CAM500SW may be used to accomplish this.
- 2 If bit 16 of module register 1 is set, clear it now.
- 3 Write the Number of Files to module register 1.
- 4 Set bit 16 in module register 1. This will cause the module to configure all registers except for the status register of the first window of the first file (module register 8) as PLC outputs allowing you to write configuration information.
- 5 Wait for bit 16 of module register 7 to be set indicating the module is ready to accept configuration information.
- 6 For each file, write the values for Record Length, Number of Windows, Maximum Record Number, and Key Length to the appropriate module registers. The record length and number of windows for each file determines the module register address for the subsequent file. Count carefully, writing a configuration value to the wrong register will cause the configuration to be invalid. Write zero to the Command Register of all windows.
- 7 Clear bit 16 of module register 1. This will instruct the module to configure itself and erase any old data records. The green "Searching" light will be illuminated during this process.
- 8 Wait for bit 16 of module register 7 to clear. This will indicate completion of the configuration. If the red "Not Found" light is glowing at the end of the configuration procedure, one or more of the configuration parameters was out of range or written to the wrong register. An attempt to assign more than 523,115 total slots or defining a file structure that requires more than 2048 registers in the rack interface will also result in a red light. Remember that each slot consumes one register in addition to the Record Length.
- 9 If using the serial port, you may wish to set the configuration lock, bit 1 of module register 8192. This will prevent the PLC from inadvertently entering configuration mode.

Access through the front panel port

The front panel port is a SY/MAX compatible communications port that accepts SY/MAX Read and Write register messages from SY/MAX PLCs, computers, or other devices. Register numbers 1 through 2048 accessed through the front panel port are accesses to the rack interface registers. Register number 8192 is accessible only through the front panel port. Setting bit 1 of this register locks the module so that it will not respond to configuration commands until this bit is again reset. This feature can be used to help prevent inadvertent erasure of all data in the case of a ladder programming error. Setting bit 2 of register 8192 write protects the module. No Store or Delete operations can be performed on a write protected module.

Read accesses through the front panel port have no effect on ongoing operations and communication between the PLC rack bus and the CAM500 module. In order to arbitrate write accesses between the PLC bus and the front panel port, any register write performed through the port automatically converts the affected register to a PLC input to prevent contention over which accessor controls the register. In the case of the Record Number and Record Image registers within a window, they will be changed back to PLC outputs at the end of a file operation. In the case of window command registers, file definition registers and the Number of Files and Baud Rate registers, only cycling rack power or reconfiguring the module will return these registers to PLC control after a front panel port write access. Avoid using the front panel port to write to windows which are intended for PLC access.

It is recommended that the CAM500SW program included with the CAM500 be used for all configuration operations. All module features including file operations are possible through the front panel port at all times. PLC reconfiguration of the module or control of an individual window may be locked out by front panel port access.

None of the above applies to access performed through a communications port on the PLC itself. These accesses are completely equivalent to and may be safely interleaved with ladder program generated register rack accesses.

Arbitration between Multiple Accessors

The use of multiple windows for a file allows multiple devices to request file operations concurrently. Typically, one window is reserved for each device that will access the file. Sometimes it may be desirable to have more than one window reserved for the use of a single device. For example, the PLC might have one window open to a file storing records in response to a scanner initiated process and another retrieving them on demand from an operator interface. The use of one window for each process relieves the PLC program from the necessity of synchronizing these two sources of requests. Another possibility is when sequential operations are being carried out in one window while other operations are performed in a second window. The use of a separate window for the sequential access process allows the CAM500 to separately maintain the context (i.e. the next record number to store to or retrieve from) of the windows and relieves the PLC from having to juggle the storage and retrieval record numbers. The CAM500 arbitrates access between all windows whether accessed from the ladder program, a port on the PLC, or the front panel port of the module.

Rack Interface Data Direction Control

When the PLC accesses the CAM500 through the rack, the CAM500 controls whether the registers are inputs or outputs. The Number of Files and Baud Rate registers are always outputs from the PLC. Likewise, the file definition registers Maximum Record Number, Number of Windows, Record Length, and Key Length for each file are always outputs from the PLC (unless changed to inputs by CAM500SW). In each window, the Command Register is a PLC output and the Status Register is a PLC input. The Record Number and Record Image registers alternate between being PLC inputs and outputs depending on the flow of the application. When the operation complete status bits are zero, the CAM500 sets the Record Number and Record Image registers to be PLC outputs so the PLC can write the parameters required for the next operation to be performed. When operations such as retrievals return data to the PLC, these registers may be converted into PLC inputs before the operation complete bit is set. In these operations, the PLC should retrieve the required data from the registers and then clear the command register to inform the CAM500 that it is safe to change these registers back to PLC outputs. When the CAM500 has reset the registers, it informs the PLC of its readiness to accept a new request by clearing the operation complete status bits. The following table outlines which operations accept PLC output data and return PLC input data.

Table 3-1 PLC I/O Operations Overview

Operation	Outputs to CAM500 (when result bit set)	Inputs from CAM500 (when result bit set)
Store by Key	Record Image	Record Number (where stored)
Store by Record Number	Record Number Record Image	None
Store by Next Record Number	Record Number (to start search) Record Image	Record Number (next to try)
Retrieve by Key	Record Image (only Key used)	Record Number (where found) Record Image
Retrieve by Next Record Number	Record Number	Record Image
Retrieve by Previous Record Number	Record Number	Record Number (where found) Record Image
Multiple Record Sequential Retrieval	Record Number	Multiple Record Numbers Multiple Record Images
Delete by Key	Record Image (only Key used)	Record Number (where found) Record Image
Delete by Record Number	Record Number	Record Number (where found)
Delete All	None	None
Configure Module	Number of Files Record Length (for each file) Maximum Record Number (for each file) Number of Windows (for each file) Key Length (for each file)	None

File Operations

Each operation that the CAM500 can perform has associated with it a command bit in the command register and a completion bit in the status register. There is one pair of these registers in each window configured in the module. Any number of windows may have operations in process at any time on the same or different files affecting the same or different records. The effective order in which operations is performed is not predictable, therefore if one window is asked to retrieve a record at the same time another window is asked to delete it, the retrieval may or may not be successful. The delete will always be successful (provided the record exists) because whether or not the retrieve is first, the record will still be present for the delete.

To request a file operation, perform the following items:

- 1 First ensure that the associated completion bit is zero. The CAM500 will clear this bit when the command register is set to zero at the end of the preceding operation. When the completion bit is zero, the command, record number, and record image registers are configured as PLC outputs (to the CAM500).
- 2 Write the parameters required by the operation to the record number and or record image registers.
- 3 Set the appropriate command bit in the command register. The operation will commence and when it has completed any results to be delivered to the PLC will be placed in the record number or record image registers which will be configured as PLC inputs.
- 4 Copy the result data from the interface registers to another memory location within the PLC for use in the ladder logic.
- 5 Clear the command bit. The interface registers will change back to outputs in preparation for the next operation and the completion bit will be reset. You should not write a new command to a given window until the CAM500 has cleared the completion bit from the previous command.

The ladder program fragment in Figure 4-2 illustrates invoking a CAM500 file operation and detecting its completion. For purposes of the example, assume that the CAM500 is rack addressed at PLC register 300 and that File 1 has a record length of 5 registers of which 3 are the key. This will place the File 1, Window 1 Command Register at PLC register 313, the Status Register at PLC register 306, and the Record Image at PLC register 308. See Figure 4-1

CAM500 Rack Interface Map		
=====		
Function	PLC Address	CAM500 Address

Number of Files	300	1
Baud Rate	301	2
File 1 Record Length	302	3
File 1 Number of Windows	303	4
File 1 Maximum Record Number	304	5
File 1 Length of Key	305	6
File 1 Window 1 Status Register	306	7
File 1 Window 1 Record Number	307	8
File 1 Window 1 Record Image	308..312	9..13
File 1 Window 1 Command Register	313	14

Figure 4-1 Example Rack Interface Map

In the example, we will monitor contact 100-1 and when it closes, copy registers 101 through 103 to the CAM500 File 1, Window 1 record image area and perform a Retrieve by Key. If the key is found, we will copy the last two registers of the retrieved record to registers 104 and 105 and close contact 100-2, if the record is not found, we will close contact 100-3.

