

051-33

128237

TDA Progress Report 42-92

18.

N88-18795

October-December 1987

# SETI Data Controllers

R. M. Gosline

Radio Frequency and Microwave Subsystems Section

*This article describes three data controllers developed for the SETI project. Two are used primarily for recording and playback of SETI data from the Radio Science Surveillance System (RSSS). The third is used as a SETI station controller for DSS 13.*

## I. Introduction

This article describes the SETI Data Controllers. There are three units, all based on the same hardware design. Variations in the software allow them to be used for somewhat different purposes.

## II. Functional Description

The basic function of the data controllers is to record real-time data on floppy disks and to read it back. The capacity of a single disk is 512,512 bytes. In addition, the unit located in the DSS 13 control room (DR-2) must have the capability of controlling the station by sending sequences of commands through the station controller to the various subsystem controllers.

There are three data controller units, designated DR-1, DR-2, and DR-3. DR-1 and DR-3 are similar (ROM-based) units (except that DR-1 has four disk drives instead of two), but they are used in slightly different ways. DR-1 (located in the RFI van at DSS 13) is used for recording data from the Radio Science Surveillance System (RSSS). DR-3 (located at JPL) is used for playback of data and recording of sequence-of-events (SOE) files to be used at DSS 13. DR-2 (located in the DSS 13 control room) has additional capabilities for station monitoring and control.

Referring to Fig. 1, DR-1 collects real-time data from the RSSS controller and responds to data controller commands, starting with the character “#” embedded in the data stream. Note that all characters in the data stream are recorded automatically (except for command characters), provided that a write enable command has been received. DR-3 (Fig. 3) utilizes the data collection port to input command (SOE) files from the radio science VAX as data. This is because DR-3 is a ROM-based system and the command files cannot be written directly in CP/M format. DR-3 also responds to “#” commands. Real-time data is assembled in blocks of 256 bytes and written to a data disk in a sequential format to allow maximum storage capability on the floppy disk (see Table 1). Individual disk sectors contain a checksum which is transparent to the user. The data blocks may be read back a block at a time and will be preceded by a block identification number and followed by a checksum for transmission integrity.

An important feature unique to DR-1 is a power failure protection system incorporating a power fail interrupt and a non-volatile core memory. The controller will not lose data upon the occurrence of a power failure. In addition, when power is restored, the controller returns to its previous status after inserting a power failure message and count in the data stream. Should a complete restart and memory clear be desired, the operator must press the TEST button within 1 second after restoring power.

DR-2 (Fig. 2) is a RAM-disk-based system capable of supporting both the CP/M and ISIS operating systems. It has the capability of creating, reading, writing, and editing command (SOE) files to control the station through the subsystem controllers installed for the unattended station development. Higher level command capability than that available with the existing station controller is provided to allow automatic repeat of a single command at specified time intervals and looping of command sequences. In addition, a command may be preceded by a time parameter, which will cause the command to not be executed until the indicated time. In normal operation, a sequence of commands is stored in a command buffer. The buffer may be loaded from a disk file, from the DSS 13 VAX via the user port, or from the local terminal.

### III. Hardware Overview

The data controllers are microprocessors assembled from standard Multibus I modules based on the 8080 CPU. They are housed in individual seven-slot chassis with internal power supplies. Each controller has a dual floppy disk drive using Shugart 801-R drives, housed in a separate chassis, and operating in the double density mode.

DR-1 and DR-3 are ROM-based systems using the Intel SBC 80-20/4 single board computer. The program is contained on board in 2716 PROMs. A Zendex ZX-200A disk controller is used to control the disk drives. A 16-kbyte RAM card is provided to allow the disk controller to operate in DMA mode. The 80-20/4 contains a serial port (designated channel A) that is used for sending and receiving data. A serial port module has been added to DR-1 to allow use of a monitor terminal, modem, and printer.

DR-2 contains 64 kbytes of RAM and is a RAM disk-based system. The program must be loaded from disk after booting the CP/M operating system. The National BLC 80/204 single board computer is used as the CPU. This card allows more flexibility in the use of an on-board monitor ROM whose contents can be moved to high RAM and then shadowed by a software command to allow use of standard CP/M or ISIS operating systems. A BLC-534 communications expansion module provides four additional serial ports.

