TO: Dave Chandler

FROM: Ali Mostafa *A.M.*

SUBJECT: CHEMICAL BANK TERMINAL BLOCK DIAGRAM

The system block diagram is shown in an attached figure.

The 1610 will be expected to perform the following functions:

1.   Initialize the RAM for clearing the screen.

2.   Output data to the AUTO dial circuit.

3.   Data manipulation and management between peripherals, i.e.

     - Read data from the Keyboard's UART.
     - Send data (when applicable) to send side of Modem's UART.
     - Read received data from Modem's UART.
     - Load received data in HI-RES RAM for display.

4.   Keep track of cursor position.

5.   Process special characters (e.g. generate line feed after receiving carriage return--etc.)

Block diagram description.

1.   PLL, Timing and Control:

     A phase locked loop is essential to double the frequency to be able to generate the HI-RES characters.

     Timing & Control, is to generate all different frequencies need for the system, it also will generate control signals for the HI-RES character generator as all of such signals are related to CBLANK and MSYNC.

2.   HI-RES Character Generator:

     This part will take alot of Real-Estate because of the RAM & ROM.

     Basically it consists of counters (to address RAM and ROM) for data display and an address latch for the 1610 to access the display RAM, an address selector (1610 is given access to RAM only during vertical Flyback)., 1Kx8 screen RAM, 1Kx8 character ROM, Bi-directional data Buffer.
     And shift register for outputing data to EXT-VIDEO PIN

3.   Keyboard UART:

     This section is basically a serial to paralled shift register with random logic for data verification

and for flagging the 1610 when it has a valid data to be read.

4. Modem and Autodial:

This section consists of 3 units;

a. Transmitting unit:

It accepts a paralled data from the 1610 serialize it and adds start and stop bits when appropriate and generates the frequencies need for FSK, amplifies data, and send it to the telephone line.

b. Recieving unit:

It accepts FSK data amplifies and filters it and it feeds these data into a UART for serial/parallel convertion and flags the 1610 with data-available flag.

c. Auto-Dialer Unit:

This unit accepts data (4-8 bit)from the 1610 and feeds it into pulse-generator logic for generating dialing pulses as well as the inter-digit pause and signal is amplified and sent to the telephone line.

The audio signal from the remote site (busy; ringing) is fed to Master Component for audio feedback to the operator.

COMMENTS:

If (for any reason) we have a problem with the die size; I recommend that the HI-RES part can be built with discrete IC's it will be around 8-9 packages.

Currently, the total number of pins is 31, adding 3 for the power supply, I think a 40-pin package would be sufficient (assuming no drastic surprise).

AM/bb

CC: G. Lorenc

CHEMICALBANK

20th Century Project Committee
349 Fifth Avenue (34th Street), New York, NY 10016, Tel: (212) 922-5642

William B. Cornfield
Assistant Vice President

February 9, 1981

Dr. David P. Chandler
Director
Product Engineering
Mattel Toys Division
Mattel, Inc.
5150 Rosecrans Ave.
Hawthorne, California 90250

Dear Dave:

I am enclosing a copy of our preliminary terminal specifications. As we discussed, we have classified each item into one of three categories:

- Requirements for support.

- Requirements for purchase consideration.

- Wants.

These categories are more fully explained in the attached material, however, you should recognize that in many instances that there is a very fine line between wants and requirements. It is possible that a want may become a requirement or a requirement may become a want under the appropriate circumstances. We do not claim to be engineers and we will be pleased to listen to any comments you might have.

If I can be of any further assistance in helping you determine the needs of Chemical Bank, I would be pleased to do so. Please feel free to call on me at your convenience.

I look forward to hearing from you in the near future.

Truly yours,

Bill Cornfield
BP

Enclosure

We have classified our specifications into the following categories:

Required For Support

We do not anticipate supporting terminals on our system which do not meet all of these requirements within the next two years. However, we will not guarantee support for those terminals that do meet these requirements.

Required To Be Considered For Distribution

Terminals which do not meet these requirements will not be considered for purchase by Chemical Bank for distribution to customers.

Wants

These items are considered to be enhancements to the standard terminal. We have rated each of the wants on a scale of 1-10 with 10 indicating a strong desire for that feature and 1 indicating only limited desire.

All terminals will be evaluated based upon their features as well as the following:

- Price
- Quality
- Aesthetic Appearance
- Other Considerations

Display

<u>Requirements For Support</u>

- Minimum display of 32 characters per line and 16 lines per screen.

<u>Requirements For Purchase Consideration</u>

- Minimum display of 40 characters per line and 24 lines per screen.

- TV or built in display.

- Full cursor control.

- Ability to supress display of passwords.

<u>Wants</u>

- Greater than minimum number of characters per line and/or lines per screen (3).

- Upper and lower case display (3).

- Availability of double height characters (1).

- Color capabilities (8).

- Graphics capabilities:

    - low resolution graphics (7)

    - high resolution graphics (7)

- Easily modifiable to "non-U.S." character set (6).

    If display is TV then the following additional wants apply.

    - adaptable to non-U.S. TV's (6)

    - easy to connect to TV (8)

    - choice of channels for display (8)

Communications

Requirements For Support

- A minimum of 300 baud transmission rate capability.

Requirements For Purchase Consideration

- A built in direct access modem.

- A minimum of 300 baud transmission rate capability.

Wants

- A 1200 or greater baud transmission rate capability (10).

- An auto dial modem with at least two software selectable numbers (9).

- Auto answer (3).

- Wireless connection between terminal and phone (1).

- Incoming call warning (7).

- Exclusion of other phones (7).

- Hang up from host (10).

Security

Requirements For Support

None

Requirements For Purchase Consideration

- Capability of storing and transmitting a unique
terminal or cartridge I.D.

Wants

- Data encryption capabilities (8).

Keyboard

Requirements For Support

- Full alphanumeric keyboard.

Requirements For Purchase Consideration

- Full alphanumeric keyboard.

- Function keys as follows:

    - menu
    - previous page
    - next page
    - yes (may be highlighted Y on keyboard)
    - no (may be highlighted N on keyboard)

Wants

- Wireless connection between keypad and keyboard (2).

- Optional handheld keypad with numeric and function keys
  (9).

- Portability (5).

Requirements For Support

None

Requirements for Purchase Consideration

None

Wants

- Support for A printer (7).

- Security systems (5).

- Energy management systems (5).

- Real time clock (5).

- Cable TV (8).

- Bar code reader (2).

- Memory device (7).

Miscellaneous

Requirements For Support

None

Requirements For Purchase Consideration

- Meets FCC shielding specifications both current and proposed.

- Protects users TV from damage including "toasting phosphorus".

- Durable.

Wants

- Family of terminals (5).

- Adaptable to non-U.S. power supplies (6).

- Audio generator (7).

Requirements For Support

None

Requirements For Purchase Consideration

- Ability to store screen masks locally.

- Ability to downline load.

- Ability to store and manipulate data locally.

- Offline games.

Wants

- Ability to communicate in background mode (5).

- Ability to expand memory (6).

- Ability to downline load into nonvolatile memory devices (10).

VIEWDATA SYSTEM SPECIFICATION
AND DESCRIPTION

In reference to Fig. (1); the system consists of a Viewdata module
that plugs directly to the Intellivision's Master Component via a
40 pin connector; on the other end of the module there is a 40 pin
connector where a cartridge may be plugged in. Up to 4K program
can be programmed into such cartridge. With the Viewdata module plugged
into the Master Component; the user may plug either a game cartridge
for game play or a Viewdata cartridge for Viewdata program.

Attached to the Viewdata module a light pen used for data entry; a
Keyboard (Optional) and a modular phone cord for plugging the unit
directly to the phone lines.

Note: That there is no extra power cord needed to power the system
as all power is supplied from the Intellivision Master
Component.

MASTER COMPONENT

VIEW-DATA MODULE

LIGHT-PEN

KEYBOARD

FIG.(1)

In order to make the man-machine interface as simple as possible and to prevent errors from long LOG-IN procedures the phone number and the LOG-IN sequence will be stored in the cartridge PROM; so on power-on and at a touch of a button or a stroke of a Light Pen the unit will dial up the host computer and LOG-IN to the system.

In an interactive mode of operation; the user may use the Light Pen AND/OR the Keyboard to select one of several services offered by the host computer (e.g. Homebanking, Yellow Pages, Ticket Reservation, Mainevent Schedules,...etc.).

On the screen there will be different display modes; where graphics and alpha-numeric data can be displayed simultanously; also different sizes of the alpha-numerics can be displayed (For more details see display Features section).

The Viewdata unit will have a modular cable, which is directly pluggable in the wall phone jack; so there is no acoustic coupler or any extra acessories needed for connection with the phone lines. (For more details see Communications section).

There are two different means of data entry:

    a) Light Pen

    b) Keyboard (Optional)

The Light Pen can be used for menu selection by pointing at a high-lighted spot on the screen corresponding to the selected item. It can be used also for alpha-numeric data entry by pointing at the desired character from a full Keyboard displayed on the lower or upper 4 lines of the screen.

The Keyboard can be used for menu selection by typing in the number corresponding to the selected item from the menu. Also it can be used as a regular Keyboard for alpha-numeric data entry. The following sections will outline the main features of the Viewdata system.

SYSTEM FEATURES

1. Plugs directly to the cartridge port of the Intellivision Master Component.

2. No extra power cord needed for the system as power is supplied from the Master Component.

3. Viewdata module is completely transparent to the Master Component i.e. user can plug a game cartridge and have all the game features without unplugging the Viewdata module.

4. Just by changing cartridges different protocol standards (AT &T, NBC,...) can be met.

5. Directly pluggable into wall phone-jack via a modular plug for direct connection to the phone lines i.e. no acourtic couplers needed.

DISPLAY FEATURES

There are two display modes that can co-exist on the screen. HI-RES
Mode and Color Graphics Mode.

   a) HI-RES Mode

      1. 40 pastel characters/line, 24 lines/screen. On a colored screen.

      2. Full ASCII character set; (5X7 matrix on a 6X8 card.)

      3. A complete AT&T Mosaic Set.

      4. A complete AT&T seperated mosaic set.

      5. Full ASCII character set in a double width mode.

      6. Full ASCII character set in a double height mode.

      7. A double width double height full ASCII character set.

     *8. Two different intensities in displaying the character set
          (Regular and Hi lighted).

     *9. Regular character set can be displayed in reverse video mode.

     10. The regular character set and mosaics can be displayed in a
          Flashing Mode.

    * Due to hardware limitation reverse video and half intensity cannot
      CO-EXIST with double width and double hight on the same screen simultaneous]

   b) Color Graphics Mode

      1. 20 Cards/line 12 ,lines/screen each card is an 8X8 matrix.