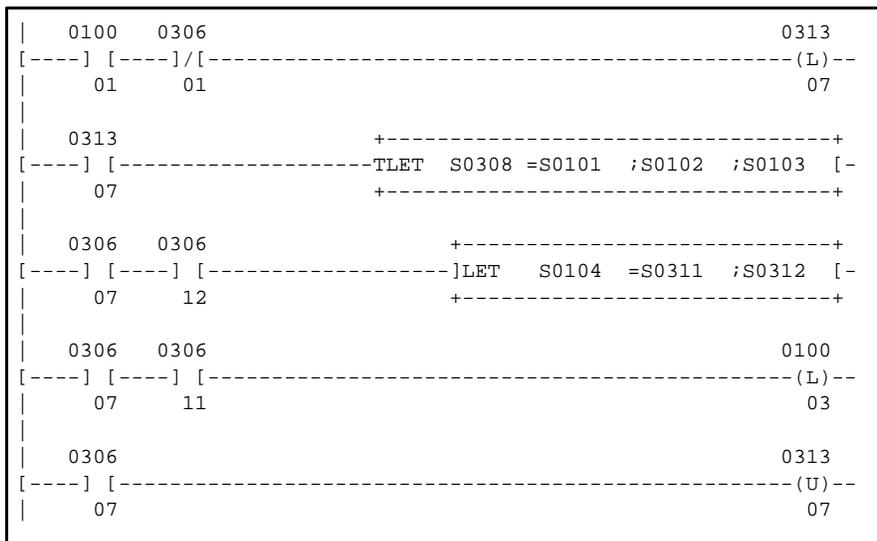


Figure 4-2 Retrieve by Key Ladder Example

Execution of this sequence will require at least three scans.

SCAN 1 - In the first scan that contact 100-1 is closed, 313-7 (Retrieve by Key) will be latched and the key data copied to the Record Image. Note in the first rung that 313-7 is not closed until any previous operation completes as indicated by 306-7 being open. At the end of this first scan, the PLC will write these registers to the CAM500 and it will start the search.

SCAN 2 - During the second scan, the CAM500 will be searching. If the CAM500 takes longer than one scan to perform the Retrieve by Key, additional scans will pass until the CAM500 closes the Retrieve by Key Complete contact, 306-7. When this contact is closed, the Record Image registers will be configured as PLC inputs presenting the retrieved record to the PLC. In most applications, the CAM500 will have ample time to perform the search in one scan. At the end of the second scan (or the first scan after the CAM500 is done searching) the PLC will read from the CAM500 rack interface including the Record Image registers 308 through 312 and the Status Register 306 with the Retrieve by Key bit set.

SCAN 3 - In the third scan, the closed contact 306-7 in conjunction with the Found bit contact 306-12 will cause the TLET to copy the retrieved data to another application register and set application contact 100-2. If the record was not found, then the third scan will detect contacts 306-7 and 306-11 (the Not

Found bit) and will close contact 100-3. In either case, the Store by Key command contact 313-7 will be unlatched. At the end of the third scan, this contact will be transferred to the CAM500 and it will turn the Record Image registers back into PLC outputs to allow the PLC to write the next command.

Although it will not be necessary in every application to copy the retrieved data from the Rack Interface, any decision or computation referencing this data must be made before the command bit contact 313-7 is unlatched because the Record Image will be output registers on the next scan after the command bit is opened.

Storage Operations

Store By Record Number

Command bit	Command Register bit 5
Completion bit	Status Register bit 5
Required parameters	Record Number and Record Image
Returned results	None
Not Found (bit 11)	Set if the record number is out of range.
Found (bit 12)	Set if the record replaced a previous copy.
Full (bit 13)	Set if all slots in the file are filled.
Empty (bit 14)	Cleared if the operation succeeds.

Store by Record Number is used to place a data record in a specified slot within a file. First copy the data record to the record image registers of one of the file's windows and set the record number to the desired destination. Then set the Store by Record Number bit in the command register. When the operation is completed, the CAM500 will set the Store by Record Number Complete bit in the status register. If a record had previously been stored in the specified file slot, either by record number or by key, the Found status bit will be set and the old record will be overwritten by the new one. If the specified slot was empty (i.e. had not been written since the last Delete Record, Delete All, or Configure Module operation), the Not Found status bit will be set and the new record will be inserted. To avoid overwriting the previous contents of a record, use a Retrieve by Record Number before the store to determine if the slot is occupied or use Store by Next Record Number which will search for a free slot. Records written using the Store by Record Number cannot be found by Retrieve by Key or Delete by Key operations.

Store by Key

Command bit	Command Register bit 4
Completion bit	Status Register bit 4
Required parameters	Record Image
Returned results	Record Number
Not Found (bit 11)	Set if no slot can be found to store the record.
Found (bit 12)	Set if the record replaced a previous copy.
Full (bit 13)	Set if all slots in the file are filled.
Empty (bit 14)	Always cleared by this operation..

Store by Key is used to write a data record to a file so that it can later be retrieved with the same key value. On successful completion the operation returns the Record Number where of the slot where the record was stored. This can be saved and later used with record number operations to delete or retrieve the record. If the key portion of the Record Image is found in an already existing record, the Found bit is set and the old record is overwritten with the new one. If the key is not found, the CAM500 locates an empty slot and inserts the new record into it, neither the Found bit nor the Not Found bit is set. The

Not Found bit is set if neither a record with the specified key nor a free slot can be found. If multiple records with duplicate keys must be stored, you must add an additional register to the key field and put a record serial number in it to make the keys unique. Records stored with Store by Key may be retrieved or deleted by providing an identical key. Records stored with Store by Key may be read out sequentially with Retrieve by Next Record Number but the retrieval order may bear no relation to the key value.

Store by Next Record Number

Command bit	Command Register bit 6
Completion bit	Status Register bit 6
Required parameters	Record Number and Record Image
Returned results	Record Number
Not Found (bit 11)	Set if no free slot can be found to store the record.
Found (bit 12)	Set if the operation succeeds.
Full (bit 13)	Set if all slots in the file are filled.
Empty (bit 14)	Cleared if the operation. succeeds.

Store by Next Record Number is used to place a record in the next available free slot within a file. First copy the data record to the record image registers of the window. The Record Number register is used to specify the first slot to examine. If the specified slot is unoccupied, the new record will be placed there and the Record Number will be incremented before being returned to the PLC as an input. If the specified slot is in use, the CAM500 will search forward through the file for a free slot. If one is found, the new record will be inserted there and the Record Number will be set to the following slot. Note that the record number returned by Store by Next Record Number is not necessarily a free slot. To sequentially store data in a CAM500 file, set the Record Number to zero and then perform successive Store by Next Record Number operations. Unlike other operations, the Not Found bit is set by Store by Record when a free slot cannot be found. The Found bit is set when a free slot is found and the new record placed in it. Store by Next Record Number will not overwrite a previously stored record (until it has been deleted). When the Record Number exceeds the Maximum Record Number for the file, searching will halt and Not Found will be set. The CAM500 will not wrap the record number back around to zero. If records are being stored with Store by Next Record and randomly deleted, you may augment the search for a new free slot by resetting the record number to zero and retrying the store in the event of a Not Found unless Full is also set in which case Store by Next Record Number will never succeed.

Retrieval Operations

Retrieve by Record Number

Command bit	Command Register bit 8
Completion bit	Status Register bit 8
Required parameters	Record Number
Returned results	Record Image
Not Found (bit 11)	Set if the specified slot was empty.
Found (bit 12)	Set if the operation succeeds.
Full (bit 13)	Set if all slots in the file are filled.
Empty (bit 14)	Set if there are no records in the file.

Retrieve by Record Number is used to retrieve the contents of a record when the slot number where it was stored is known. This will either be the record number used with Store by Record Number, the record number returned by Store by Key, or the record number returned by Store by Next Record number less one. Retrieve by Record Number sets the Found or Not Found bits depending on whether or not a record is stored in the specified slot. To sequentially read all of the records in a file, perform Retrieve by Record Number with a record number of 0 followed by successive Retrieve by Next Record Number operations until the Not Found bit is set.

Retrieve by Key

Command bit	Command Register bit 7
Completion bit	Status Register bit 7
Required parameters	Record Image (Key portion only)
Returned results	Record Number and Record Image
Not Found (bit 11)	Set if the specified key was not found.
Found (bit 12)	Set if the operation succeeds.
Full (bit 13)	Set if all slots in the file are filled.
Empty (bit 14)	Set if there are no records in the file.

Retrieve by Key is used to retrieve a record stored with the Store by Key operation. Retrieve by Key cannot locate a record stored by the other storage operations or whose key field has been modified by a Store by Record Number subsequent to the original Store by Key. Retrieve by Key will return the record image as well as the record number of the slot the record was found in and set the Found bit if successful. If no record is found matching the key portion of the record image, the Not Found bit is set.

