GM347 Manual
for
GM870
Modem Board

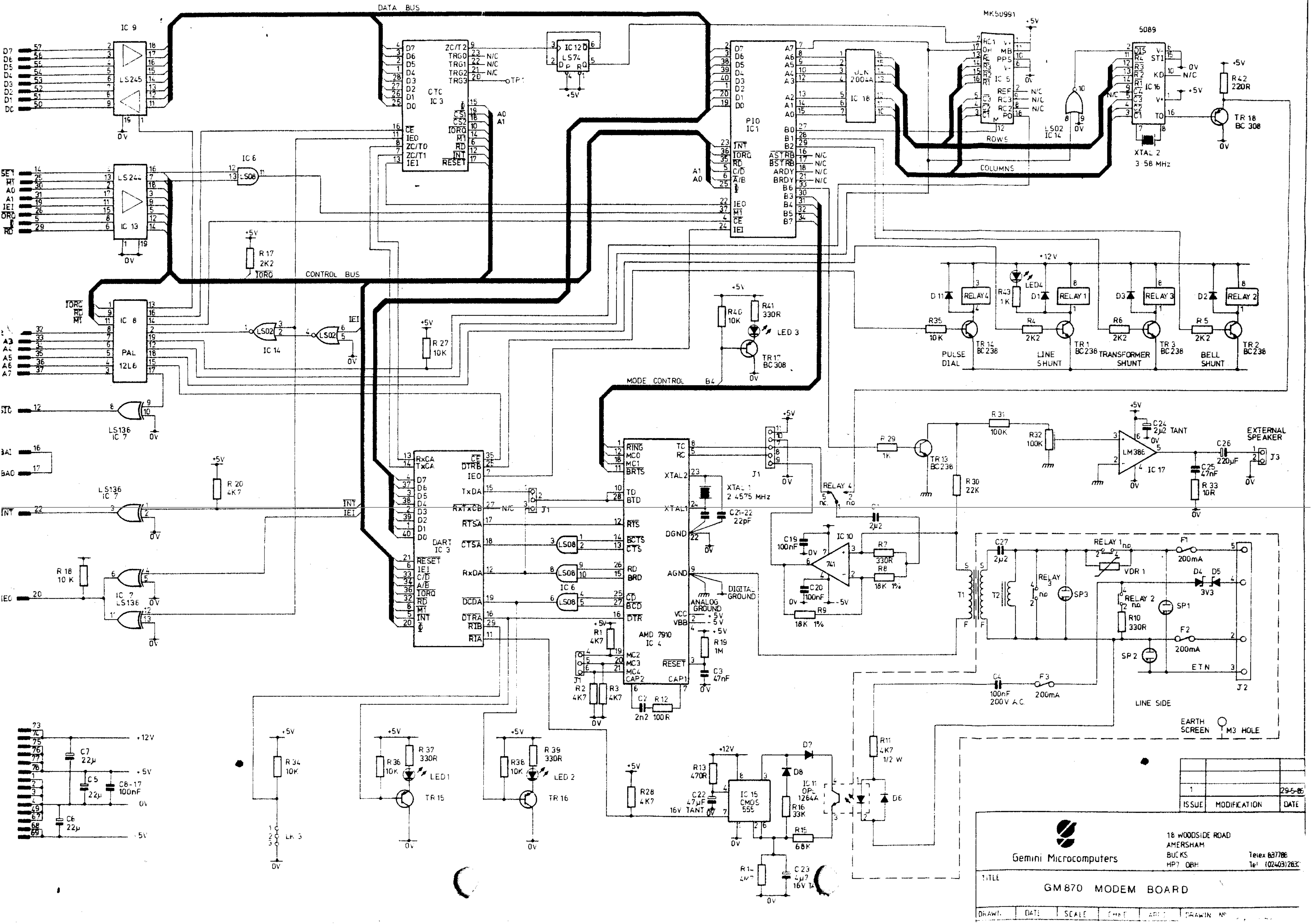
The Gemini MultiBoard Microsystem

MODEM

80-BUS
AUTO-DIAL/AUTO-ANSWER
MODEM BOARD

USER MANUAL

GM347
ISSUE 1
07-06-85



ISSUE	MODIFICATION	DATE
1		29-5-85

Gemini Microcomputers

18 WOODSIDE ROAD
AMERSHAM
BUCKS
HP7 0BH

Telex 837788
Tel (02403)2830

TITLE

GM870 MODEM BOARD

DRAWN | DATE | SCALE | CHKD | APPD | DRAWING No

TABLE OF CONTENTS

Appendices

1. INTRODUCTION

The Gemini GM870 modem is a standard 80-BUS 8" x 8" board allowing CCITT standard communication using the Gemini MultiBoard range. The board has been designed to interface directly to the 80-BUS thus alleviating the need for extra serial ports or the permanent commitment of existing serial ports. The board can be directly controlled from any of the Gemini CPU boards, (811, 813 or 888). The modem is designed around the AMD 7910, "the world chip", and provides low speed data communication capability based on the CCITT standards V.21 (300/300 baud) and V.23 (1200/75 baud) in both originate and answer modes. The Bell System equivalents are also supported. For a more detailed hardware description see section 4

1.1. CCITT standards

V.21	300/300	originate	full duplex
V.21	300/300	answer	full duplex
V.23	75/1200	originate	asymmetrical duplex
V.23	1200/75	answer	asymmetrical duplex

1.2. BELL standards

Bell 103	300/300	originate	full duplex
Bell 103	300/300	answer	full duplex
Bell 202	1200		half duplex

Note: Bell 202 has no back channel as such, only a 5 bit/sec on off signal (387 Hz = on, 0 = off) used for handshaking.

The board has both auto-dial and auto answer capability. The auto dialler will currently only perform loop disconnection operation, since this is by far the most common dialling technique, and further, with the exception of the very few IBM PABX systems (which only support the American "touch tone" system) all U.K. connection systems support it. Unfortunately this means that the board will not perform automatic dialling with any of the completely digital PABX installations. This is because these systems perform dialling using tones which are placed on the line. Loop disconnection dialling however, dials a number exactly how it suggests it does, by repeatedly connecting and disconnecting the phone line. As it can be seen, these two types of dialling are completely incompatible.

Simple modifications can be made to the GM870 to make it MF4 compatible (this is the CCITT version of "touch tone"). To find out how to modify the board to allow MF4 compatibility contact your Gemini dealer.

In order to ensure the present and on-going compatibility of the modem with the PSTN connection arrangements the board uses the "New Plan Socket" connection technique, using a primary socket, or a secondary socket connected to a primary socket. The effect of connecting to a new plan secondary outlet only would be to cause the board to ignore incoming ringing current, and hence render the auto-answer feature useless.

2. INSTALLATION

2.1. General

Carefully unpack your board and inspect it for any obvious mechanical damage. In the event of there being any damage to the board then contact your Gemini dealer at once. **DO NOT** put your board into a system and power it up hoping that the damage is only superficial. If the line side of the board has been damaged then turning it on could well not only further damage your board but could also damage the rest of your equipment and your phone system.

If the board is to be used in a wholly Gemini MultiBoard environment (i.e. there are no boards in the system from other manufacturers) then no checks have to be made before the board can be plugged in. It can be plugged directly into any vacant edge connector. However, if you wish to write your own communications packages which control the board using interrupts then you will have to ensure that the board is in the required place in the interrupt daisy chain. For the interrupt chain to function correctly then there must be no vacant slots left in the motherboard between the modem board and the CPU card.

When inserting the board into the motherboard excessive force should not be required. If any difficulty is experienced it is probably due to the keyway on the bus not lining up with the slot on the edge of the board.

The Gemini GM870 modem board occupies 12 Z80 I/O ports. These ports are located at 80h to 8Ch. Although no Gemini boards are supplied with decodes that coincide with these addresses, you should check that any boards produced by another manufacturer that are resident in your system do not conflict with the modem. Also check any Gemini boards where you have selected an alternative decode address.

2.2. Connection

To ensure the onward compatibility of the modem board and the PSTN, (public switched telephone network) the modem board has been provided with a "NEW PLAN SOCKET" for connection to the outside world. Also provided is a length of cable of the type needed to connect the unit to any six pin "NEW PLAN" socket. The connection is made simpler by the fact that this cable cannot be plugged in the wrong way round. Either end may be plugged into the modem board. If your phone system does not have these new sockets then you can connect the modem board via a spade terminal to plug conversion cable, BT ref 4/502 or 6/502. These cables are not provided by Gemini but can be obtained from RS (Cat. no. 470-156). Alternatively you will have to have some sockets installed before your modem can be used. If you wish to provide a cable of your own length it should be of the type BT ref 6/500 or 4/500. This is a reversing cable where pin 1 of plug A is connected to pin 6 of plug B.

PLUG A	PLUG B
1	6
2	5
3	4
4	3
5	2
6	1

NOTE: 4-way, 6 pin plugs can also be used. However on these plugs pins 1 and 6 are not supported. It is not possible to use 4 pin plugs as they are a different physical shape and they will not mate with the socket on the board.

2.3. Speaker connection

On the top edge of the board there are terminals provided to allow the connection of a low power 8 ohm speaker. This is provided to allow the user to monitor the phone line if he so wish. Also associated with the speaker output there is a variable resistor. This can be used to provide volume control on the speaker output. When the unit is dispatched it will be set at a fairly low level but you are free to adjust it to whatever level you prefer. The speaker output can be controlled by the operating software. For more details on how to control the speaker see the hardware section of this manual.

```
      VOLUME CONTROL     GND   SIGNAL
        .--------.      .-----.
        !________!      ! * * !
  ------------------------------------------------
```

2.4. LED's

Along the top edge of the board there is a row of 4 LED's. These give an indication of the status of the board. They are labelled DTR,CD,ORG and LN.

```
.-------------------.
!                   !
!___________________!
   !_!  !_!  !_!  !_!

   DTR  CD   ORG   LN
```

2.4.1. DTR

This LED gives an indication of whether the modem is ready to receive data from the phone line. When the LED is alight then the board has been initialised to accept incoming data and is in the ready state waiting for data to appear on the line.

2.4.2. CD

This LED shows when there is a valid carrier tone on the line. When it is alight then the board is receiving a valid carrier tone from the other modem. The LED will also light if the modem has been instructed to generate the carrier tone and the tone is now present on the line. Failure of this LED to light could be due to a) there is no tone present on the line. b) the modem is trying to operate in the wrong format. That is, although there may be a carrier on the line it may well not be the one that the board is expecting. The board could be trying to receive data using V.21 protocols and the transmitting board may be set to transmit using V.23 protocols, or vice versa.

