JRC Research Report No. 90-18
June 1988 through May 1990

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Commands Specification

Grant NAG8-720

Volume 3 of 4 Volumes

Prepared by

James W. McKee
University of Alabama, Huntsville
Huntsville, Alabama 35899
(205) 895-6257

Prepared for

Norma Whitehead
EB12
NASA/MSFC
Huntsville, Alabama

Table of Contents

1. Purpose

This volume contains the specification for the command language for the AMPS system. The volume contains a requirements specification for the operating system and commands and a design specification for the operating system and commands.

2. Introduction

The operating system and commands sits on top of the protocol. (See volume 2 for a description of the protocol.) These commands are an extension of the present set of AMPS commands in that the commands are more compact, allow multiple sub-commands to be bundled into one command, and have provisions for identifying the sender and the intended receiver. The commands make no change to the actual software that implement the commands.

The operating system is essentially a round robin loop that waits for a packet from the protocol. When the operating system receives a packet, it calls the approbate function to perform the desired operation. If data is to be sent back, the operation system sends a message to the protocol when the out going packet is ready.

3. Operating system / Protocol interface

The protocol interfaces with the operating system through mailboxes as shown in figure 1. There are three mailboxes: new mailbox, old mailbox, and command mailbox. Address of data buffers are passed in these mailboxes. All three mail boxes function in a similar manner.

The command mailbox is used to pass the address of the next command for the operating system to process. The protocol places the address of the buffer containing the command in the mailbox. The MSBC1 checks the command mailbox for a non-zero address and when one is detected, handles the command. When the MSBC1 has finished with the command buffer it places a zero address in the mailbox. The protocol check the command mailbox for a zero address and when one is detected, places the address of the next buffer containing a command in the mailbox.

The MSBC1 uses the old mailbox to return the to the protocol the address of buffers. If the SBFg is set then the buffer contains a packet, if the SBFg is not set then the buffer is no longer needed. The MSBC1 users the new mailbox to get a buffer in which to place out going data or messages.

Figure 2 shows the power up initialization relationship between the protocol and the MSBC1.

