Venus Station Automation: Communications Link

A. Zygielbaum
Communications Systems Research Section

A highly error-resistant communications system has been developed in support of the demonstration of the remote automatic control of the Venus Station (DSS 13). This article describes in general the computer communications software used to connect the JPL control terminal to the Venus Station, and describes in some detail the software used to drive the teletype line which is part of the link.

I. Introduction

Remote automatic control of the Venus Station (DSS 13) has recently been demonstrated (Ref. 1). During this demonstration, data from several pulsars were acquired by a Sigma 5 computer at JPL. The system configuration at the Venus Station is an outgrowth of that used in a demonstration of local automatic control of that station (Ref. 2). The system configuration at JPL is similar to that used previously to monitor the performance of the Mu-I ranging system at DSS 14 (Ref. 3). This article describes in general the computer communications software used to connect the JPL control terminal and Venus Station, and describes in some detail the highly error-resistant software used to drive the teletype (TTY) line which is part of the link.

At the Venus Station, an XDS 930 is used as the main control computer. It is connected to two XDS 910 computers by high-speed links known as D/R units (Refs. 2 and 4). Currently, the only available data path from the Venus XDS 930 to JPL is via a 10 character-per-second teletype line. Using standard XDS teletype buffers, the 930 is connected to an XDS 920 at JPL, which, in turn, is connected to a Sigma 5 computer via another D/R link. The software which drives this link has been previously described (Ref. 5).

II. Definition of Task

The primary task was to implement a simple to use, reliable communications link between the Venus XDS 930 computer and a Xerox Sigma 5 computer at JPL. The topology of the path between these machines in shown in Fig. 1. To facilitate user programming, identical Fortran-callable user interfaces were defined for both ends of the link. Extensive error checking and retransmission capability were required for reliability.

Three software design and development efforts were embodied in the primary task. A teletype handler was needed to link the XDS 930 at the Venus Station and the XDS 920 at JPL. To extend the connection, a "switcher" program had to be written for the XDS 920 to interface the teletype handler and the D/R software controlling the link to the Sigma 5. Finally, Fortran-callable subroutines were required for the Sigma 5 to drive the D/R software, and provide a user interface identical to that in the Venus XDS 930.

III. User Interface

The teletype handler appears as the collection of Fortran-callable subroutines shown in Table 1. Identical software is used at each end of the link. As a result, both the Venus XDS 930 user and the XDS 920 "switcher" utilize the same subroutine calls to access the teletype lines. These same calls are made available in the Sigma 5 by the transparent combination of the XDS 920 switcher and the Sigma 5 interface package.

Two logical full duplex channels are provided by the communications link. Messages conveyed by the link are blocked to facilitate retransmission. The user can assign a transmission priority of 1 to 15 (1 is highest) to either channel. The channel with the highest priority can preempt the other channel. If they are given equal priorities, data blocks on the channels are interleaved so that the traffic on both channels appears to be transmitted simultaneously.

Sending a message is equivalent to transferring a core image from one end of the link to the other. The respective users must supply the buffers from which these data are taken and into which these data are placed. At either end, two transmit flags control transmission on both channels while two receive flags control reception.

To supply a buffer and begin transmission, the SNDMSG(LOC,CHN,PRY,RTRY,CHCNT) subroutine is used. The parameters are: LOC, the first location of the buffer; CHN, the channel number (1 or 2); PRY, the channel priority; RTRY, the number of times any particular block of the message is retransmitted before a line error is declared; and CHCNT, the number of 4-bit bytes in the buffer. A buffer is designated for reception with the SETRX (CHN,LOC,CNT) subroutine. Once again, LOC and CHN are the first location in the buffer and the channel number, respectively. The size of the buffer is given as CNT computer words to prevent data overflows.

It would be unacceptable to transmit a message if a receive buffer had not been provided (i.e., if SETRX had not been executed at the receiver). Coordination of the two sides of the link for transmission is provided by transmit flags. The mechanism for coordinating flags at both ends of the link will be discussed later.

These flags are accessed by the subroutine TXSTAT (CHN,FLG), where CHN is the channel number and FLG is a dummy variable into which the flag value is placed. The transmit flag normally has the value 1, indicating that a receive buffer for that channel is available at the remote end of the link and that the channel is free. A value of 0 implies that SNDMSG has been executed and, therefore, the channel is busy. If the transmission is successful, the flag remains busy, 0, until a new receive buffer is supplied via SETRX. When the receiving user supplies the buffer, the transmit flag again becomes 1. If the transmission is unsuccessful, that is, if a particular block has been retransmitted RTRY times, then the flag is set to -1. No action is required by the receiver user as the receiving buffer is still considered empty.