2.4.3. ORG

When lit this LED indicates that the board has been set to an operational mode where it is expected to originate the carrier tone and it has placed the required tone on the line. Confusion will occur if it is trying to communicate with a board which is also trying to originate the carrier signal. NOTE When the board is operating in ANSWER mode then the board will still generate a carrier tone, but this time it will be one of a different frequency.

2.4.4. LN

The last LED indicates that the modem board is using the phone line. During auto dialling it will flash as the modem dials the required number. After dialling has stopped the LED will stay on permanantly until the line is disconnected. This LED also lights when the modem takes control of the line when it answers an incoming call when in auto-answer mode.

Situated next to the row of LED's is a stripline connector. Under no circumstances must the jumpers be removed from their factory set condition. **If any of the links are altered from their factory set condition then the board could be forced to operate in a mode that contravenes BT regulations.**

2.5. Fuses

On the line side of the board there are three fuses fitted as a preventative measure. These fuses are of 250V 200mA rating. They are fitted to prevent damage occuring to the board if the telephone exchange accidentally emits voltages which are liable to damage the modem board. Similarly they also prevent your phone system from being damaged if your computer happens to do the same. If you leave your system continually plugged into the phone system you may find it refuses to work after electric storms. If this is the case then check the fuses. Lighting strikes may well have caused transient spikes on the telephone lines, which in turn may have blown the fuses in the modem board. It cannot be guaranteed that these fuses will prevent damage to the board if a direct lightning strike is made to the line. It is recommended that the board is disconnected from the phone line during electric storms. However the board has been designed so that spark arrestors can be fitted to provide further protection if you so wish. For details on what type to fit, and how to fit them contact your Gemini dealer.

3. SOFTWARE

3.1. Software overview

The Gemini modem board is supplied with several control programs. The instructions for three of the programs are given in the following section. However here is a brief summary of all of them:

GEMTERM The GEMTERM program is a terminal emulator designed to make a Gemini CP/M computer into a terminal via the GM870 modem board. GEMTERM includes menu options to select baud rates and transmission formats for the modem and can dial and initiate a call. Since it is intended to access other computer systems via the modem the program has no facility for auto answer operation. The GEMTERM program has the facility to search a control file (GEMTERM.DAT) to get all the details of a particular call, i.e serial format, phone number etc. If the specified key is found, the program automatically dials the digits and returns to a terminal program.

DIAL This is a comprehensive communications and auto dialling program. It has been adapted for use with several types of modem and also allows the user to hand over control to another program once the autodialling has been completed. This is very useful when using UKM7 as this cannot carry out these functions for itself.

UKM7 This is a version of the CP/M user group program MODEM7 which has been modified for use in the UK. The program has also been modified so that its character input and output routines support the GM870 modem card. The program provides only basic terminal routines and some file handling. Most importantly it contains very extensive error checking and correction routines for the transmission and reception of binary files using the XMODEM protocol. The program does not contain its own modem initialisation or auto-dialling routines so another program, such as DIAL, has to be used to carry out these functions.

TERMB This program is a simple terminal emulator. Everything which is typed in at the keyboard is sent to the modem. It has been modified so that the character input and output routines support the GM870. Only two characters are trapped by the program. These being ^C which exits the program and ^P which toggles the printer echo on/off. As the program is so limited and simple to operate there are no other instructions for its use. It can be used with the DIAL program.

AUTOANSW This is a simple demonstration of the auto answer capability. The program will print all received characters on the screen until it receives a ^C, at which point it will terminate the call.

3.2. GEMTERM

3.2.1. What this program can do

The GEMTERM program is a terminal emulator designed to make a Gemini CP/M computer into a terminal via the GM870 modem board. GEMTERM includes menu options to select baud rates and transmission formats for the modem and can dial and initiate a call. Since it is intended to access other computer systems via the modem the program has no facility for auto answer operation. The GEMTERM program has the facility to search a control file (GEMTERM.DAT) to get all the details of a particular call, i.e serial format, phone number etc. If the specified key is found, the program automatically dials the digits and returns to a terminal program.

3.2.2. What this program cannot do

GEMTERM has no special protocols to suit particular mainframe computers (e.g. ICL, IBM etc.). When operating in the terminal mode all characters are passed to and from the remote computer without modification (except optionally rubout and underline) and only traps one control character to exit from the terminal emulator. Any special features supported by the Gemini GM812 IVC or GM832 SVC video boards are not trapped by the program. e.g. cursor control and the setting of screen attributes may also be programmed by the remote computer. GEMTERM has no block data transmission facilities to permit binary files to be exchanged, all data MUST be in ASCII. If these facilities are required then you are referred to the UKM7 program.

In the following description the action of pressing the carriage return key is shown as <cr>. A basic working knowledge of CP/M is assumed.

3.2.3. Running the GEMTERM program

The GEMTERM program is executed by typing

```
GEMTERM<cr>
```

or

```
GEMTERM <key><cr>
```

in response to the normal CP/M prompt. Once loaded GEMTERM prints the following sign-on message.

```
Gemini Terminal Emulator version 1.2
(c) Gemini Microcomputers Ltd 1984,85
Buffer size = xxxxx bytes.

Type H for help

Command>
```

The value following the `Buffer size´ message indicates the size of the memory buffer which is used to store characters received from the remote computer. If <key> was given then the GEMTERM program will search GEMTERM.DAT for the key, select the specified baud rate and mode, dial the number for you, and run the terminal program. See the section on GEMTERM.DAT for further details.

If no <key> has been given then the `Command>´ prompt indicates that the program is waiting for a single letter command. If H is pressed then a help menu is displayed. Some of the commands are followed by requests for further information such as a filename, others simply `flip´ the state of an option and display the new state of the option.

There are two modes of operation for the GEMTERM program, Command and Terminal. Command mode is used to examine and change the operating characteristics of the program such as baud rate, half/full duplex etc. Terminal mode is when the program is actually emulating a terminal. The program starts off in Command mode and can accept a number of single letter commands. A detailed description of the action of each command follows.

A: Flip auto terminal mode flag

The A command toggles a flag to indicate to the program that when it has finished dialling a number it automatically goes into terminal mode. On booting from disk the original copy of GEMTERM defaults to having auto terminal mode turned off.

```
Command> A
Auto terminal mode ON
Command> A
Autoterminalmode OFF
Command>
```

B: Change baud rate

The B command displays the current baud rate and allows selection of one of the specified rates. On booting from disk the original copy of GEMTERM defaults to 300 baud full duplex originate.

```
Command> B
TABLE OF MODES

(0)  300 baud full duplex orig.
(1)  300 baud full duplex ans.
(2)  75/1200 baud half duplex orig.
(3)  1200/75 baud half duplex ans.
Mode:  300 baud full duplex orig.

     New mode : (0-3)
```

To set the new mode simply press the key corresponding to the mode you require.

C: Clear memory buffer

The pointer into the memory buffer is reset to the beginning, all previously stored data is lost. (Note: the memory pointer is automatically reset after a disk write using the W command).

D: Set Delay value after file character

When transmitting a file the remote computer may not be able to keep up with full speed input even at 300 baud. In order to solve this problem a variable delay has been included after each file character sent. The value of this delay can be adjusted by experiment using the D command to ensure correct data transmission. On booting from disk the original copy of GEMTERM defaults to a delay of 0 characters.

```
Command> D
Delay count for file chars sent..................    0
Enter delay for file chars sent (0-65535)?
```

E: Echo (Full/Half duplex select)

The E command flips between Full and Half duplex mode. On booting from disk the original copy of GEMTERM defaults to full duplex operation.

```
Command> E
Half Duplex
Command> E
Full Duplex
```

H: Help command

The H command prints the list of valid commands:

```
   Command> H

- A        Flip auto terminal mode flag
- B        Set baud rate
- C        Clear memory buffer
- D        Set file transmit Delay value
- E        Flip Echo flag
- H        This Help message
- L        Set line parameters
- M        Flip memory buffer ON/OFF
- N        Set Nulls after linefeed
- P        Flip printer echo
- Q or ^C  Quit to CP/M
- R        Dial a number
- S        Send a file
- T        Terminal mode
- U        Unhook - disconnect phone line
- V        View control characters
- W        Write memory buffer to disk
- X        Flip XON/XOFF protocol ON/OFF
- ?        Examine options & status
```

L: Set Line Parameters

The P command selects the asynchronous character format for data communication between the remote computer and the Gemini system. Three parameters may be changed; parity, number of data bits, number of stop bits. Options are chosen in reply to specific questions as follows:

```
Command> L
Modem interface serial parameters:
   8 bit characters, No parity, One Stop bit(s)

Number of stop bits:        (A)     one
                            (B)     one and a half
                            (C)     two
Stop bits (A,B,C)? A                     (select one stop bit)

Parity                      (N)     No parity
                            (E)     Even parity
                            (O)     Odd parity
Parity? (N,E,O) O                         (select odd parity)

Number ofbits per character (5,6,7 or 8) ?7          (7 bits)
```

On booting from disk the original copy of GEMTERM defaults to 8 data bits, 1 stop bit and no parity.

M: Toggle memory buffer ON/OFF

This command enables the user to select whether a memory buffer is to be used. If the buffer is selected then the program automatically intercepts all characters which are to be displayed on the video screen. As well as displaying these characters the program stores them away in a memory buffer. This is so the user can save the memory buffer to disk as an ASCII file and then study it again at a later date. It helps to make phone calls cheaper if you can quickly download text information and then browse through it at your leisure when the phone is offline. On booting from disk the original copy of GEMTERM defaults to having the memory buffer enabled.

```
Command> M
Memory buffer OFF
Command> M
Memory buffer ON
Command>
```

N: Set nulls to be sent after linefeed

Some computers require a number of ASCII null characters to be sent after a line feed character. The N command is used to set the number of nulls. On booting from disk the original copy of GEMTERM defaults to a setting of 0 nulls.

```
Command> N
Nulls after linefeed.............................    0
Enter nulls after linefeed (0-255)? 2
Command>
```

Two nulls will now be transmitted after the linefeed character.

P: Toggle printer echo flag

This command toggles the printer echo flag. If the echo is enabled then whenever a character is printed on the screen it is also sent to the primary list device that is defined in the CP/M IO byte. On booting from disk the original copy of GEMTERM defaults to having the printer echo disabled.

```
Command> P
Printer echo ON
Command> P
Printer echo OFF
Command>
```

Q: Quit to CP/M

Control is returned to the CP/M operating system. If the memory buffer has been enabled then before the program will exit back to CP/M it asks if you wish to save the current contents of the memory buffer to disk. After replying to this question you are then asked if you wish to save the current line parameter settings to disk. Only after replying to these two questions will the program exit and return control to the operating system.