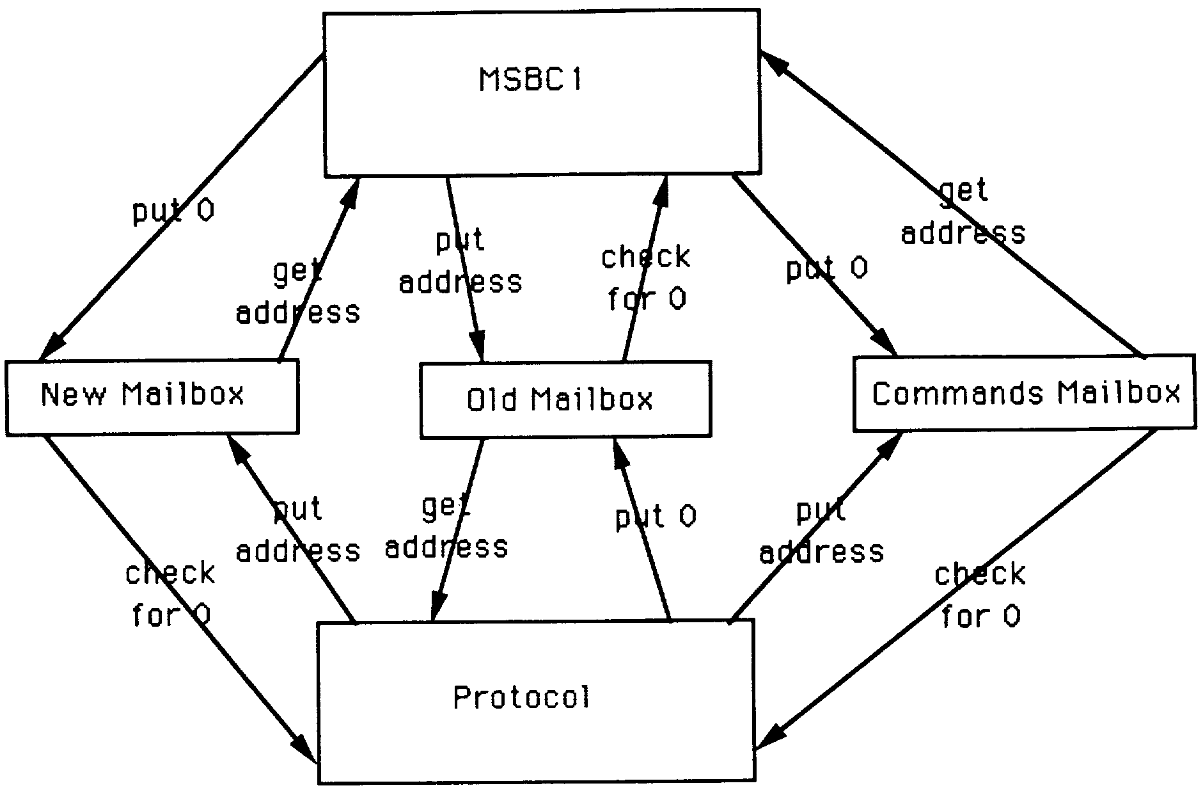

Figure 1 MSBC1 / Protocol mailbox block diagram

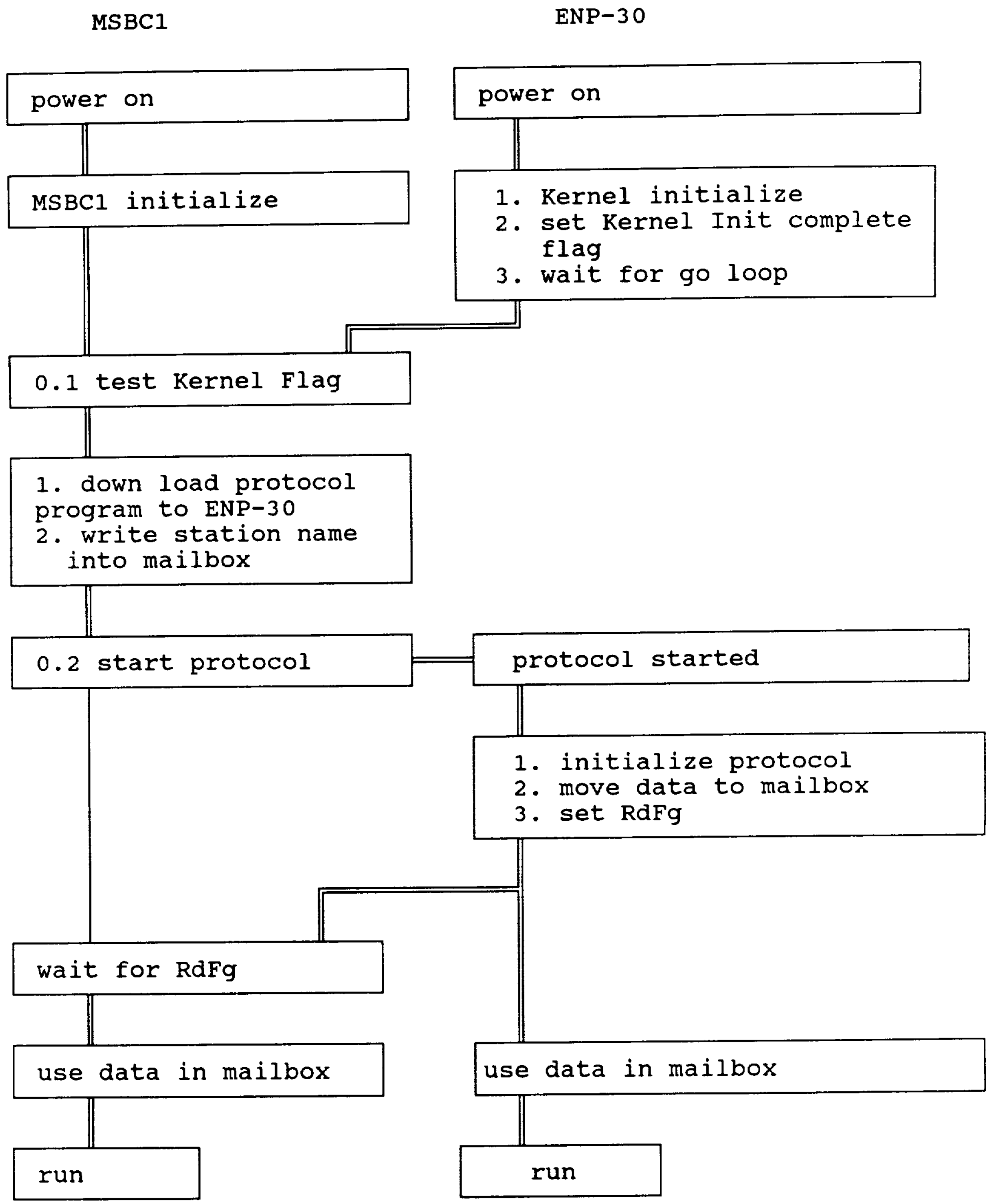

Figure 2 System Initialization Sequence

4. Operating System Command Specification
(note: for the definition of terms see volume 2, protocol)

4.1. Operating system and Command requirements specification

4.1.1. The operating system will handle the scheduling of tasks and functions in each node and will interpret and process commands.