      2. Full ASCII colored character set up to 8 different colors can
          be displayed simultaneously.

      3. Very powerful colored graphics capabilities (up to 16 different colors)

      4. 64 Different DRCS (Dynamically redefinable character set). Can
          CO-EXIST on a particular screen.

COMMUNICATION

Communication with the host computer is done over the phone line
directly i.e. no acoustic coupler is needed.

    Features:

    1. Auto dial the host computer at a touch of a button.

    2. Programmable data rate that can match several different
        communication standards.

        e.g.

        a - 300 BAUD, Bell 103 Standard

        b - 1200/75 BUAD, CCITT V.23
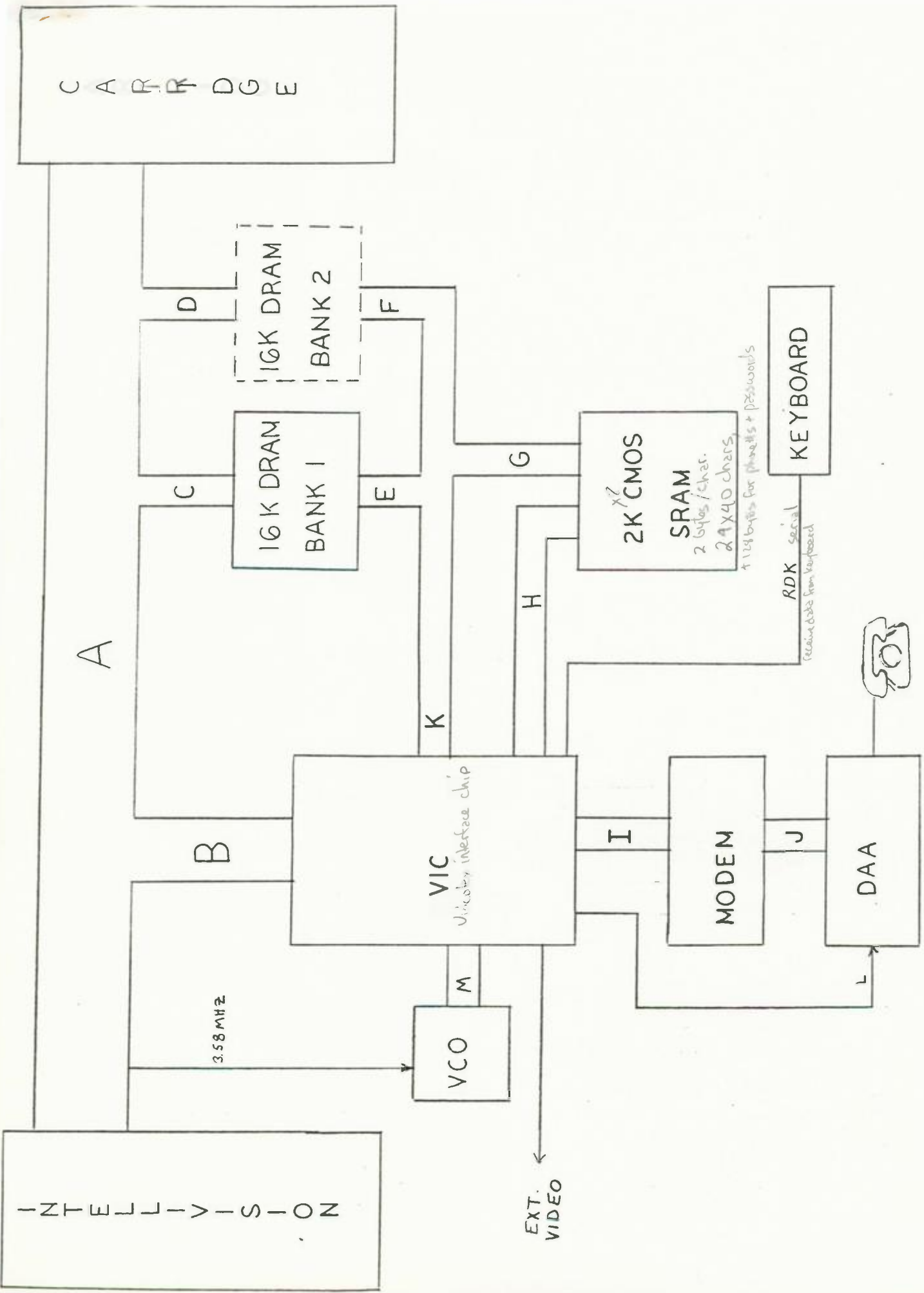
        c - 600/75 BAUD, CCIITT V.23

DATA ENTRY

A. LIGHT PEN

    1. Transparent to the host computer i.e. There is no software overhead at the host computer side needed to handle the Light Pen.

    2. Optimum for menu selection.

    3. Can be used for alpha-numeric data entry by pointing at the desired character from a full type-writter like Keyboard displayed on the screen.

    4. Rugged and very easy to use.

    5. Replaceable.

B. KEYBOARD (OPTIONAL)

    1. Full type-writter Keyboard.

    2. Can be used for menu selection or alpha-numeric data entry.

Don Buska

CARTRIDGE

INTELLIVISION

16K DRAM BANK 1

16K DRAM BANK 2

2K x? CMOS SRAM
2 bytes / char.
24 x 40 chars,
+ 128 bytes for phone #s + passwords

KEYBOARD

RDK serial
receive data from keyboard

VIC
Victory interface chip

MODEM

DAA

VCO

3.58 MHz

EXT. VIDEO

A B C D E F G H I J K L M

PROPOSAL FOR VIDEOTEX TERMINAL COMPONENT

SYSTEM BLOCK DIAGRAM

The system block diagram is shown in an attached figure.
The bulk of the circuitry is in the 40 x 24 hi res alpha
generator block. This block consists of a 6502 uP, 1K x 8
RAM, 1K x 8 ROM, character generator ROM, Intellivision bus
interface and assorted support circuitry. The 6502 will
synchronize to the CBLANK signal and, during the active
display time, step through the RAM to present data to the
character generator ROM. Once each line, a sample of the
modem, keyboard and light pen serial outputs will be taken.
During the vertical blanking time this data will be analyzed
by a software UART function. Communication between the 1610
and the 6502 will also take place during vertical blanking.

The modem block contains the telephone line interface and
filters for the different modes of operation. The telephone
line interface uses a reed relay for pulse dialing, a
transformer for isolation and a simple op-amp circuit for
the hybrid function. For transmit the tones are generated
in the Master Component sound chip and filtered to eliminate
interfering harmonics before being applied to the phone
line. On the receive side, the filters reject local carrier
and noise. Both filters are expected to use switched
capacitor technology; while standard parts are available, a
custom will be cheaper and better.

The light pen interface is a phototransistor and another
transistor providing a one bit input to the 6502.

The remote keyboard interface supplies five volt power to
the keyboard connector and separates the current modulated
data and audio from the line.

The keyboard module itself uses a National MM57499 to scan
the key matrix and serialize the results. Another IC
current modulates the data onto the five volt power line.
There are simpler ways to go with four conductors available,
but doing it this way maintains full compatability with the
Remote Keyboard Component previously proposed; even allowing
this keyboard to be remotely located with ease.

The cartridge contains a 4K x 10 PROM or EPROM (currently under development). A number of unprogrammed addresses will be allocated to allow field customizing per banking requirements. By allowing sufficient space, several updates of this data can be made extending the life of the cartridge. This approach is expected to be less expensive than using EEPROMs.

PHYSICAL CONFIGURATION

The terminal would be housed in the voice peripheral plastic. The keyboard module would be similar in appearance and slightly smaller than the existing Master Component. A model is available for examination.

PRELIMINARY COST ESTIMATES

Cost estimates are attached; be aware that a rule-of-thumb factor (x3) was applied to arrive at the "A" price from a very rough parts list.

PERFORMANCE SPECIFICATIONS

DISPLAY--The display will be our current Master Component graphics overlayed with terminal generated high resolution alpha-numeric characters. Through software, the graphics will be a form of DRCS (Dynamically Redefinable Character Set); compatability with the ATT standard is still being examined. The high resolution alpha-numerics will be 40 characters per line and 24 lines per screen. Full cursor control will be provided by software.

MODEM--The modem will feature direct connection to the phone line through a modular connector and autodialing with both pulse and tone. 300 baud full duplex and 75 baud transmit/1200 baud receive, also full duplex, will be standard. Call progress monitoring (ringing,busy signals) will be provided by routing audio from the phone line to the TV audio system. Under software control, the same routing can be made allowing synthesized voice from the host to be heard by the user

LIGHT PEN--The light pen will be provided as standard. It will be a four foot long fiber optic cable with a quick disconnect plug at the terminal end; this allows the

phototransistor to be mounted on the printed circuit board. This light pen is designed primarily for menu selection, but short alpha strings can be input if a keyboard is displayed on the screen. Through an appropriate protocol, the host establishes the string to be returned for each menu selection.

KEYBOARD MODULE--The keyboard module is optional and would find application only when large amounts of alpha data must be entered (ie electronic mail). To allow easy field upgrade, it will attach to the terminal using a standard modular cable. This cable will supply power and, using the same two wires, return serial data to the terminal. If the keyboard module is the same as that proposed for the Remote Keyboard Component, the audio from the microphone would also be sent to the terminal. This audio, with very few extra components, can provide a speakerphone option. For remote operation, this keyboard can be used in another room with a simple adapter on each end.

POWER SUPPLY--A plug mount power supply will be supplied (may not be needed if CMOS is extensively used).

COMPATIBILITY

VOICE PERIPHERAL--Compatability with the voice peripheral should not be a problem; however, the videotex terminal must be plugged into the Master Component first. This is because the voice peripheral reassigns some of the cartridge pins to its own use.

KEYBOARD COMPONENT--Again, some pins are reassigned forcing the hi res alpha-numerics to come from the Keyboard Component. In addition, few users would opt for the keyboard module when the Keyboard Component already has one. Different software or software that adapted to its environment would solve these problems. The net result is that a large part of the terminal hardware would go unused and the software will be more complicated. An alternate Keyboard Component approach would be to use a modem/light pen module at a rear expansion port with new software.

PLAYCABLE--Assuming that the voice unit is compatible, the videotex terminal should be also. However, a videotex/voice/Playcable combination may have timing problems (due to Playcable's delicate clock regeneration scheme).

FUTURE PLAYCABLE--I see a future Playcable Component

consisting of the videotex terminal (hi res alpha generator, remote keyboard interface and light pen interface, but an rf modem, 16K x 10 RAM and a small PROM (for terminal ID). This unit would allow the cable system to provide videotex and banking services. They could own the system or only provide communications services. The housing would be like the existing Playcable Component. This would create yet another market for the proposed keyboard module. The same remote operation possibilities also exist in this application.