If in reply to the question about whether you want to exit without saving the memory buffer you reply NO then you will return to the command mode. If in reply to the question about saving the current settings to disk you say yes then you will be asked to supply a filename to use. If <carriage return> is pressed on its own then the program will default to GEMTERM.COM. Once a filename has been decided upon, either by user input or by default, then the program will then proceed to save an image of itself with all of the current settings saved as the default ones. If you wish to provide a new filename, to ensure that CP/M will execute the new image the filename must include the .COM extension.

For this example assume the memory buffer is activated.

```
Command> Q
The memory buffer contains  XXXXX  bytes
Do you wish to exit without saving to disk (Y/N)? Y
Do you wish to save current status & settings to disk (Y/N)? Y
Enter filename to save to : (cr) or filename.ext
GEMTERM.COM  (if cr pressed in reply to above question)
A>
```

R: Ring a number

The R command provides the user with a way of dialling a telephone number from within the GEMTERM program. There are two ways to use this facility.

1. The number that you wish to dial can be entered directly as an ASCII string following the R command.

   i.e. R013489400

   Note that there must not be any spaces between the R and the following number, although `*' is allowed to give a delay (see GEMTERM.DAT below).

2. The R command can also be followed by an entry in the GEMTERM.DAT file.

   i.e. RTBBS

   If you study the example copy of GEMTERM.DAT that was provided with the disk you will see that in both the cases the phone numbers are the same. The program will search the data file for the key that was provided until it either finds a match or the file is exhausted. If a match is found then the corresponding number is dialled. Note that there must not be any spaces between the R and the following key.

S: Send a file

The S command is used to transmit a CP/M file from disk to the remote computer.

```
Command> Send a file from disk to line
Filename: filename.ext<cr>
Command>
```

(NOTE: On typing S the rest of the line appears automatically.)

where `filename.ext´ is the name of the disk file you wish to send to the remote computer. This facility is only suitable for transmitting ASCII files as no checking is done on the transmission, characters are sent just as though they had been typed from the keyboard. In general, anything that can be displayed on the console using the CP/M `TYPE´ command is suitable for transmission. Filetypes which probably will NOT be transmitted correctly are `.COM´, `.REL´ and `.BAS´ files stored in an intermediate format (e.g. Microsoft internal format). In the latter case use BASIC to save the file to disk in ASCII and then transmit the result. A common problem when sending files is that the remote computer cannot accept data at the full transmission speed. Delays may be added using the D command to solve this problem.

T: Terminal mode

The T command puts the program into Terminal mode. In this mode the Gemini system acts as a terminal to the remote computer. Characters typed on the console keyboard are sent to the remote computer and echoed to the console display if in half duplex mode. Characters received from the remote computer are displayed on the console and stored in the memory buffer. In order to save memory space and to avoid problems with bad characters caused by a bad connection or poor phone line only a limited set of control characters are stored in the memory. These are bell (07H), tab (09H), line feed (0AH) and carriage return (0DH). Note that bit 7 of the character is left as received. This may cause problems with some editors such as Wordstar which actually use bit 7. PIP includes a facility to strip bit 7 if problems are encountered.

On entering Terminal mode a message is printed saying which control character will return to Command mode. The default is control-V but this can be changed by simply patching a byte with DDT (see "Patch Options").

```
Command> T
^V to return to command mode
```

This control character is the only one intercepted by GEMTERM, all others are sent unmodified to the remote computer. The "delete" code (7FH) and "underline" (5FH) may optionally be converted to backspace (08H) to suit the remote computer, see "Patch Options" for further details. All other characters are sent without alteration.

To return to Command mode type control-V (or your chosen control code if you changed it). The following message is displayed:

```
     Memory buffer: xxxxx bytes used, yyyyy bytes free.
     Command>
```

indicating the state of the memory buffer. The memory buffer contents can now be written to disk using the W command.

U: Unhook - disconnect phone line

This function has been provided so that it is possible to manually disconnect the modem from the phone line. The command simply deactivates the line sieze relays and disconnects the modem from the phone system.

```
     Command> U
     Command>
```

V: Visible control characters

It is occasionally useful to see all characters received including control codes. The V command flips a flag which causes control codes to be echoed as (^x) where x is then appropriate control shifted character. For example:

```
          (^@) = null
          (^G) = bell
          (^J) = linefeed
          (^M) = carriage return
```

The format of the V command is as follows:

```
     Command> V
     Visible control characters
     Command> V
     Interpret control characters normally
```

On booting from disk the original copy of GEMTERM defaults to having this option set to interpreting control characters normally.

W: Write buffer to disk file

The W command allows the received character buffer to be written to a CP/M disk file. The entire buffer space used so far is written. As this can be as much as 50k bytes it is important to ensure that sufficient space is free on the disk BEFORE running the GEMTERM program. The format of the W command is as follows:

```
Command> W
Enter filename to save to: filename.ext<cr>
```

where `filename.ext´ is the name of the disk file which the buffer contents will be written to. If there was already a file with the name you chose it will be deleted. After the data is written the memory buffer pointer is reset to the beginning of the buffer. Note that the file will contain all characters received from the remote computer, including any introductory dialogue. You must use an editor to remove any unwanted text from the file. Clearing the memory buffer (using the C command) before receiving a file to be saved to the disk will minimise the extra unwanted text.

X: Flip XON/XOFF protocol ON/OFF

The X command is used to select whether XON/XOFF handshaking protocol is to be used for data transmission, reception. On booting from disk the original copy of GEMTERM defaults to having the XON/XOFF protocol disabled.

```
Command> X
XON/XOFF protocol ON
Command> X
XON/XOFF protocol OFF
```

?: Status of options

The ? command displays the state of all the options:

```
Command> ?
Options & Status:-

Mode: 300 baud full duplex orig.
Modem interface serial parameters:
   8 bit characters, No parity, One Stop bit(s)
Nulls after linefeed.............................    0
Delay count for file chars sent..................    0
Full Duplex
Interpret control characters normally
Printer echo OFF
Memory buffer ON
Memory buffer:     0 bytes used, 53456 bytes free.
Auto terminal mode OFF
XON/XOFF protocol OFF [XON=11H, XOFF=13H]
```

3.2.4. Examples

Sending a file `GAME.BAS´ to the remote computer.

Use Terminal mode to prepare the remote computer for receiving the file. Return to Command mode by typing control-V and type S, the `Send file´ command:

```
Command> S (send a file from disk to line)
Filename: GAME.BAS<cr>
```

The file will now be transmitted just as if you were typing it in on the keyboard. Transmission can be aborted by typing control-C twice. When file transmission is complete control is returned to Command mode. Press T to return to Terminal mode and type any commands needed by the remote computer to store the transmitted program.

Receiving and storing a file.

Since all received characters are stored in the memory buffer automatically any program listings received from the remote computer can be stored by writing the buffer to disk. Use Terminal mode to list out the program on the remote computer. Return to Command mode by typing control-V and press W, the `Write memory buffer´ command. Enter a filename for the memory buffer to be written to and press return. When the buffer has been written control is returned to Command mode. If necessary return to Terminal mode to log off the remote computer. The file containing the received program will also have all the introductory dialogue. Use an editor to remove the unwanted text leaving just the program listing.

3.2.5. Patch Options

In addition to the options which may be altered using the commands described above there are some which can be patched using DDT. These options are as follows:

1. Convert DELETE code (7FH) to backspace (08H)
2. Convert UNDERLINE code (5FH) to backspace (08H)
3. Alter `exit from terminal mode´ character (default ^V)

The following partial listing shows which bytes to patch. If the option byte is 00H then the option will be OFF, if the byte is 0FFH the option will be ON.

```
0000                     ORG     100H
0100   C3xxxx    START:  JP      BEGIN
                 ;
                 ;       OPTION SELECT BYTES
                 ;
                 ;       Bytes = 00 for option OFF
                 ;       Bytes = FF for option ON
                 ;
0103   00        EFLAG:  DB      0          ;ECHO (HALF DUPLEX)
0104   00        VFLAG:  DB      0          ;VIEW CONT CHARS
0105   FF        DEL2BS: DB      0FFH       ;CONVERT DEL (7F) TO BS
0106   00        UL2BS:  DB      0          ;CONVERT UNDERLINE (5F) TO BS
0107   00        NULLS:  DB      0          ;NULLS TO SEND AFTER LINEFEED
0108   EE02      DEFDEL: DW      750        ;DELAY AFTER FILE CHARS
010A   16        EXITCH: DB      'V'-'@'    ;EXIT CONTROL CHAR. SHOULD
                                            ;BE IN THE RANGE 01 TO 1FH
```

3.2.6. GEMTERM.DAT data file

The GEMTERM.DAT file is used by the main program to store information concerning telephone numbers. To originate a call, simply type :-

```
        GEMTERM <key><cr>
```

where <key> is a name, up to eight characters contained as an entry in GEMTERM.DAT.

The GEMTERM program will search GEMTERM.DAT for the key, select the specified baud rate and mode, dial the number for you, and run the terminal program. More entries can be added to the GEMTERM.DAT file using any text editor. Use Non-document mode in Wordstar.

Once in the GEMTERM program, there are facilities to save received data to a file, search the control file to dial another number, change baud rates and so on.

The command to search the control file is R<key> where <key> is the key word to search for in the control file, GEMTERM.DAT. Alternatively, you can dial the number direct by R012784677, this simply dials the number, you must set up the serial parameters, baud rate etc. manually.

The file contains one or more records, each with the following format.

```
<key> , <mode> , <phone number> , <monitor> , <data bits> , <parity>
 <=8     <=1       <=17             <=1          <=1           <=5   characters
```

<key> This field consists of up to 8 characters, the first of which must be a letter. Lower case letters will be treated as upper case by the program when it is searching for the setup record.

<mode> This digit represents the transmission mode.

(0) 300 baud full duplex orig.
(1) 300 baud full duplex ans.
(2) 75/1200 baud half duplex orig.
(3) 1200/75 baud half duplex ans.

<number> This field is used to store the phone number of any particular service. If a key match is found within the data file then the program will auto-dial the phone number corresponding with the identified key. If it is necessary to dial for a code for an exchange line, this can also be incorporated, and the insertion of an * into the number string will cause the program to wait for a short period before dialling the main number. This will allow time for the exchange to allocate an outside line.

<monitor> If there is the letter Y found in this field then the monitor loudspeaker will be enabled during the call.

<data> This field indicates to the GEMTERM program how many data bits are to be used for each character. If this position is empty then it will default to a character length of 7 bits. It can also contain either 5, 7 or 8.

<parity> This string specifies the type of parity that is to be used. E is entered for even parity. O is entered for odd parity and if N is entered then no parity is used. If the string is empty the it will default to even parity.