Retrieve by Next Record Number

Command bit	Command Register bit 9
Completion bit	Status Register bit 9
Required parameters	Record Number
Returned results	Record Number and Record Image
Not Found (bit 11)	Set if no record was found.
Found (bit 12)	Set if the a record was found.
Full (bit 13)	Set if all slots in the file are filled.
Empty (bit 14)	Set if there are no records in the file.

Retrieve by Next Record Number is used to sequentially read out records previously stored by any means. The record with the specified record number itself is not returned. Instead, Retrieve by Next Record starts searching for a record in the slots from Record Number plus one to the end of the file. Therefore, Retrieve by Next Record returns the record with the lowest record number that is greater than the value of Record Number at the beginning of the operation. If a record is found, the Found bit is set and the Record Number is rewritten with the slot number of the record found. This allows successive calls to Retrieve by Next Record to return all of the active records in a file without modifying Record Number between invocations. If no record is found between Record Number and the end of the file, Not Found is set and the Record Number is not altered. Retrieve by Next Record Number can be used to retrieve records stored with Store by Key but the retrieval order is not defined with respect to the key value or the sequence of storage operations.

Retrieve by Previous Record Number

Command bit	Command Register bit 10
Completion bit	Status Register bit 10
Required parameters	Record Number
Returned results	Record Number and Record Image
Not Found (bit 11)	Set if no record was found.
Found (bit 12)	Set if the a record was found.
Full (bit 13)	Set if all slots in the file are filled.
Empty (bit 14)	Set if there are no records in the file.

Retrieve by Previous Record Number is similar to Retrieve by Next Record Number described above but reads backwards towards record number zero. Each invocation of Retrieve by Previous Record Number returns the record with the greatest record number that is less than the specified Record Number. To sequentially read all records in a file, set Record Number to the Maximum Record Number in the file and perform a single Retrieve by Record Number. Then perform successive Retrieve by Previous Record Number operations until Not Found is set.

Multiple Record Sequential Retrieval

Command bit	Command Register bit 15
Completion bit	Status Register bit 15
Required Parameters	Record Number
Returned Results	Up to 128 registers of record number, record image appearing after last window of last file. Record Number.
Not Found (bit 11)	Set if less registers were found than would fit in the 128 register result block.
Found (bit 12)	Set if at least one record was found. Clear if no records were found.
Full (bit 13)	Set if all slots in the file are filled.
Empty (bit 14)	Set if there are no records in the file.

This function retrieves multiple records sequentially in a single operation. It is equivalent to performing multiple Retrieve Next Record Number operations except that the results are written to a different set of registers. As many whole records as will fit (up to the number of records in the file) are copied to the block of 128 registers immediately following the command register of the highest numbered window of the highest numbered file. The Record Number register of the window will be set to the record number of the last record retrieved.

For each record found, its record number is written to the output area followed by a copy of the record's registers. Therefore, each record consumes Record Length + 1 registers of the 128 register output area. The number of records retrieved is the lesser of the number of records found in the file or the integer part of 128 divided by the record length plus one.

To retrieve all records from a file using this operation, start with the Record Number set to zero and execute this operation repeatedly until the Not Found bit is set. The register following the last retrieved record image (which would have held the record number of a successive record) is set to zero indicating the end of the data. Like Retrieve Next Record Number, this function cannot be used to retrieve record number zero. The end of the data can also be located by comparing the Record Number register to the record numbers in the output area.

Note that only one window should be asked to perform this operation at any one time since all windows of all files write the results of this function to the same register area.

Delete Operations

Delete by Record Number

Command bit	Command Register bit 3
Completion bit	Status Register bit 3
Required parameters	Record Number
Returned results	None
Not Found (bit 11)	Set if the specified slot was empty.
Found (bit 12)	Set if the a record was found and deleted.
Full (bit 13)	Set if all slots in the file are filled.
Empty (bit 14)	Set if there are no records in the file.

Delete by Record Number erases the record with the specified Record Number making it unavailable for future Retrieve operations and freeing its slot for future Store operations. Delete by Record Number will set the Found bit if the slot specified by Record Number is occupied or Not Found if it is empty.

Delete by Key

Command bit	Command Register bit 2
Completion bit	Status Register bit 2
Required parameters	Record Image (Key portion only)
Returned results	Record Number
Not Found (bit 11)	Set if the specified key was not found.
Found (bit 12)	Set if the a record was found and deleted.
Full (bit 13)	Set if all slots in the file are filled.
Empty (bit 14)	Set if there are no records in the file.

Delete by Key searches for a record whose key value matches that of the record in the Record Image portion of the window. If found, the record will be erased making it unavailable to subsequent Retrieve operations and available for subsequent Store by Key or Store by Next Record Number operations. If the search is successful, Found is set and the record number of the slot that held the record is returned in Record Number. If no record is found matching the specified key, Not Found is set and the file is unaffected.

Delete All

Command bit	Command Register bit 1
Completion bit	Status Register bit 1
Required parameters	None
Returned results	None
Full (bit 13)	Cleared by this operation.
Empty (bit 14)	Set by this operation.

Delete All erases all records in a file. This is quicker than performing a lot of individual Delete Record operations. Delete All only erases the file in whose window it is invoked, other files in the same module are not affected.

Sequential Storage and Retrieval

Many CAM500 applications take the form of sequential storage and retrieval of data. A PLC may be logging events or reports that will be sequentially retrieved by a host computer, a host computer may be storing orders for sequential processing by the PLC, or a PLC may be performing both storage and retrieval operations to buffer data. Normally, sequential storage is performed using Store by Record Number or Store by Next Record Number and retrieval is accomplished with Retrieve by Record Number and Retrieve by Next Record Number. In some applications it may be advantageous for the external accessors to maintain their own record counters and use strictly Store by Record Number and Retrieve by Record Number. In many cases, Delete by Record Number will be used to erase each record as it is retrieved and processed. Typically a separate window is assigned for storing and retrieving data. Post retrieval deleting of data records may be accomplished by invoking the Delete by Record Number immediately after any retrieval operation.

The following ladder code fragment will store records sequentially. For purposes of the example, assume that the CAM500 is rack addressed at PLC register 300 and that File 1 has a record length of 5. This will place the File 1, Window 1 Command Register at PLC register 313, the Status Register at PLC register 306, and the Record Image at PLC register 308. In the example, we will monitor contact 100-1 and when it closes, copy registers 101 through 105 to the CAM500 File 1, Window 1 record image area and perform a Store by Next Record Number. If no free slot can be found but the file is not full, the ladder program will change the Record Number to 0 and try the operation again. If the file is full, the ladder program will close contact 100-3. If the operation succeeds either the first time or on the retry, the program will close contact 100-2. Contact 110-1 is used internally by the program to remember that a retry is needed on the next scan.

The following ladder fragment can be used to print out a sequential file. Sequentially stored data could also be retrieved by an external device such as a computer running CAM500SW. Assume the same setup as above. Contact 100-15 is closed to initiate the file readout. Contact 100-16 will be closed when the readout operation completes.