ALTERNATIVES

KEYBOARD COMMUNICATIONS TERMINAL--This would be a unit that consists of a keyboard mounted in a housing about the same size as the existing Master Component and containing most of the circuitry described above. The keyboard interface would be simplified to a parallel interface keyboard scanning IC. It is also possible that the 6502 could handle debouncing the key matrix. The unit would either sit next to the Master Component and plug directly into the cartridge port or sit in front with a short flat-cable connection to the cartridge port. Advantages are somewhat lower cost and one less module if a keyboard is required for all applications. A disadvantage is the slightly awkward side-by-side configuration or the FCC difficult flat-cable.

COMMUNICATING MASTER COMPONENT--In one unit slightly larger than an existing Master Component, the existing Master Component circuit and that described above for the keyboard communications terminal would be combined. Savings would be recognized because of the combined package only; the circuit cost would remain the same. Lower production volumes could offset some or all of the savings. Production problems could result due to longer build time and more dollars tied up in limited marketability units (as compared to other approaches where the Master Component can be shipped to anyone and the terminal component is lower cost). Development time will be longer than other approaches because of larger PC board and case and more design decisions (hand controler mounting, unit appearance and shape factor, etc).

David Hostetler
November 9, 1981

CONTENTS

D R A F T

LIST OF FIGURES

D R A F T

LIST OF TABLES

D R A F T

Unir

subject: Atari Pronto 6520 Terminal
        Emulation Specification
        Case: 000001
        File: 000001

date: June 22, 1982

from: Jim Fleming
      IN
      x317-842-7014

      George Pratt
      IN
      x317-342-7014

1. INTRODUCTION

The Atari P6520A is the Chemical Bank Home Information System
cartridge for the Atari 400 and 800 computers. The cartridge
allows a user to interact with a host computer via a 300 baud
modem (Atari 835 DCM). The P6520A supports both Direct Distance
Dialing and Public Data Networks. The user is insulated from
most of the network and login procedures to simplify the opera-
tion. Several option and help frames are built in the cartridge
to assist the user.

The P6520A cartridge emulates a subset of a TANDEM 6520/6530
block mode terminal. Data is entered into fields and is sent to
the host as blocks of data rather than individual characters.
Local editing capabilities allow a user to enter data into fields
without host intervention. In addition a simple conversational
mode is provided as well as a block-oriented remote control mode
for downloading programs. The cartridge supports the following
features:

- Local and Remote Cursor Control

- User Editing

- Protected and Unprotected Fields

- Multiple Display Pages

- Data Attributes for Unprotected Fields

- Color Control via Video Attributes

1.1 MODES

The Atari P6520A cartridge supports two main user modes. The
LOCAL mode is used for all set-up, dialing, help, and other
internal functions. The ONLINE mode is used during host interac-
tion. The ONLINE mode has three submodes, CONVERSATIONAL SUB-
MODE, BLOCK SUBMODE, and PROTECT SUBMODE.

The CONVERSATIONAL submode is a simple TTY mode where each key that is pressed is immediately sent to the host and single characters received from the host are displayed. The built-in Atari device drivers are used in this mode.

The BLOCK submode provides the ability of complete remote control by the host. A modified version of the INTEL HEX format is used by the host to read and write Atari memory locations. Because the I/O in the Atari is memory-mapped, complete I/O control is allowed.

The BLOCK submode also allows the host to read and write the Non-Volatile RAM (NOVRAM) contained in the terminal. This RAM appears in the memory space of the Atari.

The BLOCK submode is selected in the same manner as the Block mode of the 6520. The HEX format protocol is compatible with the 6520 in a limited sense. Downloaded programs will appear as ASCII characters when sent to the 6520. Obviously no I/O or actual execution can be tested using a 6520 terminal.

The PROTECT submode is the main mode used during a terminal session. A subset of the 6520 Protect submode features have been implemented.

The PROTECT submode is block-oriented and user entry is associated with fields. The fields restrict a user to data entry in only certain parts of the screen. Multiple screen memories are provided so that common displays can be selected without having to resend the data. Numerous control sequences are implemented so that the host can manipulate the fields as well as the multiple pages of screen information.

Facilities are provided to allow the host to switch between the various ONLINE submodes. The terminal defaults to the CONVERSATIONAL submode when initially brought online.

1.2 KEYBOARD

The operation of the various keys on the keyboard is dependent on the current mode. In LOCAL mode most of the keys are disabled except those keys needed to manipulate the menu-oriented services.

In the CONVERSATIONAL submode each key is sent to the host when it is hit. No display occurs in CONVERSATIONAL mode unless characters are sent from the host.

The keyboard is locked while in the BLOCK submode. The host can scan the keyboard via memory reads but should only do so for special applications.

The PROTECT submode utilizes the full capability of the Atari keyboard. Information is sent to the host in blocks when special function keys are hit. The standard alphanumeric keys are used to enter information into the screen based fields but do not cause any transmission to the host. The editing keys (arrows, insert, delete, etc.) are used by the user to change the screen based fields.

The yellow keys on the Atari are used in a manner similar to the TANDEM 6520 function keys. Each key produces a unique ESCape sequence which is sent to the host. The lower level protocol controls when data can be sent to the host.

1.3 VIDEO SCREEN FORMAT

The TANDEM 6520/6530 Terminal uses a 24 row by 80 column display with a 25th line for status and messages. The Atari personal computer can support a 24 by 40 column display without resorting to cumbersome user scrolling.

In order to maintain compatibility, the Atari will respond to 6520 commands as if 80 columns were available. Characters sent to the terminal will be "undisplayed" if sent to columns 41 to 80. No memory of those characters will be kept. If those columns are involved in a read operation the columns will contain spaces.

User editing will be modified so that "wrap around" will occur after column 40. Note that if the host sends cursor positioning commands that result in moving the cursor to an undisplayed area the cursor will disappear. If text is sent after such a command it will be lost up until the point where the cursor is repositioned to a displayable area. Note also that this applies to buffer addressing as well as cursor positioning.

2. LOWER LEVEL PROTOCOLS

All communication between the terminal and the host will be via a 300 baud asynchronous 8 bit data link. The 8 bits are divided into 7 data bits and 1 parity bit. Parity is ignored by the terminal and is not generated. The lower level protocols vary depending on the submode.

2.1 CONVERSATIONAL SUBMODE

The CONVERSATIONAL submode utilizes single 8 bit characters. Characters are displayed as soon as they are received and are sent as soon as a key is pressed. Functions such as flow control and error checking are not provided.

2.2 BLOCK AND PROTECT SUBMODES

The BLOCK submode and PROTECT submode both use the same lower
level protocol. A variable length block-oriented protocol is
created using the standard ASCII control characters. The proto-
col is a half-duplex protocol with all control coming from the
host. Error control and time-outs are provided to help maintain
link integrity. The protocol conforms to the TANDEM 6520 Point-
to-Point protocol.

The Protocol can best be described by a series of typical tran-
sactions. In each of these transactions, the TEXT information is
assumed to be variable length and can contain BLOCK submode or
PROTECT submode information.

TABLE 1. SET BLOCK MODE

```
|HOST ->                                                 <- TERMINAL|
|                              EOT->                                |
|                     SOH,'B',ETX,<LRC> ->                          |
|                              <- ACK                               |
|           *** Terminal Transmission Disabled ***                  |
|                                                                   |
```

Table 1 illustrates the exchange that occurs to set the BLOCK
submode. This can occur while the terminal is in CONVERSATIONAL
submode.

TABLE 2. WRITE TO SCREEN

```
| HOST ->                                                <- TERMINAL|
|                             EOT ->                                |
|              STX,' ',...<TEXT>...,ETX,<LRC> ->                    |
|                             <- ACK                                |
|              STX,'!',...<TEXT>...,ETX,<LRC> ->                    |
|                             <- NAK                                |
|              STX,'!',...<TEXT>...,ETX,<LRC> ->                    |
|                             <-.ACK                                |
|              STX,'"',...<TEXT>...,ETX,<LRC> ->                    |
|                             <- ACK                                |
|                             EOT ->                                |
|                                                                   |
```

Table 2 shows a typical transfer that occurs when the screen is
updated. The NAK that is shown would be sent if an error had
been detected.

The character following the STX is a sequence number used to
identify individual blocks. The first block (after the EOT)

---

* The notation SOH,'B',ETX,<LRC> -> indicates that the first
  character transmitted is the SOH and the last is the <LRC>
  even though the arrow (->) might indicate otherwise.

contains a SPACE (octal 40) for the sequence number. Each time a
new block is sent the sequence number is incremented. If an
error occurs the sequence number would not be incremeted for the
retransmitted block. In the example in Table 2 the second block
with a sequence number of octal 41 (!) was repeated to insure
proper ordering when the terminal receives the data.

Note that only ONE outstanding block is allowed. Therefore, even
if a block is successfully sent it will have to be retransmitted
if a previous block has been sent in error. This is evident from
the fact that the NAK indicating an error does not identify which
block was in error. If the host attempts to send a second block
before receiving the ACK for the first block it must be responsi-
ble for synchronizing the ACK or NAK responses with its transmis-
sion. If an error is detected, the terminal will NAK the block
and discard any blocks which arrive without the proper sequence
number. A NAK is sent for the blocks that are discarded because
of a bad sequence number.

The data flow from the host to the terminal will always take the
highest priority. If terminal is in the process of sending a
block to the host and the host sends a block to the terminal the
host-to-terminal block will be received in a normal manner. Once
this block is processed the block that was being sent from the
terminal to the host will be retransmitted.

All timing will be done by the host. If the host is waiting for
an ACK and a timeout occurs the block should be resent. If the
terminal receives a block with a sequence number that has already
been acknowledged a NAK will be sent.

TABLE 3.   READ FROM SCREEN

```
|HOST ->                                                          <- TERMINAL|
|                                  EOT ->                                     |
|                         .        ENQ ->                                     |
|                      *** Terminal Transmission Enabled ***            .     |
|             <- SOH,<FUN. KEY>,<PAGE>,<ROW>,<COL>,ETX,<LRC>                  |
|                        .         ACK ->                                     |
|                                  <- EOT                                     |
|                      *** Transmission Disabled ***            .            |
```

Table 3 is an example of a Read from the screen. The ability of
the terminal to send to the host is determined by the lower level
protocol in most cases. As shown, the ENQuire that is sent
allows the terminal to respond to the host.