3.2.7. Example entries

```
GOLD3,1,012784355,Y,8,N
GOLD12,3,9*018372844,Y,8,N
```

All the fields within a record must be separated by a comma `,´.

Example 1 is for the Telecom Gold 300 baud service. The key is used as a label indicating Gold 300. The mode field is indicating 300 baud in answer mode. Then follows the phone number which is to be dialled to obtain the service. The next piece of information informs the software that the speaker is to be enabled. Then follows information concerning the number of data bits to be used. Finally the program is told to not use parity.

In the second example the * has been utilised to allow for the delay while an internal phone exchange allocates an external line to the call.

3.3. DIAL

DIAL is a program which is designed to allow easy use of a microcomputer with a modem. It accepts a command which is looked up in a data file called DIAL.DAT, giving DIAL instructions on how it should connect to any number of different services. This file includes for example the telephone numbers to be auto-dialled. DIAL.DAT also contains details of the hardware configuration, making it easy to implement on a number of systems.

The full range of functions is as follows:-

- Load a communications program, ready for execution the moment that the connection is established and data transmission can start
- Initialise UARTs with the specified baud rates
- Initialise a modem, including baud rate and mode
- Allow the user to enter a telephone number
- Connect the line and auto-dial a telephone number
- Check for the carrier signal
- Pass parameters to the communications program
- Pass control to the communications program
- Execute a built-in communications program
- Disconnect the line

3.3.1. Execution of DIAL

To execute the program, simply type:-

```
DIAL
```

or:-

```
DIAL command
```

If just DIAL is entered, the program will ask for the command.

The command should be the same as one of the command entries in the file DIAL.DAT, which must be available on the logged in drive. If the command is not found in the file, DIAL will ask for another command.

DIAL will display the configuration, which it obtains from the first part of the file DIAL.DAT. Then it displays the instructions corresponding to the command. It will then obey these instructions, reporting on its progress.

If DIAL finds an error in the file DIAL.DAT then an explanatory error message is displayed, which should make it easy to locate and correct. If the error is minor and relates to a specific command, DIAL will then ask for another command. Otherwise DIAL is terminated.

If DIAL fails to establish communications within 30 seconds of dialling the number then it automatically disconnects the line and asks for another command.

If the built-in communications program is used, the line is disconnected when the program is terminated and DIAL then asks for another command.

If a communications program has been loaded and executed by DIAL, then DIAL cannot disconnect the line when this program is terminated. Therefore when appropriate DIAL creates a file called $$$.SUB containing the special command "DIAL X" which is executed automatically when the communications program is terminated. This command disconnects the line.

3.3.2. Setting up DIAL.DAT

The file DIAL.DAT may be set up using an editor or word processing program. It can be of any size, so that any number of commands can be supported. DIAL reads the first part of the file, which contains details of the configuration. It then searches the file until the command required is found, and uses the values specified for that command.

Each field is defined as a series of non-blank characters separated by any number of spaces, tabs and new lines. Therefore fields must not contain spaces.

Comprehensive error reporting is built into DIAL, so that errors can easily be identified and corrected.

Examples of DIAL.DAT files are given below in the sections describing the different configurations which are supported. These show many of the possible options and should be studied in conjunction with the following section, which describes the meaning of each field in detail.

3.3.3. Specification of each field in DIAL.DAT

Headings

All the underlined headings must be left as shown in the examples. In the configuration section the first field on each line must also be left unchanged.

Speed

The first value is used to indicate the processor speed. This is needed to give accurate timings during auto-dialling. Set to "75" for a 4 MHz clock, "60" for a 4 MHz clock with wait states, or "38" for a 2 MHz clock. "75" is the normal value for most systems.

The second value is the time in milliseconds which your telephone exchange requires to connect or disconnect a line. Some very old exchanges may need as long as 5000, but 3000 is normally adequate.

Transmit

The first value is the type of UART used to transmit data. This must be 8470, 8250 or 6402. The second value is the base port at which the UART is addressed. For example this is "84" for the 8470 DART on the Gemini modem, "B8" for the 8250 UART on a Gemini CPU card, or "11" for the 6402 UART on a Nascom I/O card. The port address must be entered as a two character hexadecimal value.

If an 8470 DART is selected then the same device must be used to both transmit and receive data. In this case it is assumed that the baud rates are provided by a CTC located at a port address 4 higher than the 8470 base port. This is the case for the Gemini modem. This CTC is automatically initialised by DIAL.

If an 8250 UART is selected then the baud rates are programmed directly into the UART by DIAL.

If a 6402 UART is selected then the baud rates must be provided by appropriate hardware.

Receive

The first value is the type of UART used to receive data. This must be 8470, 8250 or 6402. The second value is the base port at which the UART is addressed. For example this is "84" for the 8470 DART on the Gemini modem, "B8" for the 8250 UART on a Gemini CPU card, or "11" for the 6402 UART on a Nascom I/O card. The port address must be entered as a two character hexadecimal value. If the receive port address is the same as the transmit port address then it is assumed that the same UART is to be used to transmit and receive data.

Note that for services requiring different transmit and receive baud rates, either an 8470 DART with CTC must be used, or two separate 8250 or 6402 devices set to different speeds, or a single 8250 or 6402 with a separate external receive clock. This clock may need additional hardware or a suitable clock signal may already be available elsewhere in the system.

Modem

The first value is the name of the modem, which has a maximum of 10 characters. The name may be entered in upper or lower case. If the modem to be used is not supported or an acoustic coupler or direct connection is to be used then enter "None". In this case the line cannot be connected or disconnected automatically and auto-dialling is not supported.

The second value is the base port address of the PIO used to control the modem, if this applies. For example this is "80" for the Gemini modem. If no PIO is used enter "00". The port address must be entered as a two character hexadecimal value.

Command

The command may be up to 24 characters long. Since the command entered is translated to upper case by CP/M, DIAL will convert the command values in the file to upper case before trying to find a match.

Program

This is the name of the communications program to be executed. Lower case letters are translated to upper case. The drive may be specified, followed by a colon, otherwise the logged in drive is used. The ".COM" extension must not be entered, as it is supplied automatically. If no program is to be executed, enter "None". The program must be no longer than 24320 bytes (5F00H). If it is longer only the first 24320 bytes are loaded. When the program is loaded the length is displayed.

DIAL includes a built-in communications program which is selected by entering an asterisk `*´ in this field. This program provides a simple full duplex terminal which will transmit any character keyed in. Characters are transmitted with even parity unless this is overridden by the UART. It will display any character received, except for control characters which are ignored, other than carriage return, line feed, bell and backspace. The program also supports simultaneous printing, although this requires a printer which can keep up with the rate at which data is received. The program may be terminated by entering a control character, and the printer may be turned on and off by entering another control character. These are specified in the parameter string described below.

Parameters

This is a string of up to 24 characters to be passed as parameters to the communications program. Lower case letters are translated to upper case. Since the parameter string may need to contain spaces, whereas this is not allowed in the file, enter an underline `_´ character instead of each space which is required. If no parameters are to be passed, enter "None". The parameters are placed in the command line input buffer and are also decoded and placed in the two fcb fields. Expansion of asterisks is not supported.

If the built-in communications program is being used, the parameter string must specify the control characters used to exit the program and to toggle the printer on and off. These must be entered as a string of four characters such as "^C^P", which means that control-C will exit the program and control-P will toggle the printer.

Transmit Speed

This is the baud rate for transmitted characters. The following values are allowed: 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200, 38400, 56000.

Modem Control Command (Transmit)

This value must be two characters long and contains a hexadecimal value.

If an 8470 DART with the Gemini modem is in use then this value must be "01" to enable the speaker, or "00" to disable it.

If an 8250 UART is in use then this value is output to the modem control register of the 8250 UART used to transmit data. Bit 0=1 for DTR on, bit 1=1 for RTS on, and if the 8250 is on the Gemini CPU card then bit 2=1 to select RS232 and bit 3=1 to disable the boot ROM. Therefore the value required is normally "0F".

If a 6402 UART is in use, enter "00".

Line Control Command (Transmit)

This value must be two characters long and contains a hexadecimal value.

If an 8470 DART with the Gemini modem is in use then the value required is normally "76" which specifies *16 clock, 8 data bits and 1 stop bit, with no parity. Set to "67" to specify 7 data bits and even parity.

If an 8250 UART is in use, then this value is output to the line control register of the 8250 UART used to transmit data. The value required is normally "03" which specifies 8 data bits and 1 stop bit, with no parity. Set to "1A" to specify 7 data bits and even parity.

If a 6402 UART is in use, enter "00".

Receive Speed

This is the baud rate for received characters. The values allowed are the same as for the transmit speed above.

Modem Control Command (Receive)

This value must be two characters long and contains a hexadecimal value.

If an 8470 DART with the Gemini modem is in use, enter "00".

If an 8250 UART is in use then this value is output to the modem control register of the 8250 UART used to receive data. Bit 0=1 for DTR on, bit 1=1 for RTS on, and if the 8250 is on the Gemini CPU card then bit 2=1 to select RS232 and bit 3=1 to disable the boot ROM. Therefore the value required is normally "0F".

If a 6402 UART is in use, enter "00".

Line Control Command (Receive)

This value must be two characters long and contains a hexadecimal value.

If an 8470 DART with the Gemini modem is in use, enter "00".

If an 8250 UART is in use, then this value is output to the line control register of the 8250 UART used to receive data. The value required is normally "03" which specifies 8 data bits and 1 stop bit, with no parity. Set to "1A" to specify 7 data bits and even parity.

If a 6402 UART is in use, enter "00".

Originate/Answer Mode

This value must be one character, and should be \`O´ or \`A´. Lower case letters are converted to upper case. If \`O´ is not entered then \`A´ is assumed. The mode is normally \`O´ unless the system is acting as a host computer to remote terminals.

Telephone Number

The telephone number may be up to 24 characters long. Hyphens \`-´ may be used to separate groups of digits to make the number easier to read. Asterisks \`*´ may be used to specify an additional delay during dialling. Each asterisk will cause a delay of 3 seconds. This may be useful when using a PABX which requires a delay after requesting an outside line.

If no number is to be auto-dialled, enter "None".

If the number to be auto-dialled is to be entered when the line has been connected, enter \`?´. This is useful as the same DIAL command may be used to connect to several similar services with different telephone numbers. DIAL will ask for the telephone number to be entered. When typing it in the backspace key may be used to correct errors. The number is auto-dialled when the Return/Enter key is pressed.

3.3.4. Notes on Communications Programs

DIAL contains a built-in communications program which is suitable for many purposes such as accessing bulletin boards, timesharing services and the Prestel 300 bps service. However some services require specific communications programs. DIAL can readily be used with these because it automatically loads the appropriate program before connecting to the service, and then passes control to the program the instant that communication is established. This avoids the possibility of any initial messages being lost while the program is loaded.