4.1.2. The operating system must insure that commands are executed in the order they are received and must be able to queue a set of commands.

4.1.3. This network will be a master-slave network. The EPSC will be the master and will send commands to all of the other nodes. Each of these nodes will respond to the commands from the EPSC. The other nodes are not allowed to send commands between themselves.

4.1.4. The following is a list of needed functions.

4.1.4.1. Request data -- enables a station other than the EPSC to request data from any other station. The request data packet will be sent to the EPSC. The EPSC will send out a command to the appropriate station for the data. The requesting station will listen on the network. When the data comes to the EPSC, the requesting station will also receive the data.

4.1.4.2. Set switch -- sets the switches in the selected remote power controller RPC, to the state in the command (make before break mode). The states of the switches for a RPC will be coded into one byte. Zero indicates open; one indicates closed. Bit zero is used for line one; bit one is used for line two; and bit two is used for line three, if present. The order of the RPC's will be by increasing number, lowest number first.

4.1.4.3. Read switch -- reads the state of the switches in the selected RPC and sends the data to the requesting station. The states of the switches will be coded into one byte. Zero indicates open; one indicates closed. Bit zero is used for line one; bit one is used for line two; and bit two is used for line three, if present. The order of the RPC's will be by increasing number, lowest number first.

4.1.4.4. Read LC diode temperature -- reads the temperatures of the selected diodes in the selected RPC and sends the data to the requesting station. The temperature will be in degrees Centigrade and will range between -50 deg.C and 200 deg.C in increments of one degree. The data will be in the form of an unsigned number one byte long. The order of the selected temperature data for each RPC will be: heat sink temperature, line one diode temperature, line two diode temperature, and line three diode temperature. The order of the RPCs will be by increasing number, lowest number first.

4.1.4.5. Read LC power -- reads the current and voltage of the selected RPC and sends the data to the requesting station. The current will be the raw data produced by the A/D and will be in the form of two byte number. The voltage will be the raw data produced by the A/D and will be in the form of two byte number. The order of the data for each RPC will be voltage first then current. The order of the RPC's will be by increasing number, lowest number first.

4.1.4.6. Read battery voltage -- reads the voltages of the selected battery cells and sends the data to the requesting station. The voltage will be measured in Volts and will range between -0.5 V and 2.0 V in increments of 0.01 V for individual cells. The data will be in the form of an unsigned number one byte long. The voltage for the groups of cells will be the raw data produced by the A/D and will be in the form of two byte number. The order of the data for each RPC will be by increasing cell number, lowest cell first, cell groups last. The order of the RPC's will be by increasing number, lowest number first.

4.1.4.7. Read battery temperatures -- reads the temperatures of the selected sensors on the battery and sends the data to the requesting station. The temperature will be in degrees Centigrade and will range between -50 deg.C and 200 deg.C in increments of one degree. The data will be in the form of an unsigned number one byte long. The order of the selected temperature data will be by increasing sensor number, lowest number first.

4.1.4.8. Read PS power -- reads the selected voltage and currents in the PS and sends the data to the requesting station. The current will be the raw data produced by the A/D and will be in the form of two byte number. The voltage will be the raw data produced by the A/D and will be in the form of two byte number. The order of the selected data will be voltage, solar array current, battery current, load current, line one current, line two current, and line three current.

4.1.4.9. Switch data -- updates the switch data array in which the current state of the switches of the LCs are stored. The switch data array will be 80 bytes long (10 bytes for each of 8 possible LCs). Each byte contains the status of the switches for a RPC. A zero indicates off; one indicates on. Bit zero corresponds to line one, bit two to line two, and bit three to line three.

4.1.4.10. LC diode temperature -- updates the LC diode temperature array in which the current temperatures of the diodes of the LCs are stored. The LC diode temperature array will be

248 bytes long (31 bytes for each of 8 possible LCs). The order and value of the data for each LC is the same as in the read LC diode temperature.