Data reception is coordinated through receive flags. A receive flag is accessed by the subroutine RXSTAT (CHN,FLG). Again, CHN is the channel number and FLG becomes equal to the receive flag. Once a receive buffer has been supplied, the receive flag remains 1 until a message has been succesfully received. Note that there is no indication to the receiving user that a message receipt is in progress or that an unsuccessful transmission was attempted. If a correct message is received, the receive flag is set to 0. It will not be reset to 1 until SETRX is called to supply a new buffer.

A utility subroutine MSGCNT(CHN,CNT) is provided to indicate the number of 4-bit bytes received in a message. This subroutine will return the count as the dummy variable CNT for channel CHN provided that the receive flag is 0, that is, if a message has just come in. If the flag has any other state, MSGCNT will return -1.

Sample receiver and transmitter programs are shown in Fig. 2. The transmit routine first executes a busy wait until the channel is free. Upon exiting this loop, it starts the transmission of the buffer IRRAY. Coincidentally, the receiver waits for a message to come in. Upon receipt, MSGCNT is used to determine the incoming byte count. Execution of SETRX supplies a new receive buffer and, consequently, frees the channel.

System initialization is facilitated through the TINIT subroutine. This subroutine may be called by either user at any time. Its effect is to initialize all flags and to terminate all traffic activities. At both ends of the link, the transmit flags are set to 0 and the receive flags are set to -1. This receive flag state alerts the remote user to the fact that the link has been reinitialized and indicates that new receive buffers must be supplied as well as any interrupted transmissions restarted. When SETRX is called, the transmit flag at the opposite side of the link will be set to 2, initialization complete. Concurrently, the local receive flag will become 1, i.e., no message. A typical intialization sequence is shown in Fig. 3. The CALL TXSTAT checks for initialization complete. RXSTAT is called to detect and protect against a simultaneous initialization from the other end.

IV. The Teletype Link

Characteristics of the teletype circuit influenced the software design. Preliminary experiments indicated that a particular teletype circuit including the computer buffers, microwave, land lines, and associated equipment has a per character error rate of 10^{-3}. Two error checking schemes are used to ensure reliable communication. First, information to be transmitted is divided into four-bit bytes and encoded into the first four bits of the teletype character. The fifth bit of the teletype character is character parity. Second, data are divided into blocks of 120 characters, and the characters within a block are bit-wise exclusive-ored to form a checksum parity character, or longitudinal parity check. If an error is detected, the block is retransmitted. A block size of 120 characters was selected as small enough to minimize the probability of retransmission and yet large enough to prevent channel saturation by overhead bookkeeping characters.

Two types of teletype characters are designated, data characters and utility characters. Constrained to be of even parity, data characters contain text and block overhead information. Odd parity characters are used as utility signals called "interhandlers" to control the interlocked transmission of blocks and synchronize the initialization of the system. The use of distinguishing parity permits the interleaving of interhandlers and text with no ill effect.

The interhandlers consist of two odd parity characters transmitted consecutively twice. The five-bit characters are each denoted by two octal digits. Table 2 relates these numbers to teletype characters. For instance, the interhandler which notifies the remote teletype program to initialize is 37/01/37/01. Any combination of 37 and 01 occurring within 4 characters will cause the interhandler to be recognized (e.g., the message 47/01/37/02 is valid). Since the interhandlers are selected to be of at least distance 2, this technique yields simple error correction. The interhandlers are summarized in Table 3.

Interhandlers are given odd parity to permit them to be inserted into the text of a message. This prevents delay of up to one block time (13.7 seconds) in the transmission of an interhandler. If an odd parity character is received, its validity as an interhandler character is checked. If it is valid, it is accepted. If it is not valid and message receipt is in progress, an error flag is set to indicate that a character with improper parity was received. If a message is not in progress, the character is ignored.

Whenever data or link parameters are transmitted, they are encoded into four-bit bytes. The data bits become the four most significant bits of a five-bit teletype character, with the remaining bit used to set even parity.