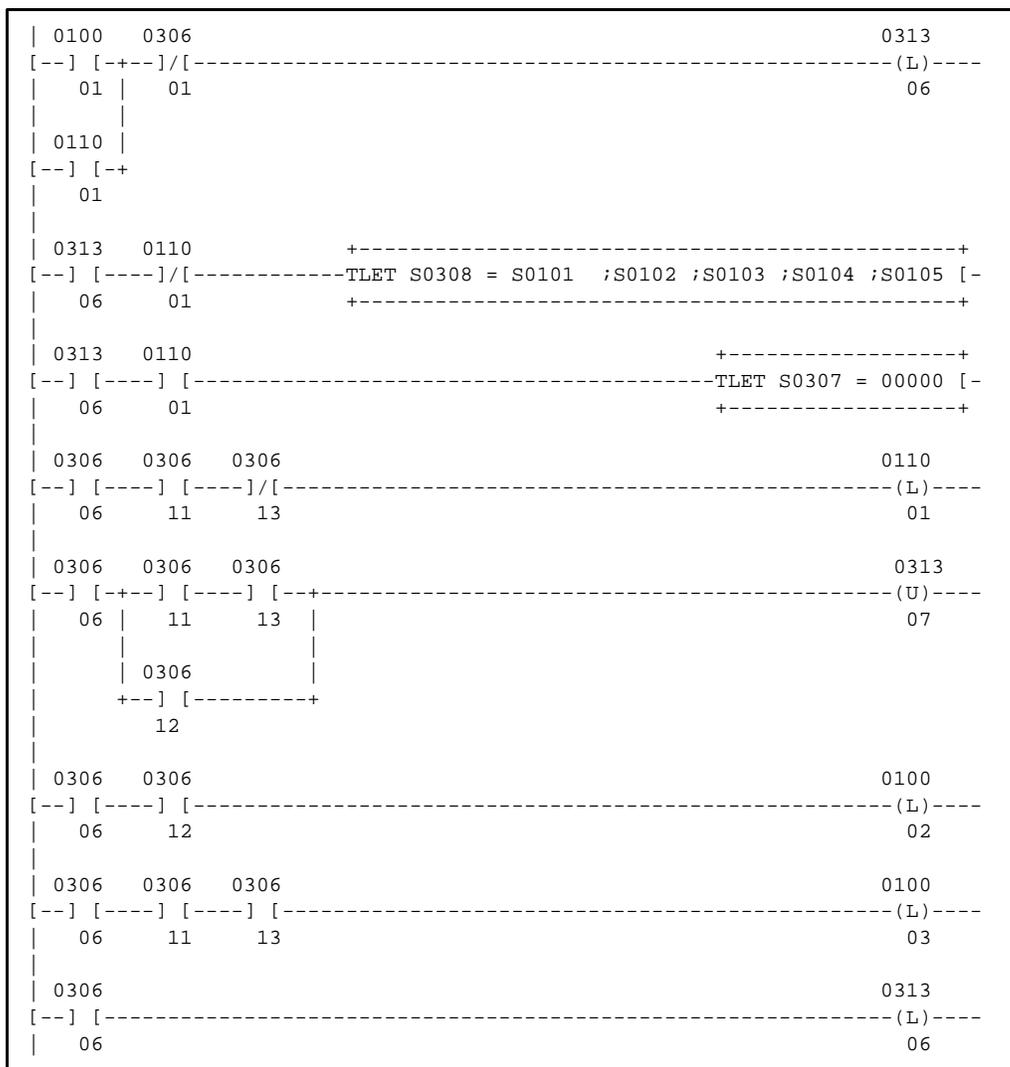


Figure 5-1 Ladder Example

The ladder fragment shown uses the second window of file 1 with Status Register at 314 and Command Register at 321. Using a different window allows storage of records to continue uninterrupted by the sequential readout. Additional bits of register 110 are used for internal state storage. Contact

110-2 indicates that the initial Retrieve by Record Number (0) is in process. Contact 110-3 indicates that the subsequent Retrieve by Next Record Number operations are in process. Register 109 is used as a status register for the print rung.

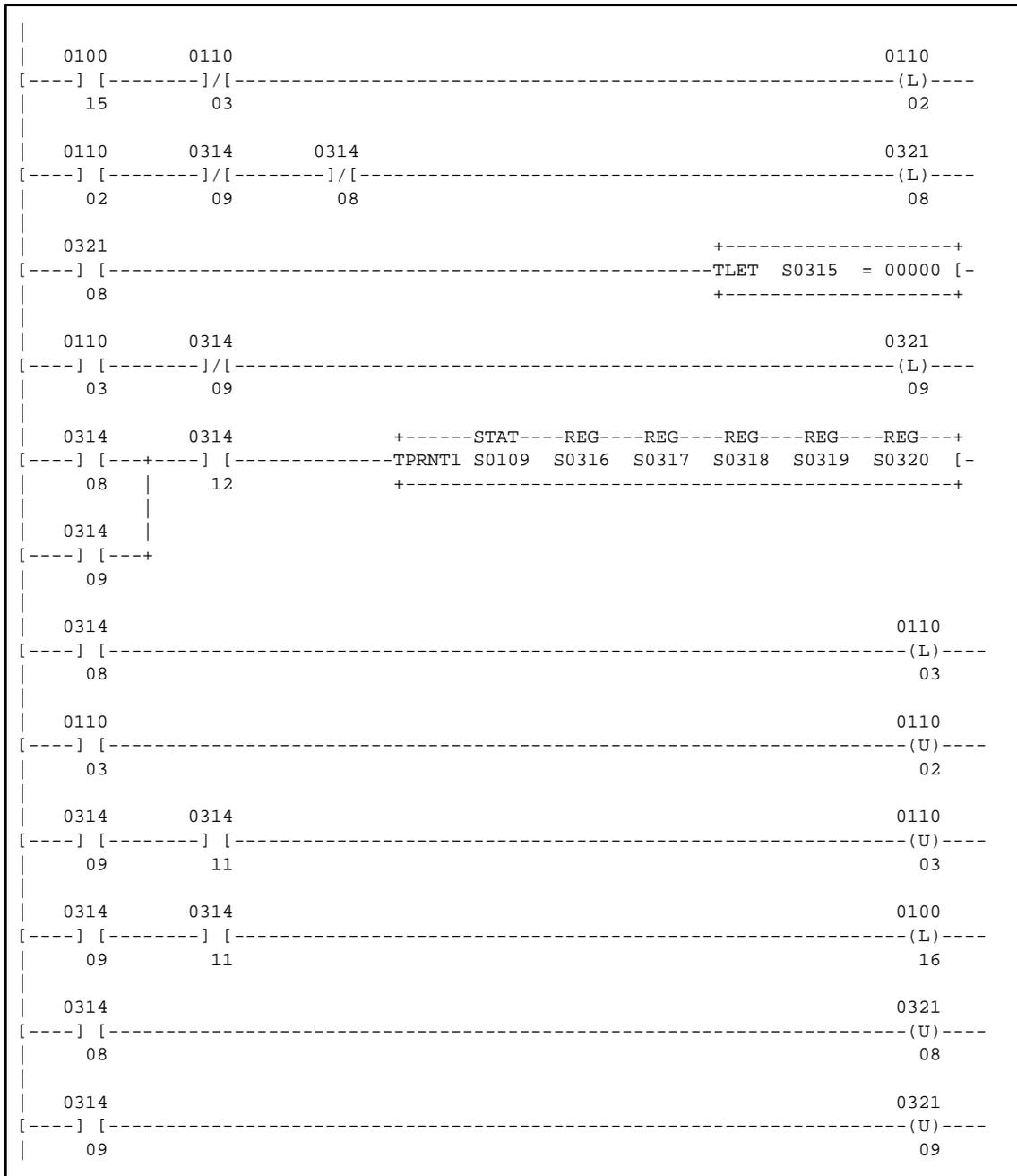


Figure 5-2 Ladder Example

Random Storage and Retrieval

Data records can be stored and retrieved randomly without including a key field in the record by using Store by Record Number and Retrieve by Record Number as long as it is possible to generate appropriate record numbers in the range of zero to the Maximum Record Number. The application must compute this number. An application with a single word key is a candidate for random access methods. Unlike key search operations, random access times are consistently fast. The worst case and best case search times are the same, given the same level of concurrent operations and port I/O.

Keyed Storage and Retrieval

When key values identifying a record occur in a greater range than the possible range of Record Numbers in a file, Store by Key and Retrieve by Key may be used. For these operations, the record is identified by one or more registers within the record up to the length of the entire record. Keyed operations are slower than random access operations but in most cases can be performed more quickly than the attached PLC can process the data anyway. Keyed storage operations are a powerful tool in application development and require minimal effort on the part of the application programmer to deliver high performance. The key searching operations in the CAM500 are extremely rugged and never need reorganization or integrity checking unlike personal computer oriented database systems.

Sequential Retrieval of Random or Keyed Storage

Sequential retrieval operations, such as the sample file readout program given above, can be used to access data that has been stored by key. The relationship between storage order, key value, and the order of interleaved delete and store operations and the order in which records will be sequentially retrieved is complex and unpredictable for all practical purposes. Nevertheless, sequential readout of keyed data is a useful tool for unloading event or report data to a host computer, checking for outstanding order records, or copying the CAM500 contents to computer for backup or transfer to another module. The general procedure is to start with a zero in Record Number and perform a single Retrieve by Record Number. If this operation is successful, copy and process the resulting data. Whether or not the Retrieve by Record Number succeeds, perform subsequent Retrieve by Next Record Number operations and process the retrieved data until the Not Found bit is set by Retrieve by next Record Number. Readout in reverse order is also possible by using Retrieve by Previous Record Number. If it is desired to delete the records as they are retrieved, follow each retrieve operation (including the original Retrieve by Record Number) with a Delete by Record Number without changing any of the other window parameters.

Read Only Applications

The CAM500 memory can be write protected by setting bit 2 in module register 8192. This must be accomplished through the front panel port. When the write protect bit is set, the CAM500 will ignore all Store and Delete command bits. Obviously, you should first load the data to be accessed before setting the write protect bit. With the configuration lock (8192.1) and write protect (8192.2) bits set, the module contents cannot be altered through the register rack bus. When the configuration lock bit is set, the file definition registers are PLC inputs only and cannot be modified except through the front panel port.