As will be seen later, sometimes the contents of the TEXT buffer
controls whether the terminal can transmit to the host. These
are typically READ commands and must be the only command in the
TEXT buffer. This allows the command to be easily detected by
the lower level protocol. So much for layered protocols. Table

TABLE 4. READ DATA FROM FIELD

```
|HOST ->                                              <- TERMINAL|
|                          EOT ->                               |
|            STX,' ',..<READ COMMANDS>...,ETX,<LRC>             |
|                 *** Transmission Enabled ***                 |
|                   <- ACK MAYBE !!!!!                          |
|            <- STX,' ',...<TEXT>...,ETX,<LRC>                  |
|                          ACK ->                               |
|            <- STX,'1',...TEXT...,ETX,<LRC>                    |
|                          ACK ->                               |
|                          <- EOT                               |
```

4 shows an example of a TEXT buffer command that controls the
transmission to the host. The MAYBE on the ACK is emphasized
because the TANDEM documentation does not indicate that an ACK is
sent. If an ACK is not sent then a very inconsistent protocol
has been designed. The ommision of the ACK is assumed to be a
misprint until proof can be obtained via live testing.

TABLE 5. SET CONVERSATIONAL SUBMODE

```
| HOST ->                                    <- TERMINAL|
|                      EOT ->                           |
|             SOH,'C',ETX,<LRC> ->                      |
|                      <- ACK                           |
|          *** Full Duplex Enabled ***                  |
```

When the terminal is in either BLOCK submode or PROTECT submode
the terminal can be placed in CONVERSATIONAL submode as shown in
Table 5.

Note that when the terminal is in CONVERSATIONAL submode and PRO-
TECT submode is desired the host must first set BLOCK submode
(with SOH,B,ETX,LRC) and then select PROTECT submode (using ESC
W).

3. CONVERSATIONAL SUBMODE

In CONVERSATIONAL submode individual characters are the basic
unit of transmission. As mentioned, when keys are hit a data
byte is sent and individual characters received are passed to the
Atari E: device driver.

If a "control" character is preceeded by an ESC (033) character
then the E: device driver will ignore the normal "control" func-
tion and will display the control character as data. Appendix A
enumerates the Atari display codes and the special functions
associated with some of the codes. Appendix B lists the codes
that are transmitted when keys are pressed on the Atari. Note

that the yellow keys have no function in CONVERSATIONAL submode.

4. BLOCK SUBMODE

When the terminal is in the BLOCK submode the terminal memory (and I/O) can be manipulated by the host processor. This mode will be used to read and modify the NOVRAM, download animation and/or music sequences, modify the colors and display lists, etc. The entire Atari memory space can be manipulated opening up an unlimited number of host controlled extensions. The host can send the Atari off to an unknown state if everything is not handled properly. This is the risk that must be taken to allow this future flexibility.

The protocol for manipulating the Atari memory conforms to the standard INTEL HEX format used in many microcomputer systems. The protocol uses the ASCII character set and can therefore be handled by most application programs on the host system. No free form binary data is transmitted or received. Because of the use of ASCII, binary data bytes must be sent and received as two ASCII characters in HEX format.

```
                        Data Length
               _____
               ||
               ||
               ||             Record Type
               ||           _____
               ||          ||                                          Checksum
               ||          ||                              _____
               ||          ||                              ||
Colon--->:LLaaaaTTddddddddddddddddddddddddddddddddCC
               |  |  |                                |
               |  |  |_____Data Field_____|
               |  |
               |  |
               |__|___Address Field
```
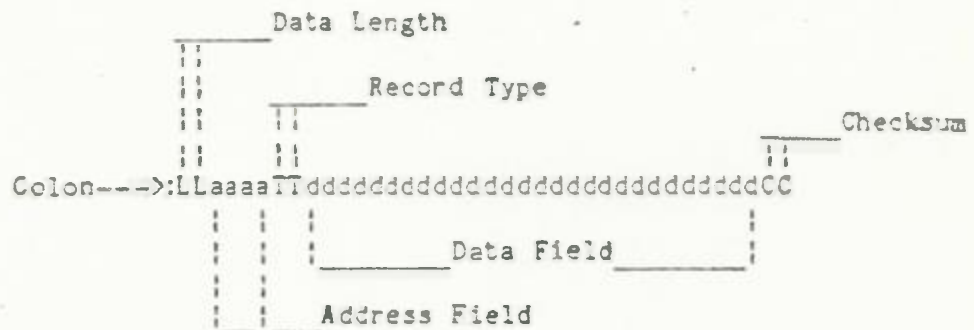
Figure 1.   BLOCK SUBMODE HEX FORMAT

Figure 1 illustrates a sample HEX data stream. As can be seen the format allows for variable length records containing address, command, and data. The command or Record Type field has been extended beyond the standard INTEL HEX format to provide needed functionality.

HEX commands in the BLOCK submode are made up of a series of records. As shown each record begins with a colon (: octal 072). Since records can contain variable length data fields the next field contains the Data Length. The length is encoded as two ASCII characters in Hexidecimal notation. For example, a length of 8 would be '08', 12 would be '0C' and 34 would be '22'. For commands that do not require a Data field the Data Length will be '00'.

Following the Record Length is the Address Field. Because the format was originally intended only for downloading data into memory locations this field normally specifies the starting location of memory to begin loading the variable length data field. The Address Field is interpreted differently for the extended commands. Note that since the Address Field is sent with each record, memory can be loaded in a random or sequential manner. The Address Field is encoded as four ASCII characters so that a 16 bit address can be specified. Addresses in the I/O area of the Atari are allowed and therefore sequences of hardware registers can be modified with one record.

The Record Type field normally contains the ASCII characters '00' when writing to memory. New commands have been added to the Record Type field to extend the functionality of the HEX format.

TABLE 6.   HEX FORMAT RECORD TYPES

VALUE	COMMAND	ADDRESS USAGE	DATA USAGE
'00'	Load Memory	Starting Address	Valid Data
'01'	Jump to Address	Jump Address	Empty
'10'	Set Address	Address to be Saved	Empty
'12'	Read Memory	Ending Address	Empty

Table 6 contains a list of the valid Record Type field values and the command along with an indication of how the Address field and Data fields are interpreted.

The Load Memory command has already been described. The Set Address command is used to set an internal address register used in other commands. No memory is modified (other than the Atari memory needed to implement the feature).

The Read Memory command is used to have the terminal send the contents of memory to the host. The response to the host is in HEX format with a Record Type field of '12' coresponding to the Read Memory command. The contents of memory starting at the previously stored address (see Set Address) to the ending address specified in the command are sent to the host. Records will be blocked in a convienent manner. Note that if the starting address is greater than or equal to the ending address then wrap around will occur after location 65535 to 0 until the ending address is reached. If random areas of memory are needed then several read commands are necessary.

The Jump to Address command is the most powerful and dangerous of the commands. The 6502 processor will jump to the 16 bit address specified by the address field of the record. A valid program should be stored into memory using the Load Memory command before a Jump is sent. Global resets can be acheived by jumping to the Atari warm start addresses.

The Checksum at the end of each record is usually used for error control. The two ASCII characters represent an 8 bit binary value which is the two's complement of the sum of the binary values of all of the fields (except the colon). Note that the ASCII values are not used to determine the sum but rather the binary values that the ASCII pairs represent.

The Checksum is ignored in the current implementation because the lower level protocol provides error control. The Checksum can be eliminated to improve efficiency. Records are delimited by the colon and the record length. Extra characters except mode control commands are ignored between records.

```
Down Load To Location 0000
        :10000000D4B021D8E241625056222E5081509C2C0F
Down Load To Location 0120
        :090120007523647717761740FA86
Jump to Address 0010
        :00001004FF
```

Figure 2. SAMPLE HEX FORMAT STREAM (With Checksums)

Figure 2 illlustrates a simple download and jump sequence. Note that the Address Field contains an address without the bytes reversed.

5. PROTECT SUBMODE

The PROTECT submode is the main operational mode of the terminal. One and one half dimensional "fields" can be set up in multiple places on the screen for user data entry. User data is entered into the fields without host intervention. Once the data is entered it is sent to the host as a block with an indication as to which field was used for the entry.

5.1 Attributes - Data and Video

Each field can have attributes associatted with it when it is set-up. Data attributes determine what type (numeric, alpha, etc.) of data can be entered into the field. Indicators can also be set to specify whether a field is protected from user entry and how the field should be handled when data has to be sent to the host. Video attributes are used to specify how characters entered into the field should be displayed.

Data attributes are not retroactive. Any field oriented activity uses the current data attributes established for the field. These attributes can be modified by the host.

The TANDEM 6520 and 6530 terminals have special video hardware to allow underlining, reverse video, blinking, etc. for designated

areas (or fields) on the screen. This hardware is such that by changing one character in an area all of the characters following the changed character (called the video attribute character) will take on the new attribute.

The Atari 400/800 video display hardware does not support this mode of video attributing. It is true that the Atari supports reverse video or blinking characters (mutually exclusive) but not in the 6520/6530 serial video attribute manner. The Atari 400/800 supports what is often called parallel attributes. A single bit is reserved for each character to determine whether reverse video (or blinking) is active.

In order to utilize the Atari reverse video and maintain compatibility with the 6520/6530 protocol a subset has been chosen. The concept of 6520/6530 video attributes and single character control should be completely ignored. Instead the video attribute of a field should be treated more like a data attribute. It is a logical attribute that affects any writing that occurs in a field. In order to activate the video attribute it must be specified in conjunction with a field. No screen change will occur when the attribute is specified. Only subsequent write operations will be done with the new attribute.

This interpretation of video attributes prevents the ability to create instant screen changes via single byte updates. Instead another capability is added which is not possible in the 6520/6530 terminals. A field in the P6520A can now contain some reverse video characters as well as normal characters because each character has its own reverse video control bit (the parallel attribute bit). In the 6520/6530 a change in the video attribute for a field results in all of the characters changing which is sometimes not desirable.

By removing the concept of hardware oriented video attributes, the Set Video Prior 6520/6530 becomes useless. As will be seen a new Atari oriented Set Color command has been added while still maintaining TANDEM syntax compatibility.

5.2 CONTROL CODES

Table 7 summarizes the ASCII control codes recognized by the terminal. All other control codes will be ignored or treated as NUL. The valid control codes are described in the following sections.

5.2.1 Fill Character (octal 00)
Used with some commands as a time delay to avoid buffer overrun.

5.2.2 Sound Audible Alarm (octal 07)
The BEL code sounds the terminal's audible alarm.

TABLE 7. PROTECT SUBMODE CONTROL CODES

CODE	ASCII NAME	USAGE
00	NUL	Fill Character for Time Delay
...		
07	BEL	Sound Bell
10	BS	Backspace
11	HT	Horizontal Tab
12	LF	Line Feed
...		
15	CR	Carriage Return
21	DC1	Set Buffer Address
...		
23	DC3	Set Cursor Address
...		
35	GS	Start Field
...		

(handwritten: 8, 9, in margin of BS, HT rows; 27 esc pointing to GS row)

5.2.3 Backspace (octal 010) *(handwritten: ← slightly different from what it already has in PVD)*
The Backspace code moves the cursor one space to the left. This
is a nondestructive move. If the cursor was in column 1, it will
go to column 80 of the previous line. If the cursor was in
column 1, row 1, it will go to column 80, row 24. If the cursor
does not end up in an unprotected field, it will continue to
search backward for the first element of an unprotected field
which it encounters. (This element will be the last element in
the field.)