### IV. Software Overview

The software for all controllers is written in the PL/M-80 language, assembled, linked, and located under the ISIS operating system. The load modules are copied to a CP/M disk which can be loaded under the CP/M operating system for DR-2 or burned in PROM for DR-1 and DR-3. Other utilities such as editors can be loaded to facilitate modification or

creation of command files. The software for all three controllers was developed in DR-2. The principles of top-down structured programming were followed.

## V. Command Descriptions

Except for loops, all commands are processed in sequence, i.e., they are not sorted or reordered. Table 2 is a list of allowable commands from the various ports. R/T commands are processed immediately. Command buffer commands are processed immediately if not preceded by a time parameter. If preceded by a time parameter, the command is processed as soon as the following condition exists:

$$\text{Station Time} \geq \text{Command Time}$$

Note that a timed command or a delay command may hold up the execution of the next command, even if it is in the immediate mode.

### A. Subsystem Commands

The subsystem commands are explained in the "Unattended Station Controller Operator's Handbook."

### B. Data Controller "#" Commands

Data Controller commands all contain six bytes and have the form #xxx"CR/LF."

#IN0	Format disk in drive 0 as a data disk. Set current drive to 0.
#IN1	Format disk in drive 1 as a data disk. Set current drive to 1.
#IN2	Format disk in drive 2 as a data disk. Set current drive to 2.
#IN3	Format disk in drive 3 as a data disk. Set current drive to 3.
#INA	Format disks in all four drives (DR-1 only), starting with drive 0 and continuing until encountering an empty drive.
RDA	Read entire data file to printer port, starting with drive 0 and switching drives until no disk is found or a hex E5 character is found. The output does not include a block ID or checksum.
#BL0	Set the block pointer for drive 0 to the value specified in the next four ASCII characters. The value must be in the range 0001-2002 and must contain exactly four characters. Set the current drive to 0.
#BL1	Set the block pointer and current drive for drive 1.

- #BL2 Set the block pointer and current drive for drive 2.
- #BL3 Set the block pointer and current drive for drive 3.
- #RDO Set the current drive to 0. Read block from current drive and increment block pointer by 1. The pointer wraps around from 2002 to 0001. If the pointer wraps around, the current drive switches to 1.
- #RD1 Set the current drive to 1. Read block from current drive and increment block pointer. If the pointer wraps around, the current drive switches to 2.
- #RD2 Set the current drive to 2. Read block from current drive and increment block pointer. If the pointer wraps around, the current drive switches to 3.
- #RD3 Set the current drive to 3. Read block from current drive and increment block pointer. If the pointer wraps around, the current drive switches to 0.
- #EOF Fill any space in the current data block with hex E5 characters and write it to the current drive. Increment the block pointer as done in the #RDO and #RD1 commands.
- #RST Reset the data controller program and memory to the initial condition after boot or load.
- #STR Return data controller status.
- #WE0 Enable drive 0 to allow write operations.
- #WE1 Enable drive 1 to allow write operations.
- #WE2 Enable drive 2 to allow write operations.
- #WE3 Enable drive 3 to allow write operations.
- #WD0 Disable drive 0 write and format operations.
- #WD1 Disable drive 1 write and format operations.
- #WD2 Disable drive 2 write and format operations.
- #WD3 Disable drive 3 write and format operations.
- #PTA Print all data in RDA format.

**C. DR-2 Control Commands**

- START Begin processing command buffer (run mode).
- STOP Stop processing command buffer.
- DISPLAY [n] Display 24 lines from line *n* of command buffer. Move command line pointer to

line *n*. If *n* is omitted, it displays from current command line.