4.1.4.11. LC power -- updates the LC power array in which the voltage and current of each RPC of the LCs is stored. The LC power array will be 160 bytes long (2*2*10 bytes for each of 8 possible LCs). The order and value of the data for each LC is the same as in the read LC power.

4.1.4.12. Battery voltage -- updates the battery voltage array in which the voltages of all the battery cells are stored. The battery voltage array will be 1792 bytes long (168 + 14*2*2 = 224 bytes for each of 8 possible PSs). The order and value of the data for each PS is the same as in the read battery voltage.

4.1.4.13. Battery temperature -- updates the PS temperature array in which the current temperatures the groups of battery cells are stored. The PS temperature array will be 96 bytes long (12 bytes for each of 8 possible PSs). The order and value of the data for each PS is the same as in the read PS temperature.

4.1.4.14. PS power -- updates the PS power array in which the voltage and currents of the power simulator are stored. The PS power array will be 56 bytes long (7 bytes for each of 8 possible PSs). The order and value of the data for each PS is the same as in the read PS power.

4.2. Operating System and Command Design Specification

4.2.1. Functions will be invoked by commands in the command buffer stack, CBSk. Once invoked a function will run to completion before returning control to the operating system.

4.2.2. The implementation of the required functions is broken down into the implementation of functions and the implementation of the commands that communicate between functions.

4.2.2.1. Functions

The following is a list of the functions required, their format, what each function does, and the name of the command that invokes the function. Buffers containing commands are queued on the CBSk. CBSOt will point to the next buffer to be processed. The CNFd in the selected buffer will be examined for the name of the command. If the command referenced by CBSOt is not recognized, the command will be ignored and the buffer pushed onto the IBSk. If the command name is recognized, it will invoke a function. If the function creates a command, the function will create the command in the buffer that invoked the function. If the function creates a command, the last thing the function will do is to check the SBFg. If SBFg is a one the function waits until SBFg goes to zero. The function will then move the address of the buffer to SBAd and set the SBFg. If the function does not create a command, the buffer referenced by CBSOt will be placed on the IBSk when the function is complete.

Functions expect any needed data to be in the buffer. If a function generates any data, the data will be stored in the buffer. For functions that use the selection field, SFd, of the command, the items to be selected are indicated by the bit settings in the SFd in the command. A one indicates that the corresponding item is selected, a zero indicates that the item is not selected.

4.2.2.1.1. Request data function -- enables a station other than the EPSC to request data from any other station. (The requesting station will send a command to the EPSC with the negative of the desired command.) EPSC converts the command (CNo) to a positive number. The EPSC searches the ANNFds of the ANASk for the destination address that corresponds to the content of the SNFd and places the address in the DAFd. The SBFg is checked and when zero, the buffer address is placed in SBAd, and the SBFg is set to one. Invoked by the request data command.

4.2.2.1.2. Set switch function -- sets the switches in the selected remote power center, RPC, to the state in the command (make before break mode). The RPCs to be selected are indicated by the bit settings in the SFd in the command. RPC1 corresponds to bit 0 through RPC10 corresponds to bit 9. The bytes of data that follow SFd contain the desired switch settings, one byte per each selected RPC. The SFd is parsed, and as each bit set to one is found, the corresponding switch byte is examined. (Bit zero is used for line one; bit one is used for line two; and bit two is used for line three, if present.) For each one in the byte the corresponding switch is closed. Then for each zero in the byte the corresponding switch is opened. Then the buffer address is pushed on the IBSk. Invoked by the set switch command.

4.2.2.1.3. Read switch function -- reads the state of the switches in the selected RPCs and sends the data to the requesting station. The RPCs to be selected are indicated by the bit settings in the SFd in the command. RPC1 corresponds to bit 0 through RPC10 corresponds to bit 9. The SFd is parsed, and as each bit set to one is found, the switch states in the corresponding RPC is determined. The states of the switches will be coded into one byte. Zero indicates open; one indicates closed. Bit zero is used for line one; bit one is used for line two; and bit two is used for line three, if present. The function generates a "switch data command" in the buffer. The contents of the SAFd is moved to the DAFd. The switch data command number is placed in the CNFd. The SFd is not changed. The state of the switches is placed in the data area. The number data bytes is placed in the DLFd. The SBFg is checked and when zero, the buffer address is placed in SBAd, and the SBFg is set to one. Invoked by the read switch command.