5.2.4 Horizontal Tab (octal 011)
The HT code causes the cursor to move forward to the beginning of
the next unprotected field. It will wrap around to the top if no
fields are found at the bottom.

5.2.5 Line Feed (octal 012)
The Line Feed code causes the terminal to move the cursor down
one line but remain in the same column. If the cursor is placed
outside of an unprotected field it will advance (right and down)
until it finds the first location of an unprotected field.

5.2.6 Carriage Return (octal 015)
The CR operates the same as the Horizontal Tab (octal 011). The
cursor is moved to the next unprotected field.

**5.2.7 Set Buffer Address (octal 021 <row> <col>)**
The Set Buffer Address code is followed by two bytes. These two
bytes determine where in the selected page the next updates will
occur. The two bytes are in row, column order and are offset by
octal 037 from the actual address.

O **5.2.8** Set Cursor Address (octal 023 <row> <col>)
The Set Cursor Address code is followed by two bytes. These two bytes determine where the cursor is in the currently displayed page. The two bytes are in row, column order and are offset by octal 037 from the actual address.

O **5.2.9** Start Field (octal 035 <video> <data>)
The Start Field code defines the start of a field and its video and data attributes. If the Start Field code address equals the start address of an existing field, the attributes for that field are modifed; otherwise a new field is created. If the transmitted attribute is outside the specified range, the normal (disabled) state of the attribute(of the attribute)s used. The format for the video and data attributes is shown below:

byte 1 (video)

6	5	4	3	2	1	0
0	1	0	H	R	0	0
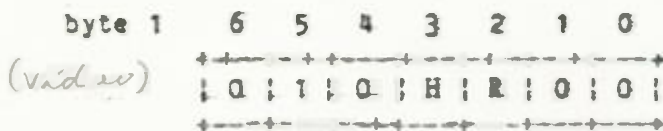
H: 1 = Hidden,   0 = Normal
R: 1 = Reverse,   0 = Normal

byte 2 (data)

6	5	4	3	2	1	0
1	P	0	0	0	0	T
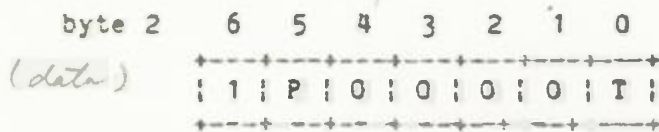
P: 1 = Protected,   0 = Unprotected
T: Modified data tag

The Hidden data attribute replaces the Blank 6520/6530 video attribute. The new Hidden Attribute indicates that when a user enters data into the field the '*' character should be echoed so that other people present in the room can not see the information that was typed. This is used in password entry. The terminal will keep track of the actual data that was entered in case a read or send occurs.

The Reverse video attribute is similar to the 6520/6530 Reverse attribute. Characters entered by the user or sent from the host after the attribute is set will be displayed with the foreground and background colors reversed.

The Unprotected field attribute is used to indicate that a field is protected from user modification. The data from an unprotected field is returned to the host when requested.

The Modified Data Tag is used to indicate that data in a field has been modified.

If the field is unprotected the MDT is set whenever the user modifies (or adds) data to a field. If the field is protected

the MDT can be set by the host at the time the field is set-up.

Whenever the host issues a Read With Address the terminal will
send all fields that have the MDT set. With this facility the
host can have the terminal return information from protected
fields as well as unprotected fields. The host can set the MDT
to 0 when the field is set-up (or afterwards) so that the con-
tents of the field will not be sent.

The Auto-Tab Disable normally found on a 6520/6530 has been omit-
ted. This feature is used on systems where key-entry personnel
are constantly interacting with the same forms. Residential
users will not realize it is missing.

The Auto-Tab Disable feature affects where the cursor is posi-
tioned when a field is filled. The normal default for a
6520/6530 when a field is filled would be to move the cursor to
the next unprotected field. Since the Auto-Tab Disable feature
is ignored the handling of this situation has been changed. In
order to be consistent with past PRONTO systems, the cursor will
wrap back to the beginning of the field when the field is filled.
The user will be required to hit either Tab or Return to move to
the next field.

5.3 ESCAPE SEQUENCES

Appendix C lists the Escape sequences recognized in the PROTECT
submode. Some of the sequences are interpreted in a slightly
different manner than the normal 6520/6530 usage. Some 6520/6530
sequences are completely ignored in a graceful manner.

5.3.1 Set Color (ESC 7 <color>)
The background color and character color can be controlled as a
global screen attribute. The normal 6520 Set Video Prior Condi-
tion will be used for color control.

Five bits of information are transmitted in the Set Video Prior
Condition (ESC 7 <color>) sequence. The five data bits normally
indicate video attributes such as dim, blink, reverse, blank, and
underscore. The five data bits allow for eight colors (3 bits)
and one of four luminances (2 bits). The data is encoded as fol-
lows:

```
+----+----+----+----+----+----+----+----+
| X | 0 | 1 | C  C | C | L L |
+----+----+----+----+----+----+----+----+
```
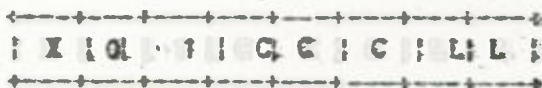
The color specification bits (CCC) will be loaded into the upper
3 bits of the Atari Color-Lum Register. The luminance bits (LL)
will be loaded into the 3rd and 4th bits of the Atari Color-Lum
Register.

The one's complement of the luminance value will be used for the character luminance so that contrasting characters are displayed. The mapping is compatible with the one and one half color capability of the Atari home computer.

The three color bits (CCC) can be used to select any one of the following background colors:

TABLE 8.   BACKGROUND COLORS

000	Grey
001	Orange
010	Pink
011	Purple-Blue (Magenta)
100	Blue
101	Turquoise (Cyan)
110	Green
111	Orange-Green

The luminance bits can be used to vary the intensity of the selected color.

5.3.2   Select Page (ESC : <page>)

The Select Page escape sequence allows the host processor to select a page for future modification via Buffer Addressing. The character following the (ESC :) specifies the page number. The character is offset by octal 040 from the page number. Page 1 is selected by the character octal 041. The maximum number for a page is a function of page size and memory. If a page greater than the maximum is specified the command is ignored and the page stays the same. The default page number when entering BLOCK submode is 1.

5.3.3   Display Page(ESC ; <page>)

The Display Page escape sequence allows the host processor to specify which page is displayed. The character following the (ESC;) specifies the page number. This character is offset by 040 from the page number. Page 1 is displayed by the character octal 041. The maximum number for a page is a function of page size and memory. If a page is greater than the maximum is specified the command is ignored and the page stays the same. Page 0 (octal 040) will cause the screen to blank-fill and lock the keyboard.

5.3.4   Reset Modified Data Tags (ESC >)

The Reset Modified Data Tags (ESC >) escape sequence resets the modified data tags of all unprotected fields of the selected page. This escape sequence has no effect on modified tags set by the host in a protected field.

5.3.5 One Second Delay (ESC @)
The One Second Delay (ESC @) escape sequence causes the terminal to stop processing the input stream for approximately one second. At the end of the delay, normal processing continues.

5.3.6 Clear To Spaces (ESC I <row1> <col1> <row2> <col2>)
The Clear To Spaces escape sequence blank-fills a specified area of a selected page of memory. The four characters following the ESC I specify the starting and ending buffer address. These addresses are of the form row, column and are represented by ASCII characters. The characters are found by adding octal 037 to the row (1-24) and the column(1-40).

The blank-filling will not cross page boundaries. If the cursor is positioned in a field to be blank-filled, it will be positioned to the next (searching right and down) unprotected address on the selected page. If no unprotected positions remain on the page, the cursor is not displayed. The current buffer address is not effected by this operation.

5.3.7 Erase To End Of Page (ESC J)
The Erase to The End of Page escape sequence blank-fills all character positions in the page starting from the current buffer address and ending at the page boundary. The position of the cursor and buffer address are not altered by this operation. Only unprotected character positions are changed.

5.3.8 Erase To End Of Field (ESC K)
The Erase to End of Field escape sequence blank-fills all character positions in a field starting from the current buffer address to the end of the field. The position of the cursor and buffer address are not altered by these operations.

5.3.9 Enter PROTECT Submode (ESC W)
Enter PROTECT submode or exit BLOCK submode causes the terminal to begin interpreting the TEXT buffer as PROTECT submode sequences. If a Read Memory is in progress when this command is received, it is terminated at the next convienent record end.

5.3.10 Exit PROTECT Submode (ESC X)
Exit PROTECT submode or enter BLOCK submode causes the terminal to begin interpreting the TEXT buffer as HEX format (BLOCK submode) commands. The keyboard is locked and the host takes complete control. No PROTECT submode ESCape sequences are recognized except the Enter PROTECT Submode (ESC W) sequence. CONVERSATIONAL submode can be entered with the proper sequence (SOH,'C',ETX,<LRC>).

5.3.11 Unlock Keyboard (ESC b)
The Unlock Keyboard escape sequence immediately unlocks the keyboard. The user may now enter keyboard data to be displayed on the screen, any transmission to the host processor is held until the terminal receives an ENQ from the host.

5.3.12 Lock Keyboard (ESC c)
The Lock Keyboard escape sequence immediately locks the keyboard.
When the keyboard is locked, the cursor is no longer displayed.
The keyboard can be locked by any of the following:

- Depression of a function key

- Display of page 0

- Transmission to/from the displayed page

While the keyboard is locked, instead of the audible keyboard
"click" on key depression, the terminal sounds an audible bell.

5.3.13 Simulate Function Key (ESC d <byte1>)
The Simulate Function Key escape sequence simulates depression of
a keyboard funtion key. The character immediately following the
escape code designates the particular code associated with the
funtion key. For example, an (ESC d B) sent to the terminal
causes the terminal to generate:

> SOH,'B',<page>,<r umber>,<row>,<col>,ETX,<LRC>

The transmission of this sequence will not occur until an ENQ has
been sent by the host.

5.3.14 Modem Disconnect (ESC f)
The Modem Disconnect escape sequence forces the 835 Direct Con-
nect Modem to hangup.

5.3.15 Set Max Page Number (ESCp <byte1>)
The Set Max Page Number escape sequence defines the maximum
number of display pages available. The character following the
escape code defines the maximum page number. This character,
offset by octal 060, starts with page 1 (061) and continues to
060 + n, the maximum number of pages available. The maximum
number of pages depends on the size of the RAM memory in the ter-
minal. Sending a value greater than the terminal can handle will
default to the maximum the terminal can handle. The maximum page
number does not take effect until a switch to BLOCK submode
occurs, or the Reinitialize (ESC q) escape sequence completes.