- DELETE [n] Delete *n* lines in command buffer from the current pointer. If *n* is omitted, one line is deleted.
- INSERT Insert lines before cursor until ESC character.
- "ESC" Terminate INSERT.
- LOOP *n* Repeat from here to ENDLOOP *n* times. Nesting is not permitted. May not follow an RPT*n* command.
- ENDLOOP Signifies end of loop.
- DELAY *n* Wait *n* seconds before processing next command.
- HELP [n] Help menus.
- LOAD file Load command file from CP/M disk. Prompt for correct disk. The maximum file size is 32 kbytes. Prompt for correct disk type.
- SAVE file Write command file to disk. Not permitted in run mode. Prompt for correct disk type.
- RPT*n* *S* Repeat next command at interval *S* (seconds). The maximum allowable is 65535. The command may be canceled with RPT*n* 0 (*n* = 0, 1, . . . , 9). Nesting is not allowed for the same *n*.
- TOFF *T* Add a positive time offset *T* to all command times and data times as they are processed until stop command. *T* is entered in day/hour/minute/second format or the character @ for current station time. The day result is not corrected for a value in excess of a year. The day total and carry may exceed 365. The command in the buffer is not changed, but the disk data and terminal display include the offset.
- DATON Start writing data to disk.
- DATOFF Stop writing data to disk.
- STAIUS Report the current command buffer status.
- CPM2DAT file Copy a CP/M file from drive 0 to an empty data disk on drive 1, then execute an automatic #EOF.
- DAT2CPM Copy a data disk on drive 0 to CP/M disk on drive 1. The file will be broken up into

32,768 byte CP/M files or at the first hex E5 character. The file name will be taken from characters 2 through 9 of the first data block, with the file extension assigned numerically starting with 001. If the CP/M disk does not contain enough room

Y  
N

for the next file, a prompt is issued for another disk on drive 1.

Affirmative response to a query.

Negative response to a query.

**Table 1. Data block format**

Byte	Contents
1-4	Block identification number—must be in the range 0001-2002 and contain only ASCII numerals
5-260	256 bytes of ASCII data containing only printable ASCII characters
261-262	Checksum formatted as 2 hex ASCII bytes (high-order first)*

\*The checksum is computed by performing an "exclusive OR" operation over the first 260 bytes of the block. The resultant byte is split into two hex ASCII digits. For example, a checksum of 91 (hex 5B) results in the ASCII characters "5" and "B" for characters 261 and 262, respectively.

**Table 2. Allowable commands**

Command	Channel A	Channel E	Channel F	Command buffer
#INn	X	X	X	X
#INA	X	X	X	X
#RDA	X	X	X	X
#BLn	X	X	X	X
#RDn	X	X	X	X
#EOF	X	X	X	X
#RST	X	X	X	X
#STR	X	X	X	X
#WEn	X	X	X	X
#WDn	X	X	X	X
#PTA	X	X	X	
START		X	X	
STOP		X	X	X
DISPLAY [N]		X	X	
DELETE [N]		X	X	
INSERT		X	X	
"ESC" (Hex 1B)		X	X	
LOOP N				X
ENDLOOP				X
DELAY N				X
HELP [N]		X	X	
LOAD file		X	X	X
SAVE file		X	X	X
RPT[N] S				X
TOFF T		X	X	X
DATON		X	X	X
DATOFF		X	X	X
STATUS		X	X	X
CPM2DAT file		X		
DAT2CPM		X		
UON		X		
UOFF		X	X	X
Y		X	X	X
N		X	X	X

n = 0, 1, 2, 3.