Data are sent in blocks of 120 characters (480 bits). The format of a block is shown in Fig. 4. The start-of-block interhandler alerts the receiver to imminent data. Following immediately, a header information sub-block includes a message number, channel number, block number, and block character count. The message number for each channel is set to 0 during initialization and incremented by one for each message of a channel. It is transmitted as one character, hence it is modulo 16. The channel number identifies whether the block is assigned to channel 1 or 2. To order the message, a block number is initialized as the number of blocks in a message. It is allowed 2 characters (8 bits). As each block of a message is sent, the block number is decremented by one. The final block of a message, therefore, has a block number equal to one. Used in conjunction with the message number, the block number helps keep track of the sequence of blocks and messages as an error detection tool. The character count is 8 bits (two characters) containing the number of characters in the block. "Text" contains the data to be transmitted. End of text is marked by the interhandler 10/25. It is followed by a checksum derived from a character-by-character bit-wise exclusive-or of the text and header information sub-block. Finally, the end of block is designated by 10/34/10/34.

After a block is sent, the transmitter awaits a reply from the receiver before continuing. This reply or handshake is used to synchronize the link and control retransmission. As a block is received, an error flag is set (1) if any particular text or header sub-block character has odd parity, (2) if the count of received characters and the block character count disagree, (3) if the derived checksum parity and transmitted checksum parity disagree, (4) if the block number is out of sequence, (5) if the message number is changed prior to receipt of block number one, or (6) if the end of block is received prior to the end of text interhandler. If the error flag has not been set, the receiver replies to the transmitter with the block-received-OK interhandler; otherwise, block-not-okay is sent. At the transmitter, in the former case, the message is continued or if complete, the transmitter channel remains busy but inactive until the receiver signals that a new receive buffer is available. In the latter case, a parameter set by the user for that channel, designated RTRY, is decremented. If it remains >0, the block is retransmitted. IF RTRY is 0, the user is notified that the channel is in error and the channel is freed.

V. The XDS 920 Switcher

With the teletype link from the Venus Station XDS 930 to the XDS 920 at JPL established, the next step is the path between the XDS 920 and the Xerox Sigma 5. These machines are, fortunately, already interconnected via the D/R high-speed link. Software to support the D/R unit is available and utilized. As this software will be described in a cursory fashion here, the reader is again referred to the article by Layland (Ref. 5) for more detail.

A program to interface between the D/R software and the teletype software described earlier resides in the XDS 920. This program, in combination with the Sigma 5 software, discussed in the next section, appears transparent to the user. Hence, the Sigma 5 user believes he is communicating directly with the teletype handlers. The main tasks, then, for the XDS 920 "switcher" program are to route and reformat message traffic through the D/R link and to extend the teletype transmit and receive flags to the Sigma 5.

Communication with the Sigma 5 is maintained via the aforementioned D/R link software. This software is capable of transferring blocks of 126 8-bit bytes at rates approaching 25 kilobytes per second. Seven logical bidirectional channels are provided. Five of these were used in this implementation. D/R channels 1 and 3 are assigned to teletype channels 1 and 2, respectively, of the teletype handler for traffic from JPL to Goldstone. D/R channels 2 and 4 were assigned to teletype channels 1 and 2, respectively, for Goldstone to JPL traffic. D/R channel 5 was used for "signals" to coordinate and control the XDS 920/Sigma 5 link.

As far as the user is concerned, the teletype handlers can transfer memory blocks of arbitrary size. On the other hand, the D/R unit software is limited to transferring blocks of 126 8-bit bytes. Therefore, the switcher and Sigma 5 software must cooperate by disassembling messages into D/R unit blocks for transmission and by reassembling these blocks into core images upon reception. To maintain compatibility with the teletype link, the Sigma 5 software must be informed of the receive 4-bit character count for the entire message. This is done by inserting the character count into the first two bytes, i.e., 16 bits of each D/R block. Although this information is unnecessarily repeated, the time penalty is negligible. In the opposite direction, the switcher makes use of the SNDMSG subroutine to transmit traffic to Goldstone. The subroutine must be supplied with channel priority, number of retrys and the four-bit character count. This information is supplied by the Sigma 5 user and sent to the switcher program as the first four bytes of each D/R block.

It should be noted that the teletype channel priority scheme is not reflected in the D/R link. This is simply because of the high speed of that link. Whenever a full message is ready to be transmitted from the switcher to the Sigma 5, or vice versa, it is sent on a first-come first-serve basis without regard to priority.

The signals used to coordinate and control the message traffic over the D/R link are given in Table 4. These signals allow the Sigma 5 applications program to direct the teletype subroutines in the XDS 920 and interrogate their status much as if the teletype were directly under its control and not separated by two additional interface programs. The sequence of events in handling this link is identical to that discussed earlier.

VI. The Sigma 5 End

An overriding requirement in the communications link design was that both ends of the link appear identical to the user. The purpose of the Sigma 5 program is to establish an interface between the Fortran-callable subroutines and the D/R link software. This interface provides compatibility with the teletype routines, splits arbitrary-length outgoing data messages into D/R blocks, and concatenates incoming D/R blocks into arbitrary-length messages.