5.3.16 Terminal Reset (ESC q)
The Terminal Reset escape sequence executes an initialization
sequence similar to that executed upon entering BLOCK submode
except: Page 1 is blank-filled, and the communication line is not
reinitialized.

5.3.17 Cursor Up (ESC A)
The Cursor Up escape sequence results in moving the cursor to the
first column of the previous unprotected field. Positioning the
cursor to an unprotected field simply results in moving the cur-
sor up one line to the previous line; no scrolling occurs. If

the previous unprotected field is located to the right of the
current cursor position, the terminal positions the cursor to the
first column of the previous unprotected field located to the
left of the current cursor location.

5.3.18 Cursor Right (ESC C)
The Cursor Right escape sequence will cause the terminal to move
the cursor one column to the right. In a protected field the
cursor will continue to move right until it encounters an unpro-
tected field.

5.3.19 Cursor Home Down (ESC F)
The Cursor Home Down escape sequence, will cause the cursor to be
placed in the first column of line 24 of the current page. If
this is in a protected field the cursor will be placed in the
first column of the last unprotected field on the current page.

5.3.20 Cursor Home Up (ESC H)
The Cursor Home Up escape sequence, will cause the cursor to be
placed in the first column of the first row. If this position is
in a protected field then the cursor is place in the first posi-
tion of the first unprotected field.

5.3.21 Back Tab (ESC i)
The Back Tab escape sequence, will cause the cursor to move to
the first column of the unprotected field it is in. If it is
already in the first column of an unprotected field then the cur-
sor will be moved to the first column of the previous unprotected
field. If no previous field exists on the current page the cur-
sor will continue to search starting at the lower right hand
corner of the current screen.

page ? screen ?

5.3.22 Read Buffer (ESC <)
The Read Buffer escape sequence, will cause the terminal to
transmit the unprotected fields to the host. The text will con-
sist of the data stored in the unprotected fields, regardless of
whether the field has been modified. Trailing spaces are not
transmitted. If the field contains only spaces, the field
separator (DC1 and 2-byte address) is the only data transmitted.
The transmission starts with the first unprotected field and con-
tinues through the last unprotected field on the display page. A
typical data stream is shown in Figure 3. The row and column
addresses are encoded in the manner used to set cursor and buffer
addresses.

5.3.23 Read With Address (ESC = <row1> <col1> <row2> <col2>)
The Read With Address escape sequence is followed by four bytes
which contain the encoded starting and ending address. The row
and column bytes are offset by octal 037. Fields which start
(attribute + 1) at or after the starting address are transmitted.
All unprotected fields before the ending address are sent. Any
field which starts at or before the ending address is sent.
(Note: If the starting address and/or the ending address equals

```
        DC1                 Set buffer address
        <ro w >             Address of first unprotected
        <col>                  field (attribute + 1)
          .
          .
        <text>
          .
          .
        DC1                 Set buffer address
        <row>               Address of second unprotected
        <col>                  field (attribute + 1)
          .
          .
        <text>
          .
          .
        etc.
```

Figure 3.   READ BUFFER EXAMPLE

the first element of an unprotected field (attribute + 1) that
field is sent. Any unprotected field in between is also sent.)
The terminal will send the fields in the same format as the Read
Buffer command.

5.3.24 Read Terminal Status (ESC^)

The Read Terminal Status escape sequence will cause the terminal
to send its status to the host processor. The format of the data
returned is shown below:

SOH '?' <byte1> <byte2> <byte3> <byte4> <byte5> <byte6> CR

```
        byte 1    6   5   4   3   2   1   0
                +---+---+---+---+---+---+---+
                | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
                +---+---+---+---+---+---+---+


        byte 2    6   5   4   3   2   1   0
                +---+---+---+---+---+---+---+
                |    TERMINAL IDENTIFIER     |
                +---+---+---+---+---+---+---+


        byte 3    6   5   4   3   2   1   0
                +---+---+---+---+---+---+---+
                |   FIRMWARE REVISION LEVEL  |
                +---+---+---+---+---+---+---+
```

```
byte 4    6  5  4  3  2  1  0
         +--+--+--+--+--+--+--+--+
         | 1| 0| MAXIMUM PAGE NUM. |
         +--+--+--+--+--+--+--+--+


byte 5    6  5  4  3  2  1  0
         +--+--+--+--+--+--+--+--+
         | 1| FIELDS PER PAGE (MSB) |
         +--+--+--+--+--+--+--+--+


byte 6    6  5  4  3  2  1  0
         +--+--+--+--+--+--+--+--+
         | 1| FIELDS PER PAGE (LSB) |
         +--+--+--+--+--+--+--+--+
```
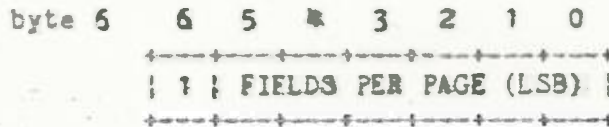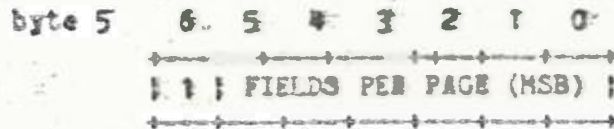
5.3.25 Read Cursor Address(ESC a) ✓

The Read Cursor Address escape sequence causes the terminal to
transmit the display screen's cursor address to the host proces-
sor.

    SOH,'_',<page>,<row>,<col>,ETX,<LRC>

        <page>:   Page number offset by (octal 040).
        <row>:   Current cursor row offset by (octal 037)
        <col>:   Current cursor column offset by (octal 037)

            Figure 4.   READ CURSOR ADRESS EXAMPLE

The message will take the form shown in Figure 4.

6. Summary

The P6520A cartridge allows for a substantial number of the TAN-
DEM 6520/6530 terminal features to be used on an Atari personal
computer. With this cartridge most TANDEM based applications
programs can be run with no change to TANDEM support packages or
device drivers. This compatibility preserves large software
bases already in place without requiring expensive data terminals
to be placed in residential settings.

                                        Jim Fleming

                                        George Pratt

APPENDIX A - ATARI DISPLAY CODES

Codes 00-7F							
00	Heart	20	Blank	40	@	60	Diamond
01		21	!	41	A	61	a
02		22	"	42	B	62	b
03		23	#	43	C	63	c
04		24	$	44	D	64	d
05		25	%	45	E	65	e
06	/	26	&	46	F	66	f
07		27	'	47	G	67	g
08		28	(	48	H	68	h
09		29	)	49	I	69	i
0A		2A	*	4A	J	6A	j
0B		2B	+	4B	K	6B	k
0C		2C	,	4C	L	6C	l
0D		2D	-	4D	M	6D	m
0E		2E	.	4E	N	6E	n
0F		2F	/	4F	O	6F	o
10	Clubs	30	0	50	P	70	p
11		31	1	51	Q	71	q
12		32	2	52	R	72	r
13		33	3	53	S	73	s
14	o	34	4	54	T	74	t
15		35	5	55	U	75	u
16		36	6	56	V	76	v
17		37	7	57	W	77	w
18		38	8	58	X	78	x
19		39	9	59	Y	79	y
1A		3A	:	5A	Z	7A	z
1B	Esc	3B	;	5B	[	7B	Spade
1C	/	3C	<	5C		7C		
1D	/	3D	=	5D	]	7D	Clear Screen
1E	<-	3E	>	5E	^	7E	Back Space
1F	->	3F	?	5F	_	7F	Tab

Codes 80-FF	
80	
.	Reverse
.	Video
.	of
.	Codes 0-7A
8A	
8B	EOL
8C	Delete Line
8D	Insert Line
8E	Clear Tab
8F	Set Tab
90	
.	More
.	Reverse
.	Video
FC	
FD	Bell
FE	Delete Char
FF	Insert Char

Code	Key	Code	Key	Code	Key	Code	Key
00	CTRL @	20	Blank	40	@	60	CTRL ;
01	CTRL A	21	!	41	A	61	a
02	CTRL B	22	"	42	B	62	b
03	CTRL C	23	#	43	C	63	c
04	CTRL D	24	$	44	D	64	d
05	CTRL E	25	%	45	E	65	e
06	CTRL F	26	&	46	F	66	f
07	CTRL G	27	'	47	G	67	g
08	CTRL H	28	(	48	H	68	h
09	CTRL I	29	)	49	I	69	i
0A	CTRL J	2A	*	4A	J	6A	j
0B	CTRL K	2B	+	4B	K	6B	k
0C	CTRL L	2C	,	4C	L	6C	l
0D	CTRL M	2D	-	4D	M	6D	m
0E	CTRL N	2E	.	4E	N	6E	n
0F	CTRL O	2F	/	4F	O	6F	o
10	CTRL P	30	0	50	P	70	p
11	CTRL Q	31	1	51	Q	71	q
12	CTRL R	32	2	52	R	72	r
13	CTRL S	33	3	53	S	73	s
14	CTRL T	34	4	54	T	74	t
15	CTRL U	35	5	55	U	75	u
16	CTRL V	36	6	56	V	76	v
17	CTRL W	37	7	57	W	77	w
18	CTRL X	38	8	58	X	78	x
19	CTRL Y	39	9	59	Y	79	y
1A	CTRL Z	3A	:	5A	Z	7A	z
1B	ESC	3B	;	5B	[	7B	CTRL [
1C		3C	<	5C		7C	
1D		3D	=	5D	]	7D	CTRL]
1E		3E	>	5E	^	7E	CTRL /
1F	3F	?	5F				

ESC 7	Set color
ESC :	Select page
ESC ;	Display page
ESC >	Reset modified data tags
ESC @	One second delay
ESC I	Clear to spaces
ESC J	Erase to end of page
ESC K	Erase to end of field
ESC W	Enter protected submode
ESC X	Exit protected submode
ESC b	Unlock keyboard
ESC c	Lock keyboard
ESC d	Simulate function key
ESC f	Modem disconnet
ESC p	Set max page number
ESC q	Terminal reset
ESC A	Cursor up
ESC C	Cursor right
ESC F	Cursor home down
ESC H	Cursor home up
ESC i	Back tab
ESC <	Read buffer
ESC =	Read with address
ESC ^	Read terminal status
ESC a	Read cursor address

ADP-MATTEL HOME BANKING PRODUCT COMPONENTS

A. VIDEOTEX PERIPHERAL

    -   PLUG INTO ROM-CARTRIDGE SLOT OF MASTER UNIT
    -   300 BAUD INTEGRAL TO VIDEOTEX PERIPHERAL
    -   AUTODIAL FROM KEYBOARD

B. MASTER UNIT (INTELLIVISION II)

    -   HOME BANKING AND COMMUNICATIONS SOFTWARE IN REMOVABLE ROM PACK

    -   VIDEOTEX AND KEYBOARD PERIPHERALS REQUIRED FOR BANKING

    -   ALTERNATIVE BANKING SERVICE WITH CONTROLLER

C. SYSTEM PACKAGING

    -   1 POWER PLUG
    -   1 TELEPHONE JACK
    -   1 WIRE TO TV
    -   ATTRACTIVE STYLING

D. KEYBOARD CHARACTERISTICS

    -   FULL ALPHANUMERIC SET IN STANDARD LAYOUT

    -   RAISED KEYCAPS FOR POSITIVE TOUCH

    -   CONVENIENT TO USE CURSOR CONTROL KEYS

    -   PROGRAMMABLE FUNCTION KEYS, PREDESIGNATED KEYS FOR SPECIAL
        OPERATIONS (AUTODIAL, HELP, ETC.)