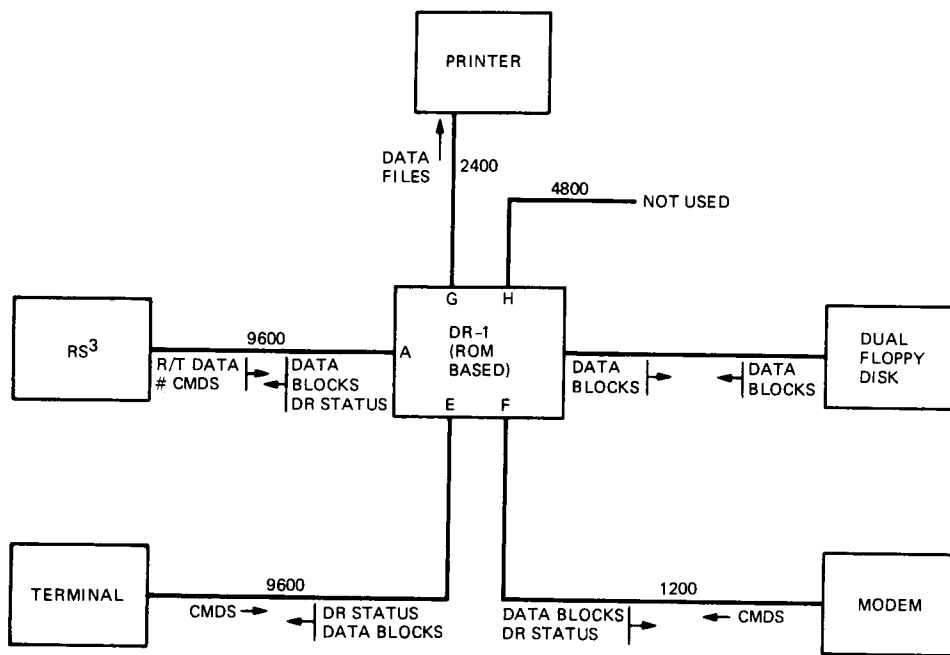
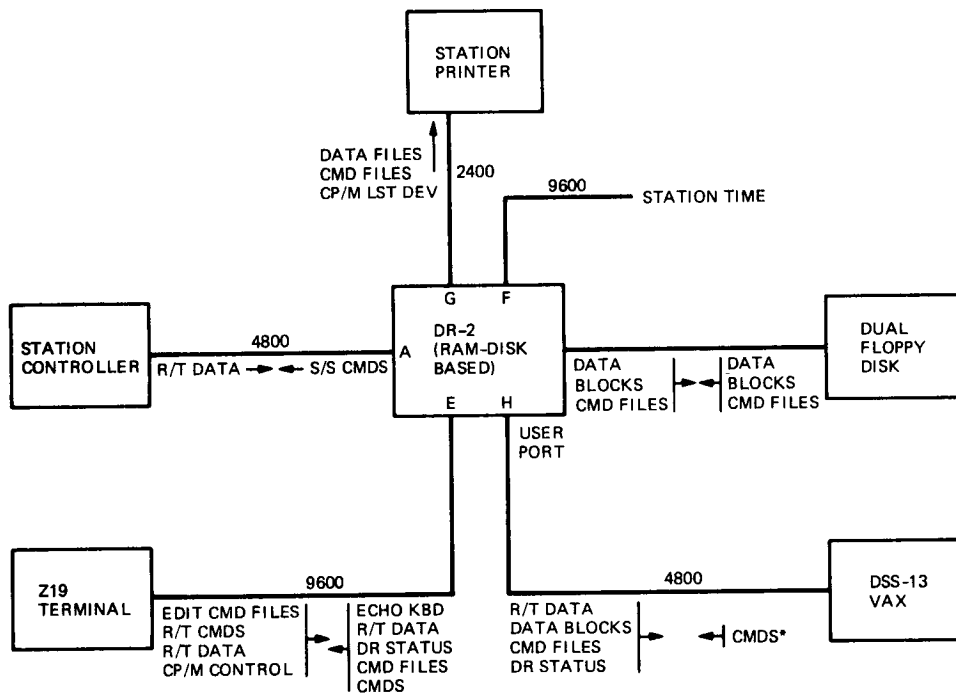


Fig. 1. Functional block diagram of DR-1



\*ALLOWS TRANSFER OF COMMAND FILES WITH INSERT COMMAND

Fig. 2. Functional block diagram of DR-2

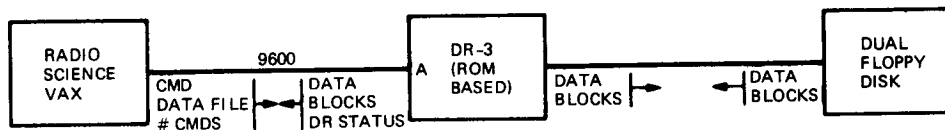


Fig. 3. Functional block diagram of DR-3