4.2.2.1.4. Read LC diode temperature function -- reads the temperatures of the selected diodes in the selected RPC and sends the data to the requesting station. The RPCs to be selected are indicated by the bit settings in the SFd in the command. RPC1 corresponds to bit 0 through RPC10 corresponds to bit 9. The SFd is parsed, and as each bit set to one is found, the temperatures of the diodes in the corresponding RPC are determined. The raw 12 bit data is scaled and converted to an unsigned eight bit number. The temperature will be in degrees Centigrade and will range between -50 deg.C and 200 deg.C in increments of one degree. The order of the selected temperature data for each RPC will heat sink temperature, line one diode temperature, line two diode temperature, and line three diode temperature, if present. The function forms a "LC diode temperature command" in the buffer. The contents of the SAFd is moved to the DAFd. The LC diode temperature command number is placed in the CNFd. The SFd is not changed. The diode temperature values are placed in the data area. The number of data bytes is placed in DLFd. The SBFg is checked and when zero, the buffer address is placed in SBAd, and the SBFg is set to one. Invoked by read LC diode temperature command.

4.2.2.1.5. Read LC power function -- reads the current and voltage of the selected RPC and sends the data to the requesting station. The RPCs to be selected are indicated by the bit settings in the SFd in the command. RPC1 corresponds to bit 0 through RPC10 corresponds to bit 9. The SFd is parsed, and as each bit set to one is found, the voltage and current in the corresponding RPC are determined. The raw 12 bit data is placed in a 16 number. The order of the data for each RPC will be voltage first then current. The function forms a "LC power command" in the buffer. The contents of the SAFd is moved to the DAFd. The LC power command number is placed in the CNFd. The SFd is not changed. The voltage and current values are placed in the data area as they are created. The number of data bytes is placed in DLFd. The SBFg is checked and when zero, the buffer address is placed in SBAd, and the SBFg is set to one. Invoked by read LC power command.

4.2.2.1.6. Read battery voltage function -- reads the voltages of the selected battery cells and sends the data to the requesting station. The battery cells are partitioned into 14 modules of 12 battery cells each. The battery modules to be selected are indicated by the bit settings in the SFd in the command. Module 1 corresponds to bit 0 through module 14 corresponds to bit 13. The SFd is parsed, and as each bit set to one is found, the voltage in the corresponding module are determined. The 12 cell voltages are measured. Then the voltage across the first 6 cells is measured. Then the voltage across the last 6 cells is measured. Each of the 12 raw 12 bit battery voltage values is scaled and converted to an unsigned eight bit number. The voltage of a cell will be measured in Volts and will range between -0.5 V and 2.0 V in increments of 0.01 V. The voltage for a battery module will be the raw 12 number stored in two bytes. The function forms a "battery voltage command" in the buffer. The contents of the SAFd is moved to the DAFd. The battery voltage command number is placed in the CNFd. The SFd is not changed. The voltage values are placed in the data area as they are created. The number of data bytes is placed in DLFd. The SBFg is checked and when zero, the buffer address is placed in SBAd, and the SBFg is set to one. Invoked by read battery voltage command.

4.2.2.1.7. Read battery temperatures function -- reads the temperatures of the selected sensors on the battery and sends the data to the requesting station. The SFd is parsed, and as each bit set to one is found, the corresponding temperature is determined. The 12 bit data is scaled and converted to an eight bit number. The temperature will be in degrees Centigrade and will range between -50 deg.C and 200 deg.C in increments of one degree. The function forms a "PS temperature command" in the buffer. The contents of the SAFd is moved to the DAFd. The PS temperature command number is placed in the CNFd. The SFd is not changed. The temperature values are placed in the data area as they are created. The number of data bytes is placed in DLFd. The SBFg is checked and when zero, the buffer address is placed in SBAd, and the SBFg is set to one. Invoked by read PS temperature command.

4.2.2.1.8. Read PS power function -- reads the selected voltage and currents in the PS and sends the data to the requesting station. The SFd is parsed, and as each bit set to one is found, the corresponding voltage or current is determined. The raw 12 bit data is stored in 16 bit number. The order of the selected data will be voltage, solar array current, battery current, load current, line one current, line two current, and line three current. The function forms a "PS power command" in the buffer. The contents of the SAFd is moved to the DAFd. The PS power command number is placed in the CNFd. The SFd is not changed. The voltage and current values are placed in the data area as they are created. The number of data bytes is placed in DLFd. The SBFg is checked and when zero, the buffer address is placed in SBAd, and the SBFg is set to one. Invoked by read PS power command.