Tips & Tricks

Setting the configuration lock bit causes the CAM500 to convert all configuration registers including Number of Files and the Record Length, Key Length, Maximum Record Number and Number of Windows registers for each file to PLC inputs. If it is later desired to write these registers from the PLC, the configuration must be unlocked through the front panel port and rack power cycled.

A file can be set up with a record length of zero. The CAM500 can be used in this way with Store by Record Number, Retrieve by Record Number, and Delete by Record Number in an application that only needs to track a binary state (exists or not) of a group of record numbers.

When any register is written through the front panel port, the CAM500 will change that register to a PLC input. If the register is a Record Image or Record Number, it will change back to a PLC output when all command bits are again zero. Any other register, including the command register for a window, will stay configured as a PLC input until rack power is cycled. A simple way to avoid problems related to this effect is to use CAM500SW through the front panel port for configuration and lock the configuration. Then perform all other CAM500SW operations with the exception of a Restore through the PLC front panel port. If you follow this rule, you don't have to understand the switching of registers between and output.

It is possible to set both Retrieve by Key Delete by Key command bits at the same time and wait for the completion status bit of the delete operation. The retrieve operation will be performed first, followed by the delete. The delete operation does not modify the record image portion of the window and the Found or Not Found bits will be set the same way by both operations. Internal logic in the CAM500 ensures that the retrieve will be performed first. Note that this is true only when both commands are requested in the same window. If this retrieve-and-delete method is being used concurrently with storing data in the same file through another window, assign the Store by Key operation to a higher numbered, lower priority window to avoid a condition where a record is unsuccessfully sought by the retrieve, then stored by a higher priority window, and then deleted even though it was never retrieved.

Likewise it is possible to set both a Retrieve by Record Number, Retrieve by Next Record Number, or Retrieve by Previous Record Number command bit and a Delete by Record Number command bit at the same time and wait for the completion status bit of the delete operation. All of the Retrieve operations leave the record number of the retrieved record in the Record Number register and the delete operation does not modify the Record Image portion of the window. The retrieve operation will be performed first, followed by the delete. Note that this is true only when both commands are requested in the same window.

The Full and Empty status bits in each window are independent. If an operation in one window changes the full or empty status of the file, this change is not reflected in the status bits of the other windows

until the next time those bits are updated by an operation completion in their respective window. The front panel port of the CAM500 can be used to provide an extra port for interfacing an operator interface or display to the PLC. To accomplish this, the ladder program keeps data needed by the external device in any part of the rack interface not otherwise being used. The external device uses standard SY/MAX protocol read and write messages to access the data.

Because the green "Searching" light is illuminated during each file operation, a rough idea of how busy the module's processor is may be gained from the relative intensity of the light.

For absolute maximum speed in Keyed operations, configure the module with a single file with Maximum Record Number equal to 65,516. Assign as few windows as possible to the file. Do not use the front panel port. If an external device must access the module, use a PLC port. (A Niobrara SPE4 can be used to get more PLC ports.) Make the record and particularly the key as short as possible. If the key length can be a single register and enough data registers are available to set the Maximum Record Number to 65535, use Record Number operations with the key as the record number.

Changing the order of registers in a key can affect search time. If key search operations on a multi-register key are slow, try rearranging the key registers.

In cases where a lot of records are Stored by Key, later Deleted by Key and Stored by Key again with the file more than 80% or 90% full and particularly with long multi-register keys, CAM500 performance may degrade with time. If this occurs, try unloading the file to disk with CAM500SW, erasing the file, and reloading it into the module. This procedure performed periodically may improve throughput.

An MS/DOS compatible software program is included with the CAM500 to facilitate configuration and maintenance of the module. This program is a convenient way to:

- Develop CAM500 module configurations.
- Save module configurations on disk.
- Load module with data from disk files.
- Unload data from the module to disk.
- Control the configuration lock and write protect bits.
- Backup and restore module contents.
- Transfer module contents to another module.
- Directly manipulate the rack interface registers.
- Convert a file from one record length to another.

CAM500SW can communicate with the module either through the front panel port or through the register rack via a SY/MAX PLC communications port. The personal computer can be connected to the module through a direct connection, via modem, through SY/NET or any other SY/MAX compatible transport. For a simple, direct, connection, Niobrara's SC406 RS232 to RS422 converter/cable is recommended.

Main Menu

CAM500SW starts with the main menu, see Figure 7-1. This menu allows access to the various functions of the software. These functions have been divided into groups including: **Configuration**, **Transfer**, **Backup**, **Utilities**, **Protect**, **Setup**, and **Quit**. These groups usually include a sub-menu which is accessed by pressing the Capital letter from the name of the group or maneuvering the highlighted cursor to the group and pressing the **Enter** key.

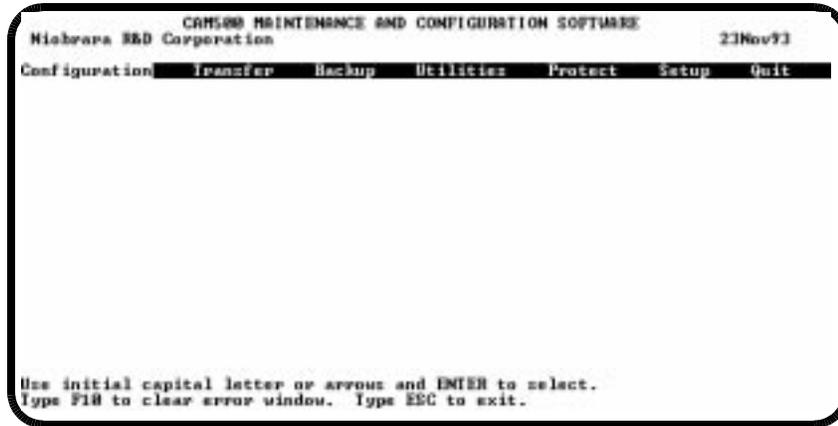


Figure 7-1 Main Menu

SETUP

The setup menu accesses the setup parameters for the personal computer to enable it to communicate with the CAM500. The parameters chosen will depend on the exact equipment involved in making the connection. The connection type is mainly determined by the method of connection to outside world and may be broken into two groups: the personal computer's COM: port and the SY/LINK Network Interface Card.

Personal Computer COM: port

If the connection from the personal computer is made through one of its serial ports then the Connection type should be one of the following:

- Sy/Max COM:
- Net-to-Net COM:
- Gateway COM:

Sy/Max COM: is the default and most likely will be the one used. In this mode the personal computer will communicate through one of its COM: ports as though it were a SY/MAX device such as a PLC. The full SY/MAX protocol is supported including routing so SY/MAX COM: may be used through SY/MAX mode ports on NIMs and SPE4s with appropriate routing. This mode is to be used when a direct connection from the personal computer COM: port is made to the port of the CAM500. In most cases an RS-232<>RS-422 conversion is required and the Niobrara SC406 cable makes this conversion very convenient.

Net-to-Net COM: is used when connecting to a NIM or SPE4 that is set to Net-to-Net mode. The first drop number in the route will be that of the address of the NIM.

Gateway COM: is used when connecting to an SPE4 port that is in Gateway mode. For more information about Gateway mode see the SPE4 instruction manual.

Port - When one of the COM: connection types is selected a particular port of the personal computer must be selected. Available choices are: COM1:, COM2:, COM3:, and COM4:. Select the port which will be used to connect to the AIM4.

Baud rate - The Baud rate selected here is the baud rate of the personal computer serial port selected. This value should be set to match the device connected to the personal computer. An CAM500 has a default baud rate of 9600 and if a direct connection is made to the CAM500 this is the setting that should be made on the personal computer. If the baud rate of the port of the CAM500 has been changed this value may need to be adjusted. The settings available for the personal computer's baud rate are of a wider selection that are actually available for the port of the CAM500. This is because

there may be instances where other intermediate devices between the personal computer and the CAM500 may require different baud rates; such as modems, SPE4s, and NIMs.

Data bits - When in SY/MAX or Net-to-Net modes the data bits is required to be 8 and may not be changed. The SY/MAX protocol requires 8 data bits. The Gateway mode uses ASCII messages which do not require the full 8 data bits and may be set to 8 or 7 depending on the attached device.

Stop bits - The stop bits are normally set to 1 but may be adjusted to 2 for some particular application. The Print/Prog port of the AIM4 is set for 1 stop bit.

Parity - SY/MAX and Net-to-Net modes normally use EVEN parity and that is the default for the Print/Prog port on the AIM4. Other choices are ODD and NONE.