The environment of the Sigma 5 led to an immediate problem. The communications software is a subprogram of the user's program. The D/R link's software is a subroutine of the communications software. While the D/R software is interrupt-driven and therefore runs in real time, the communication subprograms run only when called by the user program. The communications software must, however, chop messages into D/R blocks and "feed" them to the D/R subprograms. In the inverse, the communications software must concatenate D/R blocks into messages. To keep the link running, therefore, either repeated calls to some "keep running" subroutine or some "busy-wait" is required. The former alternative would violate the premise that both ends of the link be identical. The latter could cause possibly hazardous delays to the user program.

To avoid this problem, the interface subroutines are constructed around a FIFO list, the task list. Whenever some subroutine has a pending operation, it places a particular number in the list and executes a return to the calling program. The internal subroutine STACKPUT(N) is used for this operation. To keep the system running, the subroutines TXSTAT and RXSTAT contain calls to STACKGET. STACKGET removes, in turn, the topmost item in the list and transfers control to the designated subroutine. When the subroutine execution is complete, control is returned to STACKGET. This process is repeated until the task list is emptied. Control then returns, as appropriate, to TXSTAT or RXSTAT. Since TXSTAT and RXSTAT are routinely called by a user program, and in fact are usually part of some testing loop, the use of the task list provides satisfactory operation which is transparent to the user.

The communications link is normally operated from the timesharing terminal on the Sigma 5. Because of the design of the link, some time is spent in busy-waiting on the status of the transmit or receive flags. During these periods, it was deemed inappropriate to delay concurrent background or batch processing jobs. On the other hand, TXSTAT and RXSTAT must be repetitively called, as just described, to keep the system going. A subroutine, "ENDSLICE," was added to the link software to allow the user program to return the unused portion of a time-share time slice to the operating system. ENDSLICE will return the slice only if the link is not actively receiving or transmitting, that is, only if repeated calls to TXSTAT or RXSTAT are not necessary.

VII. Conclusion

The communications system has now successfully operated for an elapsed period approaching 100 hours. It is therefore considered "operational" and will continue to be used in the station automation experiment as long as teletype is the primary digital data link with the Venus Deep Space Station.

References

1. Moyd, K., "Remote Automatic Control of DSS 13," in *The Deep Space Network Progress Report 42-30,* Jet Propulsion Laboratory, Pasadena, California.

2. Moyd, K., "Automatic Control of DSS 13," in *The Deep Space Network Progress Report 42-29,* Jet Propulsion Laboratory, Pasadena, California, December 15, 1975.

3. Erickson, D. E., and Layland, J. W., "An Experiment in Remote Monitoring of Mu-Ranging Operation at Mariner Mars 1971 Superior Conjunction," Jet Propulsion Laboratory DSN Progress Report TR 32-1526, Vol. XV, pp. 156–166, Pasadena, California, June 15, 1973.

4. Lushbaugh, W. A., "A Driver/Receiver Unit for an Intercomputer Communications Link," Jet Propulsion Laboratory DSN Progress Report TR 32-1526, Vol. XV, pp. 109-115, Pasadena, California, June 15, 1973.

5. Layland, J. W., "Software for Multicomputer Communications," Jet Propulsion Laboratory DSN Progress Report 42-26, pp. 145–154, Pasadena, California, April 15, 1975.

Table 1. Summary of subroutines

SNDMSG(LOC,CHN,PRY,RTRY,CNT)

Causes a data block to be transmitted.

- LOC —Location of the first word of data
- CHN —Channel 1 or 2
- PRY —Priority 1 to 15
- RTRY—Number of block retransmissions
- CNT —Number of 4-bit bytes of data

TXSTAT(CHN,FLG)

Returns transmit status of channel CHN as one of the following values for FLG:

- 2—Channel initialized
- 1—Message received, channel free
- 0—Channel busy
- −1—Message not received (error), channel free

RXSTAT(CHN,FLG)

Returns receive status for channel CHN as one of the following values for FLG:

- −1—No buffer
- 0—Message in
- 1—No message in

SETRX(CHN,LOC,CNT)

Assigns a receive buffer starting at LOC, CNT words in size, to channel CHN.

TINIT

Initializes the link.

ENDSLICE *** Sigma 5 ONLY ***

Releases unused time slices to the operating system.

MSGCNT(CHN,CNT)

If the receive flag for CHN is 0, CNT will be assigned a value equal to the number of 4-bit bytes just received. If the flag is not 0, then CNT will become −1.