This section briefly describes some of these communications programs. These need to be configured for the type of UART in use. However since DIAL handles all initialisation this is usually very easy. In fact it may be necessary to remove redundant or incorrect initialisation code from some programs.

UKM7 is the improved UK version of the US MODEM7 program. It supports many terminal features, including storage of terminal sessions to disk. Most importantly it allows files to be transmitted and received using an error correcting protocol.

TERMB is a simple terminal program for the Gemini video card which is buffered so that it can handle local transmissions at very high speeds without loss of data.

PRETZEL is a Prestel terminal emulator which supports many features of a standard Prestel terminal using a Gemini video card.

BSTAM is a program for transmitting and receiving files using an error correcting protocol.

3.3.5. Operation Without a Modem

DIAL may be used without a modem. In this case it initialises the UART and follows the instructions in the normal way, but does not in fact connect or disconnect the line or auto-dial the telephone number. This is useful when DIAL is used to allow the system to be directly connected to another device such as a computer, printer or terminal. This method of operation also applies when using a modem which is not supported or an acoustic coupler.

This is an example of a DIAL.DAT file for operation without a modem:-

```
Configuration
-------------
Speed       75          3000
Transmit    8250        b8
Receive     8250        b8
Modem       None        00

Command     Program     Params     Trans MC LC    Rec   MC LC O/A  Telephone
-------     -------     ------     ----- -- --    ---   -- -- ---  ---------
PRESTEL300  TERMB       None       300   0f 03    300   0f 03 0    248-5747
PRESTEL300M UKM7        T_TSAV.DAT 300   0f 03    300   0f 03 0    248-5747
boarda      termb       None       300   0f 03    300   0f 03 A    None
revpres     None        None       1200  0f 03    75    0f 03 A    None
prestel     p:pretzel   None       75    0f 03    1200  0f 03 0    618
vid         pretzeln    None       75    0f 03    1200  0f 03 0    ?
b300        *           ^X^P       300   0f 03    300   0f 03 0    ?
b300m       ukm7        T          300   0f 03    300   0f 03 0    ?
```

3.3.6. The Gemini GM870 Modem

The Gemini modem is fully supported by DIAL.

This is an example of a DIAL.DAT file for the Gemini modem:-

```
Configuration
-------------
Speed       75          3000
Transmit    8470        84
Receive     8470        84
Modem       Gemini      80

Command     Program     Params     Trans MC LC    Rec   MC LC O/A  Telephone
-------     -------     ------     ----- -- --    ---   -- -- ---  ---------
PRESTEL300  TERMB       None       300   01 76    300   00 00  O   248-5747
PRESTEL300M UKM7        T_TSAV.DAT 300   01 76    300   00 00  O   248-5747
boarda      termb       None       300   01 76    300   00 00  A   None
revpres     None        None       1200  01 76    75    00 00  A   None
prestel     p:pretzel   None       75    01 76    1200  00 00  O   618
vid         pretzeln    None       75    01 76    1200  00 00  O   ?
b300        *           ^X^P       300   01 76    300   00 00  O   ?
b300m       ukm7        T          300   01 76    300   00 00  O   ?
```

3.4. UKM7

UKM7 is the improved UK version of the US MODEM7 program. It supports many terminal features, including storage of terminal sessions to disk. Most importantly it allows files to be transmitted and received using an error correcting protocol.

To execute the program type :-

```
UKM7
```

NOTE: UKM7 does not provide facilities to setup any hardware. Due to this fact UKM7 is normally executed from within DIAL.

As soon as the program has loaded the following sign on message will appear on the screen :-

```
UK Modem7   D.R. Back   Version 1.4
Control port =085H  Data port =084h

A ==>>
```

The letter that precedes the first "=" gives an indication of the currently selected drive. For instance, if you were currently selected as being on drive B: then when the program sign on message appeared then the prompt would be :

```
B ==>>
```

The program will now sit and wait for a command to be entered at the keyboard, followed by a carriage return. To display a menu of the commands enter M followed by the return key.

```
  A ==>>M (cr)

  SYNTAX: primaryoption[secondaryoption] [d:][filename][afn]

                    PRIMARY OPTIONS

  S     Send binary files, afn list
  R     Receive binary files, drive:
  T     Terminal mode. Terminal filename option
  DEL   Delete terminal file
  DIR   Directory list, afn optional
  CPM   Return to CP/M, closing any open files
  X     Expert, toggle menus on/off
  M     Menu display

                    SECONDARY OPTIONS

  N     None batched mode, send or receive file
  Q     Quiet mode. Remote system Send/Receive
S,R,V   Monitor data Sent, Received or view file
  T     Go to terminal mode after file transferes
```

3.4.1. Primary options

The program has two modes of operation, Terminal mode and command mode. There now follows an explanation of the primary options for the command mode:-

S: Send a binary file

This option is used to transfer a binary file from the host computer to the remote terminal. After the S there should be a secondary option if required, followed by the filename of the file to be transfered. The file transmission and reception algorithms contain extensive error detection and correction routines. For these to work successfully the data is transmitted in packets of a fixed size and format. They are not transmitted as a continuous stream. This means that binary file transmission can only take place between two systems using the XMODEM protocol.

R: Receive a binary file

The receive binary file option contains the same error detection and correction routines as the send binary file option. As such it can only be used for receiving files which have been transmitted using the XMODEM protocol When initiating the routine no filename needs to be specified as this is included in the data transmitted from the remote terminal. If you wish to change the default drive then another one can be specified. Again when receiving the file the program transmits acknowledge signals to the transmitting computer.

T: Terminal mode

This option causes the program to enter the terminal mode. Once in terminal mode the program echoes all characters typed in at the keyboard to the remote terminal. Certain characters are trapped by the program and are used for functions within the program. To find the list of keys that are trapped by the program use the X function before entering the terminal mode. If the expert function had been previously enabled then on entry to the terminal mode a menu of available functions is displayed. If on entry to the terminal mode you follow the T command with a filename then a file will be opened with this name and will be updated to contain all of the information that is transferred to and from the computer. Any previous files with the same name are deleted.

```
A ==>>X (cr)
A ==>>T (cr)

^T   Transfer (send) ASCII file without checks
^X   Abort transfer initiated above
^C   Computer mode, toggle echo on/off
^^   Send following char literally
^E   Exit to command menu
^D   Display terminal menu
```

Explanation of terminal commands

^T After this key has been pressed the program will prompt for the filename of the file to be passed directly onto the remote computer. If no filename is given then a bad syntax error is generated. The specified ASCII file is transmitted directly to the receiving computer with no checks. This enables UKM7 to pass ASCII files to computers which are not operating UKM7.

^X This will abort any file transfers that are in progress.

^C After typing this key the full duplex operation of the program is reversed. This makes the system act as host where the other computer is in terminal mode.

^^ This command is useful if you wish to transmit control codes to the remote terminal.(ie cursor addressing). It prevents the program from stripping out control codes. (ASCII codes below 20H.)

^E This command directly aborts the terminal mode and returns the user to command mode. If a terminal file was specified on entry to terminal mode then the file remains open. Re-entry to terminal mode at a later time will continue to update the previous terminal file.

^D This will display the menu of command keys for the terminal mode.

DEL: Delete terminal file

This command will delete the file with the name that matches the one specified after the command letter.

CPM: Return to CP/M

The CPM command returns the user to the CP/M operating system in a tidy state. If the user has previously opened any files for reading or writing then they are closed down so as to avoid loss of data. You should not return to CP/M by pressing reset because this will fail to close any open files in the correct manner.

X: Expert

The expert flag simply decides whether, on entry to or exit from the terminal mode, the relevant menu tables are displayed.

M: Menu display

Issuing the M command automatically displays the main command menu, regardless of whether the X command has disabled its display or not.

3.4.2. Secondary options

These secondary options can only be used with primary option S and R

N: Non batched mode

The non batch mode will transfer files without using the multiple file batching protocols which are used by the error detecting and correction routines. It is provided primarily to provide compatibility with older versions of the MODEMX program. These older versions require B to be entered to specify batch mode.

Q: Quiet mode

If the quiet instruction command is executed then the program will go into quiet mode. This will allow a remote terminal to pass instructions directly to the program. This allows terminal to invoke automatic file transfers.

S/R/V: Monitor files

These secondary options will allow the user to monitor, on the screen, the contents of the files that are being transferred between the two machines. In normal mode any file transactions are carried out transparent to the user. Using these commands tell the program to pass information on to the user concerning the progress of the transfer.

T: Go to terminal mode

When this option is used then whenever a file transfer is performed then the program will revert to terminal once the transfer has completed.

3.4.3. Explanation of command syntax

The menu indicates a general command syntax which applies to all commands but as syntax varies somewhat from one command to another, the explanation below may be useful.

SEND FILES IN BATCH MODE, primaryoption S

Either single or multiple files may be sent with one command by use of this option.

The Sending command is:

```
        S[Q][S|R|V][T]  [drive:]afn [ [drive:]afn]
```

i.e. you may use a list of ambiguous filenames separated by spaces. Files with the $SYS attribute set, i.e. system files, cannot be sent. If the receiving station is not using UKM7 you must tell it that you are going to use Batch Mode.

RECEIVE FILES IN BATCH MODE, primaryoption R

Filenames are transmitted along with the file and used to make new files at the receiving station. If a file of the same name already exists the existing file will be renamed .BAK

The Receiving command is:

```
R[Q][S|R|V][T] [drive:]
```

If a drive is not specified then files will be directed to the default drive.

TERMINAL MODE, primaryoption T

To enter Terminal mode:

```
T [drive:][unambiguousfilename]
```

In order to start a terminal file, a filename must be specified when entering terminal mode.

Exit from terminal mode to Menu level (using ^E) does not close an existing terminal file.

If a terminal file is open then re-entry to terminal mode without a filename will allow its continued use.

Entry to terminal mode with a filename specified will close the current terminal file, if one is open, and open a new file with the specified name.

When a return is made to CP/M command level by using the CPM primary option, the current terminal file will be closed. Do not exit to CP/M by re-booting else an open terminal file will not be closed and data will be lost.

The terminal menu may be displayed after entry to terminal mode by typing ^D. When a terminal file is open, its name will be displayed in the menu. No name means no file and data cannot then be saved.

The Computer mode may be entered by typing ^C. Only one end of a communication link may be in Computer mode at any one time. In Computer mode characters received via the Modem are echoed back to the originator and characters sent are echoed to the local console. Line feeds are automatically appended to carriage returns. Both terminal file save and file transfer with the ^T option work in Computer mode as they do in Terminal mode.

When a terminal file save is in operation then a colon, `:´, is displayed at the start of each line.