Route - The route is used to determine the path from the personal computer to the CAM500. If a direct connection is made from the personal computer to the CAM500, i.e. without going through a SY/NET network or an SPE4, this value is set to NONE by pressing the Delete key. If a SY/MAX connection is made to a SY/MAX mode port on an NIM or SPE4 the first drop will be that of the drop number of the NIM or SPE4 port. If any Net-to-Net drops are included between the port connected to the personal computer and the port connected to the CAM500, they must be included in order of occurrence from the personal computer to the CAM500. The last drop number listed will be that of the NIM or SPE4 SY/MAX mode port that is connected to the CAM500. Up to 8 total drops are supported by the SY/MAX protocol.

If the personal computer is in Net-to-Net mode the first drop will be that of the Net-to-Net port of the NIM or SPE4 that the personal computer is connected to. Subsequent drops will be included like above.

The Gateway mode route will include the Gateway port on the SPE4 that the personal computer is connected and any subsequent Net-to-Net and SY/MAX drops to reach the CAM500.

Connected VIA - If you are connected to the module through a PLC communication port, change the Connected VIA to PLC PORT otherwise select Front Panel.

Rack Base Address - If you are connected to the module through a PLC communication port, make sure the Rack Base Address parameter matches the first PLC register assigned to the module in the PLC's rack address table.

SY/LINK Connection

CAM500SW REV 02Feb94 and later provides full support of the Square D SY/LINK network interface card. Setup for the network interface is provided along with setup for the RS-422 port on the card.

Connection type - The RS-422 port may be set to SY/MAX or Net-to-Net modes. For a direct connection to the CAM500 from the RS-422 port of the SY/LINK card choose the Sy/Link Direct mode. If an indirect connection from the RS-422 port of the card is made through other Net-to-Net ports choose Sy/Link Net-to-Net. If the RS-422 port is not used and the connection is made through the SY/NET network to another NIM, the choice does not matter.

Base address - This is a hex value that represents the SY/LINK's cards address range selected by DIP switches on the card. Select the same range that is set on the card.

RS422 Baud rate - Select the baud rate to match the external device, normally 9600.

RS422 Data bits - Select the data bits to match the external device, normally 8.

RS422 Stop bits - Select the stop bits to match the external device, normally 1.

RS422 Parity - Select the parity to match the external device, normally EVEN.

Sy/Net speed - Select to match the speed settings of the other devices on the SY/NET.

Sy/Net size - Select to match the setting on the other SY/NET devices.

Route - The first drop in the route defines the network address of the SY/LINK board. Since the personal computer is connected to the SY/LINK card through the edge connector of the card, port 0, the drop number must start with 0. The remaining two digits of that drop should be selected not to match

any other device on the SY/NET. For instance, there are three NIMs on the network addressed 01, 02, and 03. It seems logical to make the SY/LINK card be at address 04 so the first drop in the route field will be 004. The next drop will be that of the NIM port connected to the CAM500, or another Net-to-Net port. If Sy/Link Direct was selected and the CAM500 is connected directly to the RS-422 port of the SY/LINK card the full route statement would be 004 104 as the RS-422 port is considered to be port 1.

Command Line Parameters

CAM500 (REV 02Feb94 and later) may be started from the DOS command line with a sequence of letters which represent the keystrokes necessary to perform any operation. This allows batch processing of various commands such as downloading of stored setup files. The special characters /R, /D, and /E refer to the Return key, Delete key, and Escape key respectively. The parameters are not case sensitive.

Another way of changing the Setup is to copy the setup file to another name in DOS, then copy it back to CAM500SW.STP in the batch file before calling CAM500SW.

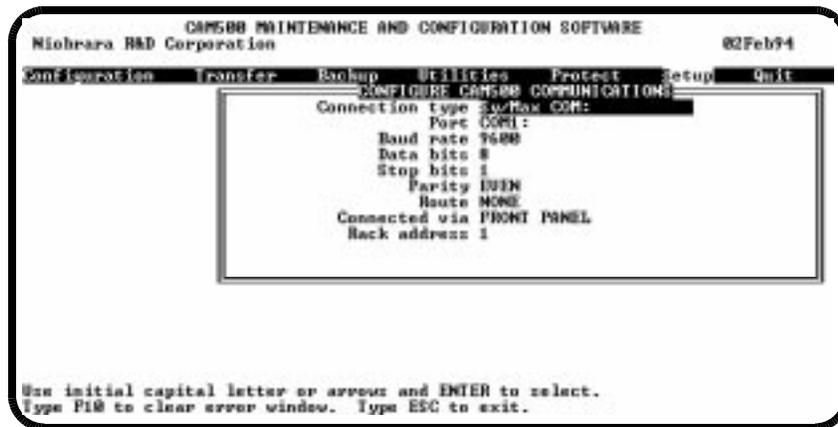


Figure 7-2 Setup Screen

Configuration Menu

CAM500SW provides an editor for preparing and maintaining module configurations. To create a new configuration, simply select **Configuration**, see Figure 7-3, from the main menu and **Edit Configuration in PC memory** from the sub-menu that then appears. To edit an already prepared configuration, it can either be fetched from the attached CAM500 module with the **Fetch configuration from module** option or be loaded from a disk file with **Load from disk to PC memory**. The disk file must have previously been created with the **Write from PC memory to disk** option.

Fetch configuration from module

A previously configured CAM500 module may have its configuration fetched and placed into PC memory by the use of the **Fetch configuration from module** routine. If the fetch was successful, a message will be displayed, otherwise an error message will appear and the setup and cabling should be checked.

Edit Configuration in PC memory

When the editing window is displayed, Figure 7-4, you use the **up-** and **down arrows** to select one of the fields in the description of a file and use the **Page -Up** and **-Down** keys to select any file in the configuration. Press the **F1** key to accept a configuration and return to the Configuration Sub-menu. Pressing the **Esc** key will abort any changes in the configuration in the PC memory. Below the file definition window, the number of free data and interface registers available is continuously displayed to assist in optimum allocation of CAM500 resources.

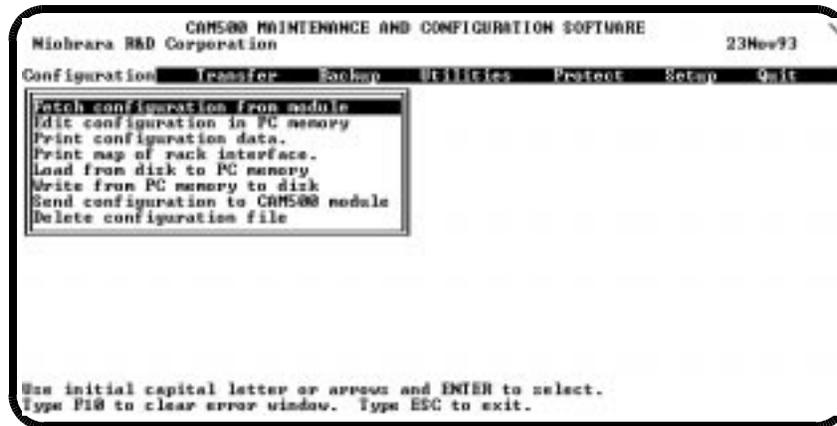


Figure 7-3 Configuration Sub-menu

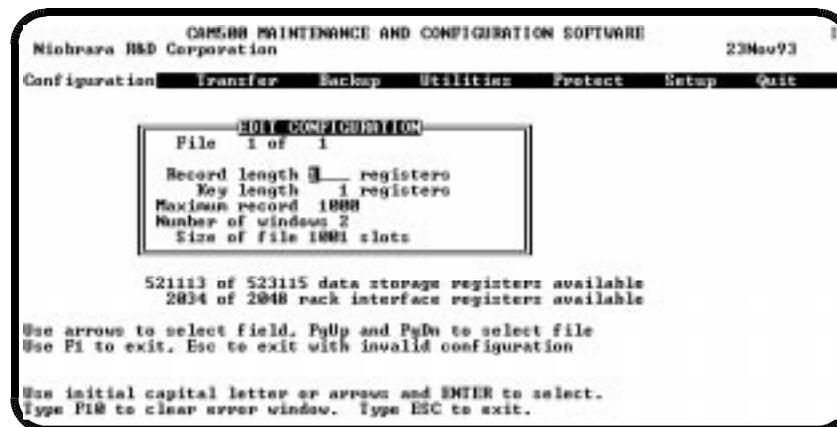


Figure 7-4 Edit Configuration screen

Print configuration data, Print Map of rack interface

CAM500SW can produce two reports which facilitate application design. The Configuration Listing report shows the parameters of all files in the configuration. The Interface Map report shows, for the configuration loaded in CAM500SW memory, the register numbers for the Status, Record Number, Record Image, and Command registers of each window. This report is handy during design of a CAM500 ladder program. In order for the PLC register numbers to be correctly reported, the Rack Base address in the CAM500SW Setup menu must match the first register number allocated to the CAM500 module in the PLC's rack address table.