4.2.2.1.9. Switch data function -- moves the data in the switch data command into the switch data array, SDAr. The SDAr is an array in which the current state of the switches of the LCs is stored. The SDAr will be 80 bytes long (10 bytes for each of 8 LCs). Each byte contains the status of the switches for a RPC.

A zero indicates off; one indicates on. Bit zero corresponds to line one; bit one to line two; and bit two to line three. The SNFd is examined to determine the starting offset into the SDAr. The contents of the SFd is parsed, and as each bit set to one is found, the corresponding byte in the SDAr is updated with the data in the data section of the buffer. Then the buffer address is placed on the IBSk. Invoked by switch data command.

4.2.2.1.10. LC diode temperature function -- moves the data in the LC diode temperature command into the LC diode temperature array, LCDTAr. The LCDTAr is an array in which the current temperatures of the diodes of the LCs are stored. The LCDTAr will be 248 bytes long (31 bytes for each of 8 LCs). The order and value of the data is the same as in the read LC diode temperature function. The SNFd is examined to determine the starting offset into the LCDTAr. The contents of the SFd is parsed, and as each bit set to one is found, the corresponding bytes in the LCDTAr are updated with the data in the data section of the buffer. Then the buffer address is placed on the IBSk. Invoked by LC diode temperature command.

4.2.2.1.11. LC power function -- moves the data in the LC power command into the LC power array, LCPAr. The LCPAr is an array in which the voltage and current of each RPC of the LCs is stored. The LCPAr will be 320 bytes long (2*2*10 bytes for each of 8 possible LCs). The order and value of the data is the same as in the read LC power function. The SNFd is examined to determine the starting offset into the LCPAr. The contents of the SFd is parsed, and as each bit set to one is found, the corresponding voltage and current values in the LCPAr are updated with the data in the data section of the buffer. Then the buffer address is placed on the IBSk. Invoked by LC power command.

4.2.2.1.12. Battery voltage function -- moves the data in the battery voltage command into the battery voltage array, BVAr. The BVAr is an array in which the current voltages of all the battery cells are stored. The BVAr will be 1792 bytes long (168 + 48 = 224 bytes for each of 8 possible PSs). The SNFd is examined to determine the starting offset into the BVAr. The order and value of the data is the same as in the read battery voltage function. The contents of the SFd is parsed, and as each bit set to one is found, the corresponding bytes in the BVAr are updated with the data in the data section of the buffer. Then the buffer address is placed on the IBSk. Invoked by battery voltage command.

4.2.2.1.13. PS temperature function -- moves the data in the PS temperature command into the PS temperature array, PSTAr. The PSTAr is an array in which the current temperatures the groups of battery cells are stored. The PSTAr will be 96 bytes long (12 bytes for each of 8 possible PSs). The SNFd is examined to determine the starting offset into the PSTAr. The order and value of the data is the same as in the read PS temperature function. The contents of the SFd is parsed, and as each bit set to one is found, the corresponding byte in the PSTAr is updated with the data in the data section of the buffer. Then the buffer address is placed on the IBSk. Invoked by PS temperature command.

4.2.2.1.14. PS power function -- moves the data in the PS power command into the PS power array, PSPAr. The PSPAr is an array in which the voltage and currents of the power simulator are stored. The PSPAr will be 112 bytes long (2*7 bytes for each of 8 possible PSs). The SNFd is examined to determine the starting offset into the PSPAr. The order and value of the data is the same as in the read PS power function. The contents of the SFd is parsed, and as each bit set to one is found, the corresponding byte in the PSPAr is updated with the data in the data section of the buffer. Then the buffer address is placed on the IBSk. Invoked by PS power command.

4.2.2.2. Commands

The following is a list of the commands. All the commands are processed in the same way -- the name of the command invokes a function and the command supplies data to the function. The CBSk contains the stack of command buffers to be processed; CBSOt points to the next command buffer to be processed.

The following table lists each command and shows the command number, CNo, of the command that appears in the CNFd of the command.

Table 7.1 Commands

command	CNo	user name
Set switch command	1	SSCm
Read switch command	2	RSCm
Switch data command	3	
Read LC diode temperature com.	4	RLCTCm
LC diode temperature command	5	
Read LC power command	6	RLCPCm
LC power command	7	
Read battery voltage command	8	RBVCm
Battery voltage command	9	
Read battery temperature com.	10	RBTCm
PS temperature command	11	
Read PS power command	12	RPSPCm
PS power command	13	
Request data command	(-CNo)	RDCm