3.4.4. Direct entry from CP/M command level

Any of the above commands may be given to UKM7 direct from CP/M command level. For example

```
A>UKM7 ST B:*.COM IQ*.ASM
```

will transmit all .COM files from drive B: and all .ASM files which begin with IQ from drive A: and finish up in Terminal mode. There will be no terminal file.

```
A>UKM7 T B:TFILE
```

will open terminal file B:TFILE and enter terminal mode.

Both M and X commands may also be executed direct from CP/M command level.

NON BATCH MODE, secondaryoption N

This mode has been retained for compatibility with older versions of MODEMX. One unambiguous filename is required for both Send and Receive, i.e. only one file may be transferred at a time.

Send a file, non batch mode:

```
SN[Q][S|R|V][T] [drive:]unambiguousfilename
```

Receive a file, non batch mode:

```
RN[Q][S|R|V][T] [drive:]unambiguousfilename
```

4. HARDWARE

4.1. Dialling

In order to ensure accurate and repeatable dialling waveforms a dialling I.C. (the Mostek 50991) is used, which in turn controls the dialling/siezing relay. This relay has the dual function of siezing and subsequently holding the telephone line for the duration of the call and, during the dialling interval, is intermittently operated to provide loop disconnection at the requisite 10 pulses per second.

The timebase for dialling is provided by a 4 Khz signal provided by a Z8430 CTC (timer 0) dividing the 4 Mhz frequency oscillator associated with the Z80 CPU chip. The I.C. takes care of all the necessary timing and also provides an indication of when dialling is over, so that the processor can then continue the process of connecting a call for the modem.

The processor interface to this dialling I.C. is via a Z8420 PIO device (port a). The PIO simulates a 3 by 4 keyboard matrix common in telephone practice. A digit is dialled by taking both one of the column inputs and one of the row inputs low. The coding of these inputs is shown in table 2. The inputs are only taken as significant whilst the off/on hook input to the dialling I.C. is in the off hook position. The off/on hook signal, for the purposes of hardware simplicity, is connected to the loop line output from the PIO (discussed later).

The reason this kind of interface is necessary is because of the dialler chips designed application. The chip is designed to debounce the "keyboard" in order to simplify design and reduce component count. Thus it is required that any digit-to-be-diallied selection is held active for at least 50 ms , and the no digit selected condition is held for at least 20ms between digits-to-be-dialled.

The digits to be dialled should be loaded sequentially following the sequence above simulating the normal depression sequence of keys as in a normal telephone. As soon as the first selection is made the dialler I.C. will start dialling.

4.2. Line interface control

The line interface is the means by which the modem connects to the telephone line and manipulates the telephone system, by effectively mimicking a standard telephone. In this particular case auto-answering is required so that a means of detecting incoming ringing current is incorporated. Further it is necessary to isolate the actual computer system from the fully floating, high voltage (150 volts) telephone pair. This isolation is most easily achieved by the use of a rather bulky (in the terms of modern technology silicon) transformer. The isolation requirement is to form a barrier proof against 5.3 Kv (this applies to all relays, transformers and opto-isolators used to bridge the cpu enviroment to telephone enviroment).

The transformer is a simple 1:1 winding ratio device designed to match 600 ohms on both sides. The regulatory requirement for this element means that in conjunction with its related terminating resistors etc the return loss of the finished, complete modem must be better than 12 db across the range 300 to 3400 Hz.

The transformer is coupled into the modem chip via a series of active devices. Firstly there is an active three port network. This circuit's purpose is to try and reduce the amount of transmit power actually seen by the receiver in the modem chip during V.21 operation (300/300 baud full duplex). As much as 6 db of improvement can be gained and this improves performance in marginal conditions.

There are three relays associated with the mimicking of the normal telephone:

1. The dialling/ loop sieze relay- This is driven from the dialler chip and the PIO together so that, as described earlier the relay is pulsed during dialling and then held in the made state to hold the telephone circuit for the duration of that call.

2. The Bell shunt relay- This relay clamps the second interface wire coming from the "New Plan Socket" to the "b" leg of the telephony pair. This is done via a 330 ohm resistor and two back to back zeners each of cut-off value 3.3v. The purpose of this arrangement is to prevent auxilary telephone bell tinkle (telephones may be connected to secondary "New Plan Sockets" and thus in parallel with the modem). This connection is usually held for the duration of a call, from before dialling until after the releasing of the line loop by the line sieze/dialling relay contact. Providentially this connection de-arcs the dialling relay contacts via the capacitor in the primary "New Plan Socket".

3. The transformer shunt relay- This relay is used to shunt out the resistance of the line isolation transformer during dialling, so as to reduce dial pulse distortion.

Also involved in the line interface is the ringing current detector which is used to detect incoming ringing for auto-answer applications of the modem. An opto isolator is used to give an isolation barrier. The LED in the isolator is connected to the line side of the bellshunt contact. This contact is connected via a 330 resistor and pair of back-to-back zener diodes to the capacitor in the primary "New Plan Socket". Thus the ringing current detector is AC coupled to the telephone line. In series with the LED in the isolator is a resistor of 4K7 ohms. This limits the current through the diode and makes the circuit of reasonably high AC impedance. A further consequence of having the ringing current detector wired as indicated is that the possibility of having false incoming call indications during out going calls is much reduced since the detector is effectively shorted out by the bell shunt contact, which is normally made during the complete duration of an outgoing call.

The output of the optoisolator phototransistor is used to charge a capacitor. After several rings the capacitor becomes sufficiently charged to trigger a monostable (NE555). The output from this is then connected to the ringing indicator input of the DART serial I/O chip. This can be programmed to be a source of interrupts so that the system can process incoming calls. In the auto-answer mode the dial tone detector may cause the apparent loss of carrier, due to having detected dial/engaged tone. In this case and in any other the absence of carrier for more than 500ms will cause the call to be dropped and the circuit cleared. The dial tone detector is automatically placed in circuit as soon as the RI signal is given to the AMD modem chip. When an incoming call interrupt is generated by the DART having an input on its ringing indicator line the software will set the RI input on the modem chip to be active (ie set to 0 volts). This will enable the dial tone detector to be combined with the carrier detect input on the DART so that dial tone will cause a carrier detect fail interrupt to the system, which can then process it, dropping the line as a result.

4.3. Modem Control

The modem chip has 7 mode control inputs. Six of them set the actual mode ie V.21 or V23 etc, the remaining one is the RI input which when used in combination with the normal modem controls a la RS232C causes the modem to send out the V.25 answering tone (2100 Hz) for the requisite period before going into the mode set by the 6 mode control lines. The 6 mode control lines are reduced to only 2 to force the modem chip into V.23 and V.21 modes. The other 4 are permenantly wired to set them as required. The PIO manipulates these 2 mode control inputs as required.

The DART supplies the other normal RS232C signals (DTR,RTS,CTS,CD) to the modem chip. The modem chip has a peculiar back channel arrangement inherited from the bell standard modems, but the rules are very simple. If in V.21 mode (300/300 baud) then the RTS signal from the DART is used. If V.23 is used to gain access to Prestel, for example, then the BRTS signal (B7 on the PIO) should be used instead and the RTS signal left in its inactive state (5 volts).

4.4. The Monitor Loudspeaker System

This is provided to help the user decide on the progress of the call he is making by audible means. The volume of the output through the in built loudspeaker is controlled by the CPU so that it can be completely muted or turned up to a user chosen level (set by a potentiometer) under software control.

4.5. Hardware allocation

There are three main devices under the direct control of the Z80 CPU:-

4.5.1. Z8420 - PIO

This is a TTL compatible 16 bit I/O device when used in this system (mode 3). The A port (a0 to a7) is used to control the dialling chip (Mostek 50991) as previously described. A7 is used both to give the dialling chip the off/on hook indication and to control the line sieze relay. The device occupies a total of 4 I/O ports and the decoding for them is provided below. The allocation of the A port is as follows:-

```
PIO:        EQU    80H       base address
PADATA:     EQU    PIO+0     port A data
PBDATA:     EQU    PIO+1     port B data
PACNTRL:    EQU    PIO+2     port A control
PBCNTRL:    EQU    PIO+3     port B control
```

PIO port		50991 function	pin no.
A0	pin 15	column 1	3
A1	pin14	column 2	4
A2	pin13	column 3	5
A3	pin12	row 1	16
A4	pin10	row 2	15
A5	pin 9	row 3	14
A6	pin 8	row 4	13
A7	pin 7	on/off hook (line loop)	17

N.B. the off hook condition is when A7= 0 volts

Translation table- PIO port A hex codes to dialled digit

This table assumes that the chip has been told to go off hook ie A7 = 0volts.

digit to be dialled	row active	col active	hex code
1	1	1	09
2	1	2	0A
3	1	3	0C
4	2	1	11
5	2	2	12
6	2	3	14
7	3	1	21
8	3	2	22
9	3	3	24
0	4	2	42
redial	4	3	44

N.B. `Active´ rows and columns are when the corresponding outputs are low ie = 0 volts.

PIO port B allocation

Port	Function
B0	Mute input from the dialler chip the 50991
B1	Bell shunt relay control output. shunted when B1= 5 volts
B2	Transformer shunt relay control output. Bell shunted when B2 = 5 volts
B3	RI output to AMD 7910 modem chip answer mode selected when RI = 0 volts
B4	MC0 output to AMD 7910 modem chip } SEE TABLE BELOW
B5	MC1 output to AMD 7910 modem chip } SEE TABLE BELOW
B6	Loudspeaker volume mute control output Muted when = 0 volts
B7	BRTS on AMD 7910 modem chip active when = 0 volts

Table of modes of modem chip vs MC0, MC1 settings

MC0	MC1	Modem mode
0	0	V.21 originate 300 baud full duplex
0	1	V.21 answer 300 baud full duplex
1	0	V.23 mode2 1200 baud half-duplex
1	1	V.23 mode 2 1200 baud half-duplex equalised

The settings for the other mode control lines are:-

MC2 =5 volts -This bit sets Bell/ CCITT modes
MC3 =0 volts
MC4 =0 volts.

The direction of the ports of the PIO should all be set to outputs with the exception of B0 , which should be an input.

4.5.2. Z8430 - CTC

This device provides a source of clock from the 4 MHz system clock. The fourth counter/timer is not used. All of the other counter timers are set to timer mode with a prescaler of 16 set. The DART divide ratio should be set to 16. The port decodes for the CTC are given below.

The CTC channel allocation.

CTC 0 is used to generate the baud rate clock for the serial device receiver channel.

CTC 1 is used to generate the baud rate clock for the serial device transmitter channel.

CTC 2 is used to generate the 8kHz clock which is in turn divided down to provide a 50:50 duty cycle 4kHz clock to the pulse dialler chip.