E. DISPLAY CHARACTERISTICS

    -   40 CHARACTER WIDTH
    -   20 ROWS MINIMUM
    -   ADDRESSABLE CURSOR
    -   CHARACTER BACKGROUND/FOREGROUND COLOR SELECTABLE
    -   HOME TV DISPLAY
    -   SPECIAL GRAPHICS

F.   EXPANSION (OPTIONAL)

     -    SYSTEM CHANGER (GAMES ONLY)
     -    INTELLIVOICE (GAMES ONLY)

G.   SOFTWARE AVAILABILITY

     -    CARTRIDGE GAME PACKS OVER THE COUNTER
     -    SPECIAL BANKING APPLICATIONS THROUGH BANKS/ADP

H.   MAINTENANCE

     -    FULL SERVICE LOCATIONS IN ALL MAJOR CITIES
     -    WARRANTY PROGRAM

I.   MARKETING

     -    MAJOR BASE OF UNITS IN FIELD

     -    MATTEL TO ASSIST IN PACKAGING AND ADVERTISING OF BANKING
          SERVICES

     -    FOLLOW-ON PRODUCT TO AQUARIUS HOME COMPUTER SYSTEM

3/22/83
ALA/st

ADP-MATTEL HOME BANKING

<u>OBJECTIVE</u>

STANDARDIZE BANKING ENVIRONMENT TO REACH MAXIMUM MARKET

BANKS, MERCHANTS

ADP/TCS SERVICE CENTER

ADP BANKING SYSTEM HOST

NETWORKS

TRANSACTION COLLECTION (Standardized Protocol)

MATTEL HOME ENTERTAINMENT/COMPUTER PRODUCTS

ADP-MATTEL HOME BANKING
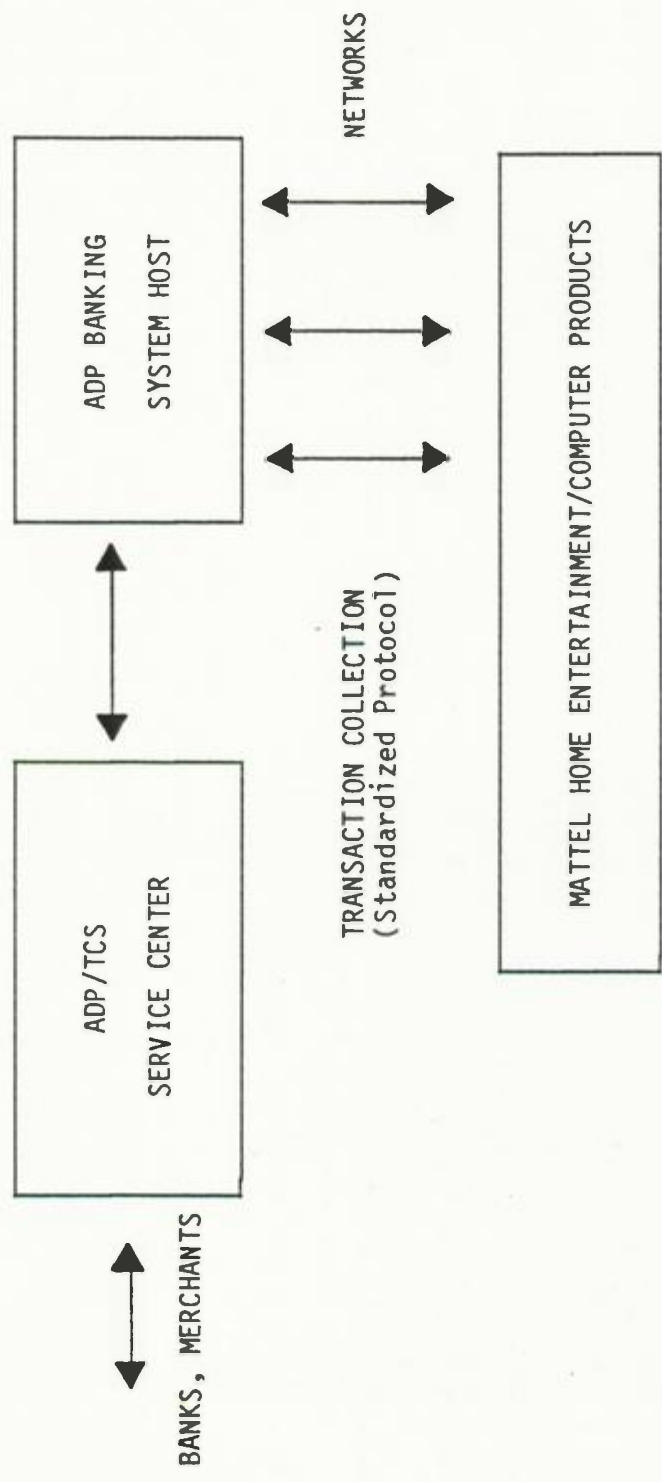
OBJECTIVE

MAXIMIZE USE OF HOME COMPUTER INTELLIGENCE AND FEATURES

NETWORK

MATTEL
PRODUCT
TYPES

(INTELLIVISION II)

ADP COMPUTER SYSTEM
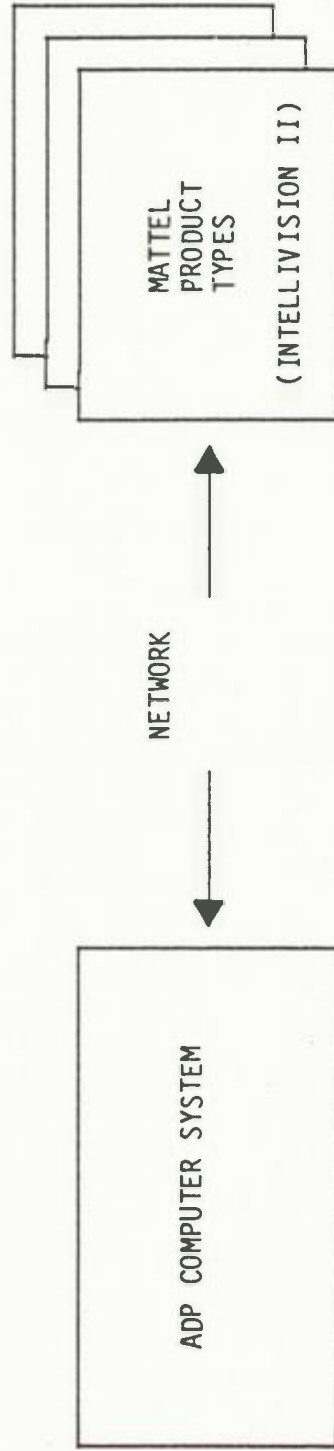
HOME COMPUTER SUPPORT PACKAGE

o USER LOG-ON AND ACCOUNTING

o BANKING DATA BASE ACCESS

o TRANSACTION COLLECTION AND
  LOGGING

o STANDARDIZED BANKING SESSION
  PROTOCOL

HOME COMPUTER-RESIDENT SOFTWARE

o GENERAL PURPOSE O/S

o COMMUNICATION DRIVERS AND
  STANDARD PROTOCOL SESSION
  FOR BANKING

o FORMAT BANKING DISPLAYS -
  GRAPHICS/COLOR TECHNIQUES

o ACCESS BANKING INFORMATION
  FROM HOST, PROCESS TRANSACTIONS
  TO HOST

3/22/83
ALA/st

ADP/MATTEL HOME BANKING PRODUCT IMPLEMENTATION
(Preliminary)

PRODUCT SPECIFICATION

SESSION PROTOCOL
NETWORK INTERFACE
FUNCTIONAL USE

DEVELOP AND TEST PROTOTYPE HARDWARE
(MATTEL)

FINAL ASSEMBLY FOR RESEARCH SAMPLE
(MATTEL)

PROTOTYPE BANKING SERVICES
(ADP)

BANKING APPLICATION PROGRAMMING
(MATTEL)

INTEGRATE PROTOTYPE BANKING DISPLAYS
(MATTEL)

FUNCTIONAL TEST OF MATTEL BANKING
PRODUCT (assembly performance)

ASSEMBLE BANKING CARTRIDGES

PLACEMENT OF TERMINALS TO RESEARCH
SAMPLE

BEGIN TEST

ALA/st
3/22/83

ADP

BANKING SESSION DATA TRANSFER

TERMINAL INQUIRIES:

(1.1)   REQUEST SERVICE

        DEVICE ID, LEVEL OF SERVICES,
        CONFIGURATION, BANK TRANSIT AND
        ROUTING, ENCRYPTION PARAMETER

(1.2)   ACCOUNTS AND BALANCES

(1.2)   PAYEES FOR BILL PAYMENT

(1.2)   HISTORY DETAIL FOR SELECTED ACCOUNT

(1.4)   POSTING

        TRANSACTION CODE, PAYEE OR ITEM CODE,
        AMOUNT, DATE TO PAY, SPECIAL STATUS

HOST RESPONSE:

ACKNOWLEDGEMENT, VALIDATION OF COLLECTED USER ID
AND PIN DATA

ACCOUNT TYPE, ACCOUNT NUMBER, STATUS, OPTIONS,
USER SELECTED NAME, BALANCE

INTERNAL PAYEE CODES, USER SELECTED NAME, CLEARED OR
UNCLEARED STATUS, DATE AND AMOUNT OF LAST PAYMENT

TRANSACTION DATE, AMOUNT, DESCRIPTION, DEBIT/CREDIT
STATUS, TRACE NUMBER, CLOSING DATE FOR STATEMENT

ACKNOWLEDGEMENT

ALA/st
3/22/83

ADP

MATTEL INTELLIVISION TERMINAL ENVIRONMENT	ADP BANKING HOST ENVIRONMENT

USER AUTODIAL AND
LOG-ON TO NETWORK,
ADP HOST

⟶

HOST ACKNOWLEDGES CONNECT,
REQUESTS TERMINAL ID

⟵

(1.1)

TERMINAL ACKNOWLEDGES,
PROVIDES BANKING
SECURITY FRAME

⟶

HOST SWITCHES CONTROL TO
BANKING APPLICATIONS

- -

USER RESPONDS TO
REQUESTED ACCOUNT,
PIN

⟶

HOST CHECKS ACCOUNT AND PIN,
SENDS VALIDATED RESPONSE TO
TERMINAL WITH ACCOUNTS AND
BALANCES

(1.2)

⟵

(1.3)

USER SELECTS
BANKING SERVICES
FROM MENU

⟵

HOST PROVIDES IN BLOCK DATA
FORM: (A) PAYEE DATA, OR
      (B) HISTORY DATA
NOTE: THIS STEP TO BE MODIFIED
      FOR FUTURE SOFTWARE
      SUPPORTING TERMINAL
      RESIDENT FILES.