Load configuration from disk to PC memory

A previously saved configuration may be loaded from disk to the PC memory for editing, or sending to the CAM500 module. Select the Load configuration from disk to PC memory selection and select the filename of the configuration file.

Write from PC memory to disk

When a configuration is prepared, it may be saved to a disk file for future reference. Select the write option from the Configuration menu.

Send configuration to CAM500 module

To load a configuration into a physical CAM500 module, the module must be connected to the personal computer. If the module configuration lock has been set, the connection must be through the front panel

port. If access to the configuration lock and write protect bits is not necessary, the connection may be through the PLC programming port. This method allows configuration of the module through the same connection used to program the PLC. Ensure that the Rack Base Address of the software matches the first register allocated to the module in the processor's rack address table.

Before sending the configuration to the module, load it into CAM500SW either by entering it in the configuration editor, loading it from disk, or fetching it from another module. The host PLC and any other device accessing the CAM500 module should be halted during the configuration process. When the desired configuration is present in CAM500SW memory, select Configuration from the main menu and Configure from the sub-menu which appears. Successful transmission of the configuration is indicated by a message on the personal computer screen. If the CAM500 detects a problem in the configuration parameters, the red "Not Found" light on the front of the module will be illuminated at the end of the configuration process.

Delete configuration file

The Delete operation allows the user to remove unwanted configuration files from the disk. Select the specific file to be removed and press **Enter**.

Transfer Menu

The Transfer menu allows the options of loading record data into the CAM500 from a computer, unloading data from a CAM500 to a computer, changing the record size of a file, and deleting saved disk files. See Figure 7-5.

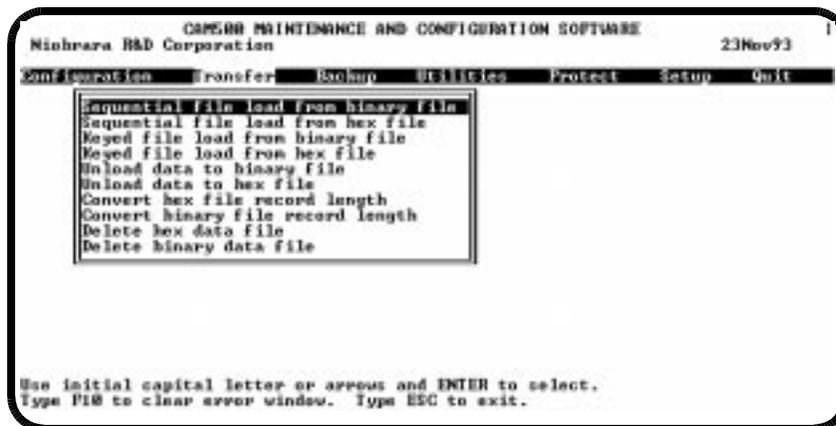


Figure 7-5 Transfer Menu

Loading computer data into a module

A file prepared by another computer program may be transferred to the module by CAM500SW. The software supports transferring of two types of files, binary format and hex format, in two CAM500 modes, Store by Key and Store by Record Number.

A binary file should be prepared as a direct image of the binary records to be loaded into a CAM500 file. Each register is composed of two bytes with the least significant byte first according to the Intel 80x86 convention. Each record is composed of Record Length number of two byte registers. There must be no delimiters between registers or records. The CAM500SW load utility will fetch the configuration from the connected CAM500 to determine the record length and key length of the file. When prompting for the file to read, binary mode displays all files in the default directory with the ".BIN" extension.

A hex file is similar in format with the exception that each register is composed of four hexadecimal characters in the range 0-9 and A-F with the nybbles in order of descending significance. The alphabetic characters may be in any mixture of upper and lower case. The eighth bit of the characters is ignored. The hex file loader ignores all other characters in the input file allowing complete freedom in delimiting the file with tabs, blanks, commas, quotation marks, carriage returns, line feeds or any other character.

When prompting for the file to read, hex mode displays all files in the default directory with the ".HEX" extension.

If one of the sequential load modes is selected, **Sequential file load from binary file**, or **Sequential file load from hex file**, CAM500SW will prompt for the starting record number in the CAM500 file and the number of records to load. The actual number of records loaded will be the smaller of this specification, the number of records found in the disk file, or the number accepted by the CAM500 before it sets its File Full status bit. A sequential load replaces records already existing in the module with the records from the file at the same record number. Records in the module outside the range of the sequential load are unaffected.

The keyed load modes, **Keyed file load from binary file** and **Keyed file load from hex file**, operate similarly but do not require a starting register number as the location of the stored data will be determined by the CAM500's internal key processing. A keyed load replaces records already in the module with the same key value but does not affect records already in the module whose key values do not appear in the disk file.

When selecting the window to be used for the transfer, bear in mind that use of a window for this transfer process from the front panel port will render that window inaccessible to the host PLC until the next time rack power is cycled. As long as no other accessor is using the same window, the transfer can be performed with impunity on a running system through either the front panel port or a PLC communication port.

Unloading module data to the computer

When unloading data from the CAM500 to a computer disk file, the only choice for access is sequential. The software prompts for a starting record number and a record count. The disk file may be written as a straight binary image of the data, **Unload data to binary file**, or as a hex-ASCII formatted file, **Unload data to hex file**. The hex ASCII format separates registers in a record with commas and delimits records with a carriage-return line-feed sequence. These formats are compatible with the input requirements of the Load operations described above and convenient for access by external software.

Figure 7-6 displays the hex dump of two files: TEST1.BIN and TEST1.HEX. These files are the binary and hexadecimal representations of the same CAM500 data file. Notice that the hex file contains commas (2C hex), carriage returns (0D hex) and line feeds (0A hex) for delimiters and the binary file contains no delimiters.

```
C:\CAM500>dump test1.bin
00000000 01 00 01 00 02 00 16 00 03 00 21 00 04 00 05 00 .....!.....
00000010 7B 00 21 00 7C 00 21 00 91 2B 21 00          <.!.!..æ!..

C:\CAM500>dump test1.hex
00000000 30 30 30 31 2C 30 30 30 31 0D 0A 30 30 30 32 2C 0001,0001..0002,
00000010 30 30 31 36 0D 0A 30 30 30 33 2C 30 30 32 31 0D 0016..0003,0021.
00000020 0A 30 30 30 34 2C 30 30 30 35 0D 0A 30 30 37 42 .0004,0005..007B
00000030 2C 30 30 32 31 0D 0A 30 30 37 43 2C 30 30 32 31 .0021..007C,0021
00000040 0D 0A 32 42 39 31 2C 30 30 32 31 0D 0A          ..2B91,0021..

C:\CAM500>type test1.hex
0001,0001
0002,0016
0003,0021
0004,0005
007B,0021
007C,0021
2B91,0021

C:\CAM500>_
```

Figure 7-6 Binary and Hex DOS files

Converting the record format of module data

When application improvements demand a change, usually an increase, in the number of registers in a CAM500 file record, the module must be reconfigured. The normal procedure is to unload the module to one or more binary disk files, reconfigure the module, convert the file to a new record size, and reload the data again. The CAM500SW utilities menu includes functions for converting the record length of a CAM500 image file on the personal computer disk., **Convert hex file record length** and **Convert binary file record length**.

Deleting data files

The Transfer menu includes two delete functions for removing binary and hex files: **Delete binary data file** and **Delete hex data files**.

Utilities Menu

The Utility menu provides a convenient register viewer, a method for erasing a file in a CAM500, and hex and binary disk file record converters. See Figure .



Figure 7-7 Utility Menu

Viewing and Editing module registers

The **View module registers** function in the Utilities menu generates a display of register data similar to the Data screen in the PLC programming software, Figure 7-8. Register values displayed may be modified by moving the cursor to one of the data fields and typing a new value. If View is used to access the module through the front panel port, any write operation will change that register to a PLC input. If View is used with CAM500SW connected to a PLC port, any register in the PLC can be examined or modified. Caution should be exercised when modifying PLC register contents.

Erase CAM500 file

The **Erase CAM500 file** function in the Utilities menu allows a convenient method for erasing the data within a file in a CAM500 module, Figure 7-9. Simply select the file to be erased and the window for performing the delete. Pressing the **Esc** key will abort the erasure.

File record length conversion

The **Hex file record length conversion** and **Binary file record length conversion** functions are the same as those located in the Transfer menu. See section **Converting the record format of module data** on page 41 for details.