CTC 3 is not used.

timer0 -receive baud rate to DART

	4Mhz	
V.21 (300/300)	52	300 baud
V.23 (1200/75)	13	1200 baud

timer1 -transmit baud rate to DART

	4 Mhz	
V.21 (300/300)	52	300 baud
V.23 (75/1200)	208	75 baud

N.B. prescaler
in timer 1 set to 64
in this case only

timer2 -dialler chip reference frequency source

4 Mhz

63 decimal

CTC port decodes

```
CTC:        EQU     88H             base address
CTC0:       EQU     CTC             channel 0
CTC1:       EQU     CTC0+1          channel 1
CTC2:       EQU     CTC0+2          channel 2
CTC3:       EQU     CTC0+3          channel 3
```

4.5.3. Z8470 - DART

The DART - This device is a dual channel asynchonous receiver transmitter. Its function is to interface the 8 bit parallel data streams inside the Z80 computer to the serial world of the modem chip. Only one of the channels is used, leaving the other free. Channel A is the utilised one.

As well as this function the DART can coordinate interrupts. This feature is used in this system to generate interrupts for incoming ringing detection, through the RIA pin on the DART, and carrier detect failure.

The DART should be set for a divide by 16 ratio.

The DART port decode.

```
DART:       EQU   84H       base address
DADATA:     EQU   DART+0    channel A data
DACNTRL:    EQU   DART+1    channel A control
DBDATA:     EQU   DART+2    channel B data
DBCNTRL:    EQU   DART+3    channel B control
```

DART channel allocation.

Channel A is used for serial communication to and from the modem chip.

Channel B serial is not use for anything. However, /DTRB and /RIB pins are used for other function i.e /RIB is used to select between PULSE(0) or TONE (1) dialling and /DTRB pin is used to select between PULSE or TONE dialler chip.

For users who wish to know more specific details about how to program either the DART, CTC or PIO you are referred to the relevant MOSTEK data sheets.

5. SOFTWARE INTERFACE PROCEDURES

The following section gives details of the procedures that have to be carried out if you wish to write your own communications software.

5.1. The Flow for making a call

Remove DTR from the modem chip via the DART
Similarly remove RTS
Set the modem mode by setting MC0, MC1 using the PIO
Disable the ringing interrupt source in the DART
Make the bell shunt contact using the PIO
Make the transformer shunt contact using the PIO
Go off hook (PIO A7) and hence sieze the line by actuating the sieze contact
Wait for 5 seconds
Load the digits-to-be-dialled into the dialler chip as described in the text.
Wait for the mute signal from the dialler chip to go inactive by monitoring the PIO.
Wait 100ms
Release the transformer shunt contact using the PIO.
Wait 1 second
Enable the carrier detect interrupt source in the DART.
Wait for carrier detect.
If no carrier is detected within 60 seconds then drop the call.
Activate the RTS (or BRTS) on the modem chip using the DART (or PIO).
Activate DTR on the modem chip using the DART
During the call wait for carrier failure.

5.2. The Flow for finishing a call

Drop DTR
Drop RTS
Go on hook
Release The bell shunt contact using the PIO.
Enable the ringing interrupt source in the DART
Disable the carrier detect interrupt in the DART

6.4. Connection and disconnection

```
;     Subroutine to connect the modem to the telephone line
;
;     This routine simply controls the relays
;
OFFHOOK:LD      A,00000000B         ;OFF HOOK
        OUT     (PADATA),A
        IN      A,(PBDATA)          ;GET PIO B DATA
        AND     11111110B           ;BIT 0 IS MUTE INPUT
        OR      00000110B           ;MUTE AND BELL SHUNT ON
        OUT     (PBDATA),A
        LD      HL,3000             ;3000 MS DELAY
        CALL    DELAY               ;1ms DELAY SUBROUTINE
        RET
```

* * * Subroutine to end a call * * *

This routine ends a call by clearing all the relays, and turning off DTR and RTS.

```
ENDCALL:LD      A,10000000B         ;ON HOOK TO FINISH CALL
        OUT     (PADATA),A
        LD      A,(REG5)            ;GET REGISTER 5 DATA
        AND     01111101B           ;RESET DTR AND RTS
        LD      (REG5),A
        LD      C,A
        LD      B,5                 ;REGISTER 5
        CALL    WRREG
        IN      A,(PBDATA)          ;GET PIO B DATA
        AND     01111001B           ;TRANSFORMER AND BELL OFF
        OR      11001000B           ;RI, BRTS FALSE, AND LINE MONITOR ON
        OUT     (PBDATA),A
        RET
```

N.B. Because register 5 in the DART cannot be read back, a copy needs to be kept in memory (REG5) whenever data is written to register 5.

This is to update the DART status register to check CTS and DCD status.

```
UPSTAT: LD      A,00010000B         ;RESET EXT INTERRUPTS FLAGS
        OUT     (DACNTRL),A
        RET
```

6.5. Dialling

A subroutine to dial a number entered from the console or redial the previous number (stored in the dialling chip).

Valid characters are the digits 0-9, * and £. £ on its own will cause the last number to be redialled. Embedding a * in the number will cause an inter-digit pause of 3 seconds, useful when there is a delay after obtaining an outside line from an internal exchange. The number should contain only the above characters, no spaces or tabs.

```
;
;     DIAL A NUMBER SPECIFIED BY THE USER
;
CALL:   LD      C,10             ;READ CONSOLE BUFFER TO GET DIGITS
        LD      DE,NUMPTR        ;WHERE NUMBER GOES
        CALL    BDOS
;
        LD      A,(NUMPTR+2)     ;CHECK FOR REDIAL
        CP      '£'
        JR      Z,RDIAL
;
AUTOCAL:LD      A,0FFH           ;NORMAL DIAL
        JR      CALL1
;
RDIAL:  XOR     A                ;REDIAL
CALL1:  LD      HL,NUMBER        ;POINTER TO NUMBER FOR DIALLING
        CALL    DIAL             ;DIAL IT AND RETURN TO COMMAND
;
;     SET DTR AND RTS OR BRTS DEPENDING ON THE MODE SELECTED
;
        LD      A,(BAUD)         ;GET MODE OF 7910
        CP      2                ;75TX/1200RX
        LD      B,10000010B      ;ASSUME DTR AND RTS
        JR      NZ,SETDTR
        IN      A,(PBDATA)       ;GET MODE BYTE AND CONTRO, SIGNALS
        AND     01111111B        ;MASKED BRTS
        OUT     (PBDATA),A       ;SET BRTS
        LD      B,10000000B      ;NO RTS
;
SETDTR: LD      A,(REG5)         ;GET WHAT WAS WRITTEN TO REG 5 OF THE DART
        AND     01111101B        ;MASK RTS AND DTR
        OR      B                ;Ored IN DTR AND PERHAPS RTS
        LD      B,5
        LD      C,A
        CALL    WRREG
        LD      HL,3000          ;FOR CARRIER TO SETTLE
        CALL    DELAY            ;3 SECONDS
        RET
;
NUMPTR: DB      17               ;MAX OF 17 DIGITS
NUMBER: DS      20               ;NUMBER OF DIGITS
```

```
;------------------------------------------------------------------------
;    DIALLING ROUTINE
;    ENTRY WITH A+OFFH & HL -> DIALLING TO BE DIALLED
;         FIRST BYTE = NUMBER OF DIGITS (BINARY)
;         NEXT BYTES = ASCII DIGITS TO DIAL
;
;    TO REDIAL THE LAST NUMBER (STORED IN THE DIAL CHIP)
;    ENTER WITH A=00
;
DIAL:     PUSH     HL
          PUSH     AF
          CALL     OFFHOOK          ;GET READY FOR DIALLING
          POP      AF
          POP      HL
          OR       A                ;REDIAL ?
          JP       Z,REDIAL
          LD       A,(HL)           ;GET NUMBER OF DIGITS TO DIAL
          LD       B,A              ;INTO B
NXTDIG:   INC      HL
          LD       A,(HL)           ;GET FIRST DIGIT
          CP       ´*´              ;DELAY BETWEEN DIGITS
          PUSH     HL
          PUSH     BC
          JP       NZ,DIALIT
;
;    We found a ´*´ so put a delay before the next digit
;
CHKMUT:   IN       A,(PBDATA)       ;IS MUTE OFF
          AND      1
          JR       NZ,CHKMUT        ;MUTE IS OFF
          LD       HL,3000          ;WAIT FOR 3 SEC
          CALL     DELAY
          JR       NEXTD            ;DO NEXT DIGIT
DIALIT:   CP       ´0´              ;LESS THEN ZERO
          JP       C,BADDIG
          CP       ´9´+1            ;GREATER THEN 9
          JP       NC,BADDIG
          SUB      ´0´              ;TURN ASCII TO NUMBER
          CALL     SENDIG           ;DIAL THE DIGIT
NEXTD:    POP      BC
          POP      HL
          DJNZ     NXTDIG           ;UNTIL FINISH
MUTE:     LD       A,00000000B      ;OFF HOOK AND NO DIGITS
          OUT      (PADATA),A
          LD       HL,20
          CALL     DELAY
MUTE1:    IN       A,(PBDATA)       ;TURN MUTE OFF
          AND      1                ;BIT 0 IS MUTE
          JR       NZ,MUTE1
          LD       HL,100           ;WAIT 100MS
          CALL     DELAY
          IN       A,(PBDATA)       ;GET PORT B DATA
          AND      11111011B        ;DE-SELECT TRANSFORMER SHUNT
          OUT      (PBDATA),A
          RET
```

```
;
REDIAL: LD      A,10            ;REDIAL CODE
        CALL    SENDIG
        JR      MUTE
;
BADDIG: CALL    PTEXT
        DM      CR,LF,´Invalid telephone digit´,CR,LF
        CALL    ENDCALL         ;HOOK PHONE BACK ON
        JP      0
;
;     SEND A DIGIT CONTAINED IN A
;     REGISTER CONTENTS ARE NOT SAVED
;
SENDIG: LD      HL,DIGPTR       ;POINT TO TRANSLATION TABLE
        LD      C,A
        LD      B,0
        ADD     HL,BC
        LD      A,(HL)          ;GET HEX CODE
        OUT     (PADATA),A      ;SEND IT
        LD      HL,50           ;HOLD IT FOR 50MS
        CALL    DELAY
        LD      A,0             ;DESELECT THE KEYS
        OUT     (PADATA),A
        LD      HL,20           ;20MS NO DIGIT DELAY
        CALL    DELAY
        RET
;
;     DELAY WITH HL= NUMBER OF MS
;     DESTROY REGISTERS HL, BC, A
;
DELAY:  LD      BC,200          ;DELAY COUNT FOR 1MS
DEL1MS: DEC     BC              ;4 MHz CPU clock
        LD      A,B
        OR      C
        JR      NZ,DEL1MS
        DEC     HL
        LD      A,H
        OR      L
        JR      NZ,DELAY
        RET
;
;     DIGIT TO HEX CODE TRANSLATION TABLE
;
DIGPTR: DB      (NOT 3DH) AND 7FH       ;DIGIT 0
        DB      (NOT 76H) AND 7FH       ;1
        DB      (NOT 75H) AND 7FH       ;2
        DB      (NOT 73H) AND 7FH       ;3
        DB      (NOT 6EH) AND 7FH       ;4
        DB      (NOT 6DH) AND 7FH       ;5
        DB      (NOT 6BH) AND 7FH       ;6
        DB      (NOT 5EH) AND 7FH       ;7
        DB      (NOT 5DH) AND 7FH       ;8
        DB      (NOT 5BH) AND 7FH       ;9
        DB      (NOT 3BH) AND 7FH       ;REDIAL
;
```

6.6. Initialisation

This initialisation routine will set up the board using a configuration defined in a number of memory locations.