Table 2. Teletype characters

Letters character	Octal code	Figures character	Octal code
Letters	37	Figures	33
A	03	0	26
B	31	1	27
C	16	2	23
D	11	3	01
E	01	4	12
F	15	5	20
G	32	6	25
H	24	7	07
I	06	8	06
J	13	9	30
K	17	–	03
L	22	?	31
M	34	:	
N	14	$	11
O	30	!	15
P	26	&	32
Q	27	#	24
R	12	'	13
S	05	(	17
T	20	)	22
U	07	.	34
V	36	,	14
W	23	;	36
X	35	/	35
Y	25	"	21
Z	21	Blank	00
Bell	05	Space	04
Line feed[a]	02	Carriage return	10

[a]Figures character.

Table 3. Teletype link interhandlers

Interhandler	Teletype characters
Order initialize	37/01/37/01
Initialization complete	37/31/37/31
Block received OK	37/26/37/26
Block received in error	37/23/37/23
Start of block	10/02/10/02
End of text	10/25
End of block	37/34/37/34
Channel 1 buffer available	37/15/37/15
Channel 2 buffer available	37/13/37/13

Table 4. D/R link signals

Signal	Number
Order initialize	0
Initialization complete	1
Channel 1 free	2
Channel 2 free	3
Start message channel 1	4
Start message channel 2	5
End message channel 1	6
End message channel 2	7
Bad teletype data channel 1[a]	8
Bad teletype data channel 2[a]	9

[a]920 to Sigma only.

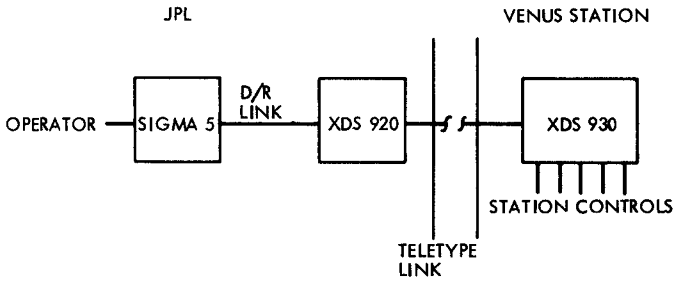


Fig. 1. Communications system

```
      CALL TINIT
3     CALL SETRX (1, LOC1, CNT1)
      CALL SETRX (2, LOC2, CNT2)
1     CALL TXSTAT (1, FLG)
      IF (FLG-2) 1, 2, 1
2     CALL RXSTAT (1, FLG)
      IF (FLG) 3, 4, 4
4     CONTINUE
```

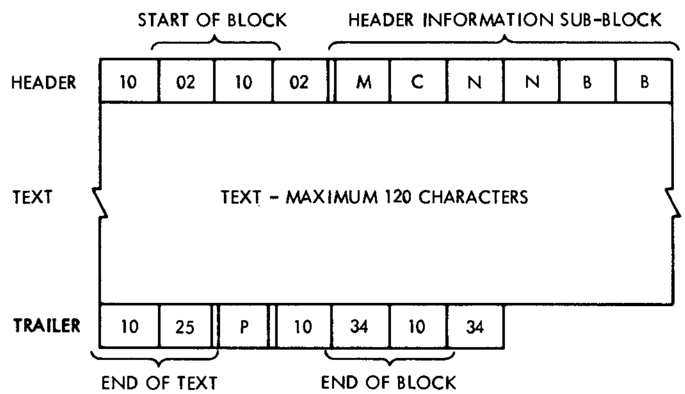
Fig. 3. Initialization sequence

```
TRANSMITTER
        .
        .
1     CALL TXSTAT (1, IFLG)
      IF (IFLG) 3, 1, 2
2     CALL SNDMSG (IRRAY (1), 1, 1, 2, 200)
        .
        .
        .
3     TYPE 4
4     FORMAT ($TRANSMISSION ERROR$)
        .
        .
        .

RECEIVER
        .
        .
1     CALL RXSTAT (1, IFLG)
      IF (IFLG) *, 2, 1
2     CALL MSGCNT (1, CNT)
      CALL SETRX (1, IRRAY (1), 2000)
```

* THIS INITIALIZATION STATE IS DISCUSSED ELSEWHERE

Fig. 2. Sample message transfer programs



M = MESSAGE NUMBER
C = CHANNEL NUMBER
N = 2-DIGIT BLOCK NUMBER
B = 2-DIGIT CHARACTER COUNT
P = LONGITUDINAL PARITY

Fig. 4. Block format