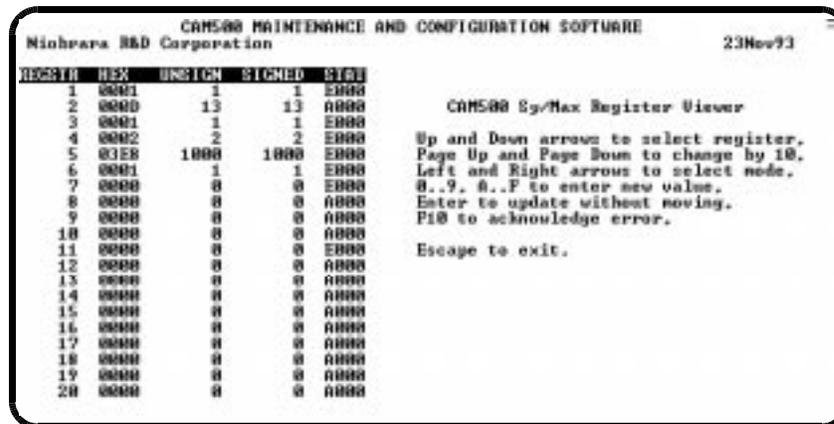


Figure 7-8 Register viewer screen

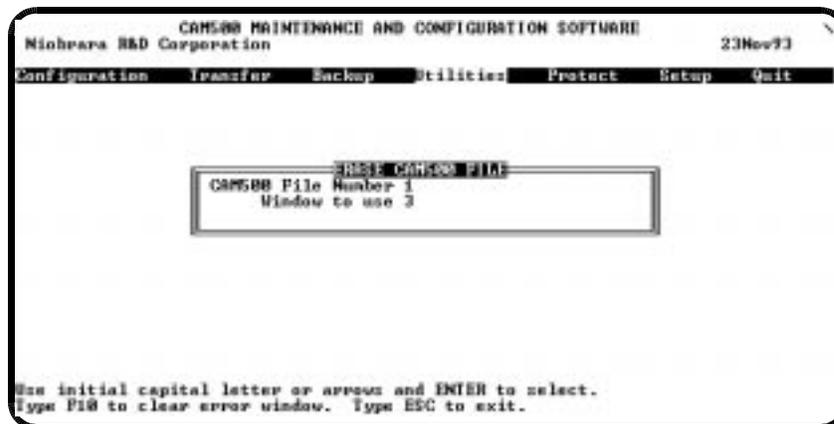


Figure 7-9 Erase Module Screen

Backup Menu

The CAM500SW module backup function makes a copy of the module configuration as well as all data from all files in the module in a single disk file. This file can later be reloaded into the same module or loaded into an additional or replacement module. If performed through the front panel port, backup also saves the state of the configuration lock and write protect bits. Unlike the Load disk file function, the Restore operation destroys all previous module contents. Restore returns all records to the same slot they were found in by Backup. Backup uses the highest numbered window for each file, if this is not compatible with your application you must halt the system prior to performing the backup. Restore will invoke the module's internal configuration function and should be performed on a halted system. By default, Backup writes and Restore reads files with the extension ".500".

Protect Menu

The Protect Menu allows access to the configuration lock and write protection lock bits.

The **Lock Configuration** function sets bit 1 of register 8192 in the CAM500. When this bit is set, the module is locked so that it will not respond to configuration commands until this bit is cleared. The **Clear Configuration** function clears bit 1 of register 8192 and allows the module to accept configuration commands.

The **Set write protect** function sets bit 2 of register 8192 in the CAM500. With bit 2 set, the module will not respond to Store or Delete operations. The **Clear write protect** function will clear bit 2 and allow Store and Delete operations to function.

Local Registers

The following table describes the map of registers available in the CAM500 module.

Table 8-1 CAM500 Register Map

Register Number	Description	Valid Values
1	Number of Files	1..292
2	Baud Rate	0..15 (see Table 8-4)
File 1 Block	File 1	
File 2 Block	File 2	
	...	
Last File Block	Last File	
8192	Module Protection	bit 1 - Configuration Lock bit 2 - Write Protection

The block of Registers needed for each File are arranged according to the following format:

Table 8-2 File Register Block

Register Number	Description	Valid Values
File Register 1	Length of Record	0..2039
File Register 2	Number of Windows	1..16
File Register 3	Maximum Record Number	0..65535
File Register 4	Length of Key	0..2039
Window Block 1	Window 1	
Window Block 2	Window 2	
	...	
Last Window Block	Last Window	

Each block of registers for a Window is arranged as follows:

Table 8-3 Window Register Block

Register Number	Description	Valid Values
Window Register 1	Status Bits	See Table 8-5
Window Register 2	Record Number	0..65535
Window Register 3	First Record Image Register	Any Value
Window Register 4	Record Image	Any Value
...	...	
Next to Last Window Register	Last Record Image Register	Any Value
Last Window Register	Command Bits	See Table 8-5

The baud rate of the CAM500 front port is controlled by setting the correct value in the baud rate register of the rack image table of the CAM500. The values are as follows:

Table 8-4 Baud Rate Values

Baud Rate Register Value	Baud Rate
3	134.5
4	150
5	300
6	600
7	1200
8	1800
9	2400
10	3600
11	4800
12	7200
13	9600
14	14,400

The values for the Command Register and Status Register are listed in Table 8-5.

Table 8-5 Command and Status Bit Listing

Bit	Command Register	Status Register
1	Erase All	Erase All Done
2	Delete by Key	Delete by Key Done
3	Delete by Record Number	Delete by Record Number Done
4	Store by Key	Store by Key Done
5	Store by Record Number	Store by Record Number Done
6	Store by Next Record Number	Store by Next Record Number Done
7	Retrieve by Key	Retrieve by Key Done
8	Retrieve by Record Number	Retrieve by Record Number Done
9	Retrieve by Next Record Number	Retrieve by Next Record Number Done
10	Retrieve by Previous Record Number	Retrieve by Previous Record Number Done
11		Not Found
12		Found
13		Full
14		Empty
15	Retrieve Multiple Records	Retrieve Multiple Records Done

Connector Pinouts

RS-422 Port (DE9S with slide lock posts)

- 1 TX- transmit data (inverted) from CAM500 to SY/MAX device
- 2 TX+ transmit data (noninverted) from CAM500 to SY/MAX device
- 3 RX- receive data (inverted) from SY/MAX device to CAM500
- 4 RX+ receive data (noninverted) from SY/MAX device to CAM500
- 5 +5 volts for NR&D SC406 smart cable
- 6 +5 volts for NR&D SC406 smart cable
- 7 Logic ground and +5 volt return.
- 8 Logic ground and +5 volt return.
- 9 Shield ground. AC coupled to the chassis.

Glossary

In this manual, the terms "bit" and "contact" are used interchangeably.

Register

The 16 bit basic unit of storage both in the file memory and in the rack interface area.

Record

A group of a user-defined number of registers that hold data determined by the application.

Record Length

The number of registers in a record, including the key field if used.

Record Number

The slot number, ranging from 0 to the maximum record number, where a record is found. Also the Record Number register, the second register in a window.

Key

A user-defined number of registers at the beginning of a record that can be used to identify this record for store, retrieve, and delete operations.

Key Length

The number of registers within a record that are used as the search key.

Window

A group of records appearing in the rack interface and used to control file access operations including the Status Register, Record Number Register, Record Image Area and the Command Register. A window has the same number of registers as a record plus three.

Record Image

A copy of an application defined record in a window. The third register of each window is the start of the Record Image.

File

A set of data storage slots each of which can hold zero or one records.

Slot

A position in a file to hold a record. A slot is filled when a record is inserted in it by a storage operation. A slot becomes empty when a delete operation acts on it. Each slot requires the number of registers defined by the Record Length for the file plus one register used internally.

Maximum Record Number

The maximum value the record number can be assigned. This number plus one is the number of slots reserved in a file.

Command Register

The last register in a window. The PLC (or other accessor) sets a bit in the Command Register to request an operation be performed.

Status Register

The first register in a window. The CAM500 sets bits in this register to indicate completion of an operation and, depending on the operation performed, the result of the operation.

Rack Interface

A group of up to 2,048 registers which are accessed by the PLC through its rack addressing capabilities. The Rack Interface registers are used for communication between the CAM500 and the PLC (or other devices) requesting operations. The Rack Interface includes two registers with global effect, four registers defining each file, and three plus the length of a record registers for each window. These registers may also be accessed through the front panel port.

Module Register

The register address of a register in the Rack Interface. When accessed from the front port, these registers are addressed 1 through 2,048. From the rack backplane, PLC register addressing must be calculated to determine the PLC's address for these registers.