MDART is a byte storage location and contains the serial parameters

Bit 0 - Parity enable(1) and disable(0)
1 - Parity even(1) and odd(0)

3 2 - 01= 1 stop bit/character
10= 1.5 stop bit/character
11= 2 stop bit/character

5 4 - 00= 5 bits/character
01= 6 bits/character
10= 7 bits/character
11= 8 bits/character

7 6 - 00= X1 clock
01= X16 clock (normal value)
10= X32 clock
11= X64 clock

BAUD is a byte storage location and contains the mode of the 7910 modem chip.

Value 0 = 300 baud full duplex Originate.
1 = 300 baud full duplex Answer.
2 = 75/1200 baud half duplex Originate.
3 = 1200/75 baud half duplex Answer.

MON is a byte storage location and contains either ASCII `Y´ or `N´, this is used to enable or disable the speaker.

Example :

300 Baud full duplex Originating mode
16x clock, 8 Data, no parity, 1 stop bit.

MDART = 01110100B
BAUD = 00000000B

NOTE: The MDART value is the same as that for the "Line Control Commands" described in the documentation of the DIAL program.

```
;
;     INITIALISE DEVICES WITHIN THE MODEM INTERFACE
;
MINIT:
;
;     SET UP CTC2 AT 4KHZ
;
        LD      A,00000101B     ;NO INTERRUPT, AND TIMER MODE
        OUT     (CTC2),A
        LD      A,31            ;8KHZ
        OUT     (CTC2),A
        LD      A,0             ;NO INTER. VECTOR
        OUT     (CTC2),A
;
;     SET UP THE PIO CHANNEL A FOR O/P AND CHANNEL B
;     BIT 0 FOR I/P AND THE REST AS O/P
;
        XOR     A               ;DUMMY INTERRUPT VECTOR
        OUT     (PACNTRL),A     ;TO A
        LD      A,11001111B     ;A=CONTROL MODE 2
        OUT     (PACNTRL),A
        LD      A,00000000B     ;ALL OUTPUT
        OUT     (PACNTRL),A
        LD      A,00000111B     ;NO INTERRUPTS
        OUT     (PACNTRL),A
;
        LD      A,10000000B     ;ON HOOK, AND NO DIGITS
        OUT     (PADATA),A
;
        XOR     A               ;DUMMY INTERRUPT VECTOR
        OUT     (PBCNTRL),A     ;TO B
        LD      A,11001111B     ;B=CONTROL MODE 2
        OUT     (PBCNTRL),A
        LD      A,00000001B     ;BIT 0 IS INPUT ON PORT B
        OUT     (PBCNTRL),A
        LD      A,00110111B     ;NO INTERRUPTS
        OUT     (PBCNTRL),A
        LD      A,11111111B     ;MASK
        OUT     (PBCNTRL),A
;
;     INITIALISE THE DART USING MODE BYTE
;
        LD      BC,1*256+0      ;REG 1 AND NO INT
        CALL    WRREG           ;WRITE TO REG
        LD      BC,2*256+0      ;REG 2 AND DUMMY INTERRUPT VECTOR
        CALL    WRREG
        LD      A,(MDART)       ;GET MODE
        AND     11001111B       ;MASKED OFF DATA BITS
        LD      B,4             ;LINE PARAMETERS
        LD      C,A
        CALL    WRREG
        LD      A,(MDART)       ;GET MODE
        AND     00110000B       ;ONLY DATA BITS
        RRCA
        RRCA
        RRCA
        RRCA
        PUSH    AF              ;SAVE A COPY
        LD      C,A
        LD      B,0
        LD      HL,TDTAB        ;DATA BITS TRANSLATION
        ADD     HL,BC
        LD      A,(HL)          ;GET IT
        LD      B,5
        LD      C,A
        LD      (REG5),A        ;SAVE A COPY FOR DTR AND RTS USE
        CALL    WRREG
;
        POP     AF              ;GET COPY BACK
        LD      C,A
        LD      B,0
        LD      HL,RDTAB
        ADD     HL,BC
        LD      A,(HL)
        LD      B,3             ;WRITE TO REG 3
        LD      C,A
        CALL    WRREG
;
        LD      A,(MON) ;GET MONITOR OR NOT
        CP      'Y'             ;MONITOR ?
        LD      A,10001000B     ;MONITOR AND NO MODE
        JR      Z,MINIT2
        LD      A,11001000B     ;DO NOT MONITOR AND NO MODE
MINIT2: LD      B,A             ;SAVE A COPY
        LD      A,(BAUD)        ;GET BAUD RATE
;
;       SET BAUD RATE ON DART AND 7910 MODEM CHIP
;
        RLCA
        RLCA
        RLCA
        RLCA
        OR      B               ;ORed WITH DATA
        OUT     (PBDATA),A
        LD      A,(BAUD)        ;GET BAUD RATE AGAIN
        LD      L,A
        LD      H,0
        ADD     HL,HL           ;*2
        ADD     HL,HL           ;*4
        LD      BC,CTCTAB       ;TABLE OF BAUD RATES FOR THE CTC
        ADD     HL,BC
        LD      C,(HL)          ;PRESCALER FOR TX
        INC     HL
        LD      B,(HL)          ;TIMER CONSTANT FOR TX
        INC     HL
        LD      E,(HL)          ;PRESECALER FOR RX
        INC     HL
        LD      D,(HL)          ;TIMER CONSTANT FOR RX
```

```
;
;     SET UP RX BAUD RATE
;
        LD      A,00000101B     ;NO INTERRUPT, AND TIMER MODE
        OR      E               ;PRESCALER
        OUT     (CTC0),A
        LD      A,D             ;GET RX BAUD RATE VALUE
        OUT     (CTC0),A
        LD      A,0             ;NO INTER. VECTOR
        OUT     (CTC0),A
;
;     SET UP TX BAUD RATE
;
        LD      A,00000101B     ;NO INTERRUPT, AND TIMER MODE
        OR      C               ;PRESCALER
        OUT     (CTC1),A
        LD      A,B             ;GET TX BAUD RATE VALUE
        OUT     (CTC1),A
        LD      A,0             ;NO INTER. VECTOR
        OUT     (CTC1),A
        RET
;
;
;     WRITE TO A REG IN DART
;     ENTRY WITH B=REGISTER AND C=DATA TO WRITE INTO REG
;
WRREG:  LD      A,B             ;SELECT REG
        OUT     (DACNTRL),A
        LD      A,C             ;GET DATA
        OUT     (DACNTRL),A
        RET
;
TDTAB:  DB      00001000B       ;5 BITS
        DB      01001000B       ;6 BITS
        DB      00101000B       ;7 BITS
        DB      01101000B       ;8 BITS
;
RDTAB:  DB      00100001B
        DB      10100001B
        DB      01100001B
        DB      11100001B
;
CTCTAB: DB      0,52,0,52               ;PRESCALER OF 16
        DB      0,52,0,52               ;/16
        DB      32,13,0,13              ;/256, AND /16 75TX/1200RX
        DB      0,13,32,13              ;/16 AND /256 1200TX/75RX
;
```

A. Disk software

The diskette normally supplied with the GM870 is in Gemini QDDS format. This may be read on all Gemini Galaxy systems, Gemini MultiBoard systems with GM825 disk sub-systems, all Quantum systems, Kenilworth 83G models, and Nascom systems fitted with the Gemini 809/829 FDC board, GM825 disk sub-system and Gemini GM556 CP/M.

If you are unable to read Gemini QDDS format then you should return the diskette to the supplying Gemini dealer asking for it to be exchanged for:

a) A Gemini SDDS format disk (Nascom with GM805 disk sub-system)

b) A Gemini DDDS format disk (systems with GM815 disk sub-systems)

c) Certain dealers MAY be able to provide other formats.

The files included on the disk are:

GEMTERM.COM
GEMTERM.DAT - The Gemini terminal program

DIAL.COM
DIAL.DAT - A program providing auto-dialling and UART setup procedures.

UKM7.COM &
UKM7.MAC - The CP/M user group terminal program with source file.

TERMB.COM &
TERMB.MAC - A fairly simple terminal program providing only simple terminal emulation, and source file.

AUTOANSW.MAC - This is a simple routine to demonstrate the auto-answer capability of the board.

B. Use with Nascoms

The GM870 design has been optimised for use with the Gemini MultiBoard range of 80-BUS boards. However, due to the similarities of the 80-BUS and Nasbus, it should be possible to use the GM870 in Nascom based systems, although it must be emphasised that this combination has not been tested. Please check the following points.

1. If the board is to be used in Nascom systems then several checks will have to be made before your modem will work. This board provides an open-collector /NASIO signal and this is connected directly to the bus. If any other peripheral boards are connected to your system and they do not provide /NASIO as an open collector output, but rather as a normal TTL output then the other board should be changed so that it no longer generates the /NASIO signal. On all Gemini boards (except 870) this can be achieved by removing a link on the board.

2. With Nascom 1 the I/O internal/external link (Lk1) should be set to external and, because of a decoding error on the Nascom 1, the on-board PIO (IC35) must be removed. With the Nascom 2 the I/O internal/external switch (LSW2,8) should be set for external operation.

3. The GM870 does not provide the Nasbus DBDR signal, and so if the board is to be used with the Nascom 1 then it is necessary that additional circuitry is implemented on the Nascom to establish the required data-bus direction and switch DBDR accordingly.

4. The programs supplied with the GM870 modem board will only run on Nascoms that are running CP/M supporting one of the Gemini video boards (GM812 IVC or GM832 SVC).

Gemini Microcomputers

18 Woodside Road Amersham Bucks HP7 OBH England.