TRANSACTION BUILDING
IN TERMINAL

(1.4)

USER EXITS BANKING

⟶

HOST RECEIVES TRANSACTION DATA
IN BLOCK FORMAT, RETURNS
ACKNOWLEDGEMENT TO TERMINAL

⟵

- NETWORK APPLICATION - - - -

TERMINAL LOG-OFF
FROM NETWORK

3/22/83
ALA/st

Product Description

Mattel Electronics
Intelletex Module

Don Buska   April 1983

INDEX

1. GENERAL DISCRIPTION

Mattel Electronics Intellitex terminal (fig.1) is a low cost add-on module for the Intellivisions Master Component. It is inserted between the game cartridge port and the users cardridge, either a game or terminal operation cartridge maybe placed into the Intellitex cartridge port. Using different terminal operation cartridges allow Intellitex to act as a ASCII (American Standard for Communications Information Interchange) or NAPLPS (North American Presentation Level Protocol) data terminal. Additional cartridges can be developed for other communications techniques.

Data entry is buy the Intellivision hand controllers, using a menu available on the television screen, or by a optional typewriter style keyboard.

Connection to the phone line is via standard modular phone cord.

Power for Intellitex is derived entirely from the Intellivision Master Component. Display memory back-up is provided by a battery, to retain memory content on power down.

~~Game cartridges can still be played, using Intellitex cartridge slot.~~

Figure 2 shows an internal block diagram of Intellitex, with Individual component ciscriptions given in the Hardware Description section.

MASTER COMPONENT
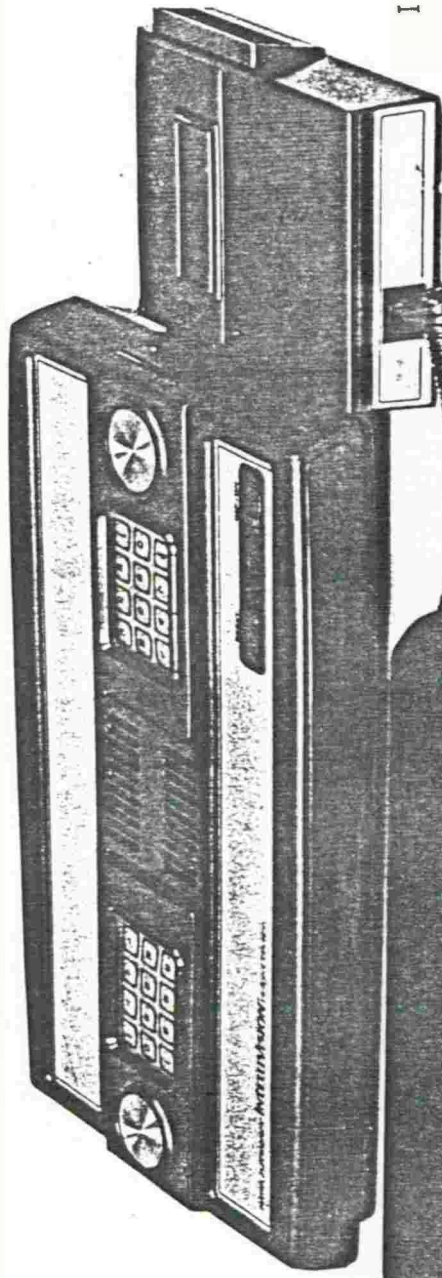
INTELLITEX

KEYBOARD

(FIG.1)

2. HARDWARE DESCRIPTION

2.1 Videotex Interface Chip (VIC)

Central control of the Intellitex Module is accomplished by the VIC (NMOS VLSI) integrated circuit. VIC will provide a external video signal, to combine both Intellitex video and Intellivisions graphics abilities.

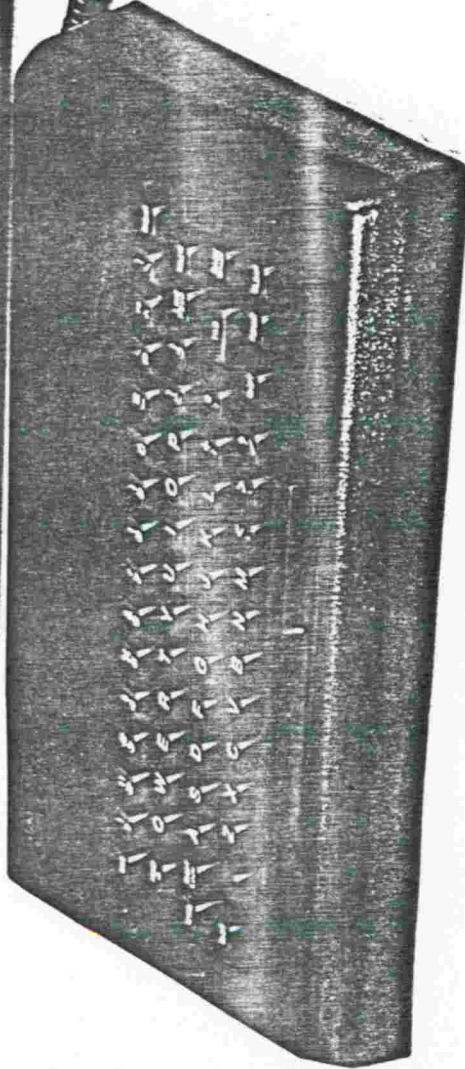VIC provides the following features:

-Video timing to allow generation of 40 character x 24 line display capabilities on a standard television screen.

-A Universal Asynchronous Receiver/Transmitter (UART), containing two receivers and one transmitter.

-UART parity detection (odd/none)

-UART data word consists of: 1-start, 8-data, 2-stop bits.

-UART programmable for 75, 150, 300, 600 and 1200 baud rates.

-Character ROM for full ASCII and mosaics sets.

-Generates Double-height, Double-width, Reverse and Flashing video on character sets.

-Dynamic RAM controller for two 16K banks of memory.

-Provides all address decoding for Intellitex module.

-TTL compatable input/output.

2.2 Voltage Controlled Oscillator (VCO)

The VCO generates the 5.369317 MHZ signal, required by VIC, from the 3.579545 MHZ MCLK generated by Intellivision. Dividing the VCO signal by 1.5 is done in VIC, to provide the phase comparing frequency FOUT.

2.3 Display Memory

Character sets, and their associated attributes, must be stored in memory, to be continuously displayed. This display memory resides in a 2Kx8 CMOS logic family Static Random Access Memory (SRAM) integrated circuit. A small amount of remaining memory area is reserved for phone numbers and passwords, used in auto-dialing and login.

Battery back-up is provided on this memory to retain information during power down.

2.4 Operation Memory

When need arrises, for additional memory, 16 or 32K of Dynamic Random Access Memory (DRAM) maybe added. Control for DRAM is already provided by VIC.

Possible uses include game download area, and additional page memory.

2.5 Modem

In order for the telephone line to carry asynchronous data, leaving and entering the VIC, a modem is used to convert TTL level binary data to audio tones suitable for transmission.

Since the modem is apparently destined to be of the single chip variety, it's baud rate capabilities will largely be a factor of cost and availability. VIC's UART and modem interface have been designed with a wide selection of single chip modems in mind. More information with regard to modem chip selection can be found in the Product Goals section.

2.6 Direct Access Arrangement

A modem must connect to the telephone line through a device called a Direct Access Arrangement (DAA). An FCC-registered DAA is needed when connecting into the public telephone lines. This usually consists of a few cheap components; transformer, operational amplifier IC. Devices of this type have been used in other Mattel Electronics products.

2.7 Keyboard (optional)

An alternative to the hand-controller and menu method, of data entree, is a typewriter style keyboard. Much more ease in operation, of intellitex, will be noticed with the keyboard option attached. Connection is made using a standard stereo plug, and coiled cord. An eleven bit serial data word is sent to the VIC, exactly the same as from the modem.

3. SOFTWARE

Each Intellitex unit will be supplied with a terminal operation cartridge. This software package determines which service the terminal recognizes; NAPLPS, ASCII, etc.

3.1 NAPLPS

The North American Presentation Level Protocol Syntax is a method for encoding visual information in a standard and compact manner, which can then be exchanged among a variety of different computer systems. Using a method of sending information called code extension, allows fast and efficient communications. It appears, at this time, that NAPLPS will be a standard for videotex data transfer.

The following is a list of NAPLPS functions supported:

		YES	NO	PARTIAL
3.1.1 Text				
	ASCII		*	
	Mosaics		*	
	Separated Mosaics			*
	Supplementary Graphics			*
	DRCS (64 per screen)			*
3.1.2 PDI (Picture Discription Instruction)				
	Control			*
	Geometric Primitives			*
3.1.3 Control Sets				
	C0		*	
	C1			*

For a more detailed analysis of supported NAPLPS see document North American Presentation Level Protocol Supported by Intellitex.

3.2 ASCII

Perhaps the easiest supported information interchange which Intellitex can handle is ASCII. A full character set is stored in VIC, allowing a fully interactive computer terminal with 40 character/24 line display.

4. PRODUCT GOALS

Intellitex, when combined with Intellivision and a color television , can be used as a low cost communications terminal. Changing software cartridges allow handling different informatic standards.

Intellitex Information Capabilities:

-Videotex (home-banking, home-shopping, etc.)

-Computer services (Source, Compuserve, etc.)

-home computer video terminal (serial RS-232)

(START)

Display Program title + copyright notice
Actual or Tutorial Session?
Request phone number from user
Auto dial + sign on to network
SEND DEVICE ID + CONFIGURATION
SELECT HOME BANKING APPLICATION

APR 22 1983

(Tutorial Session)

STEP USER THROUGH
VARIOUS MENUS USING
SIMULATED DATA

SET ENCRYPTION PARAMETER
DISPLAY BANKING SECURITY FRAME

(RETURN TO START)

User enters personal ID + security code
Allow corrections, when complete, transmit
to host for verification

< VALID USER? > NO

YES

HOST DOWNLOADS ACCOUNT BALANCES, BILLS OUTSTANDING
ACCOUNT HISTORY FOR PAST MONTH

Display ACCOUNT BALANCES
AND BANKING MENU

(MENU)

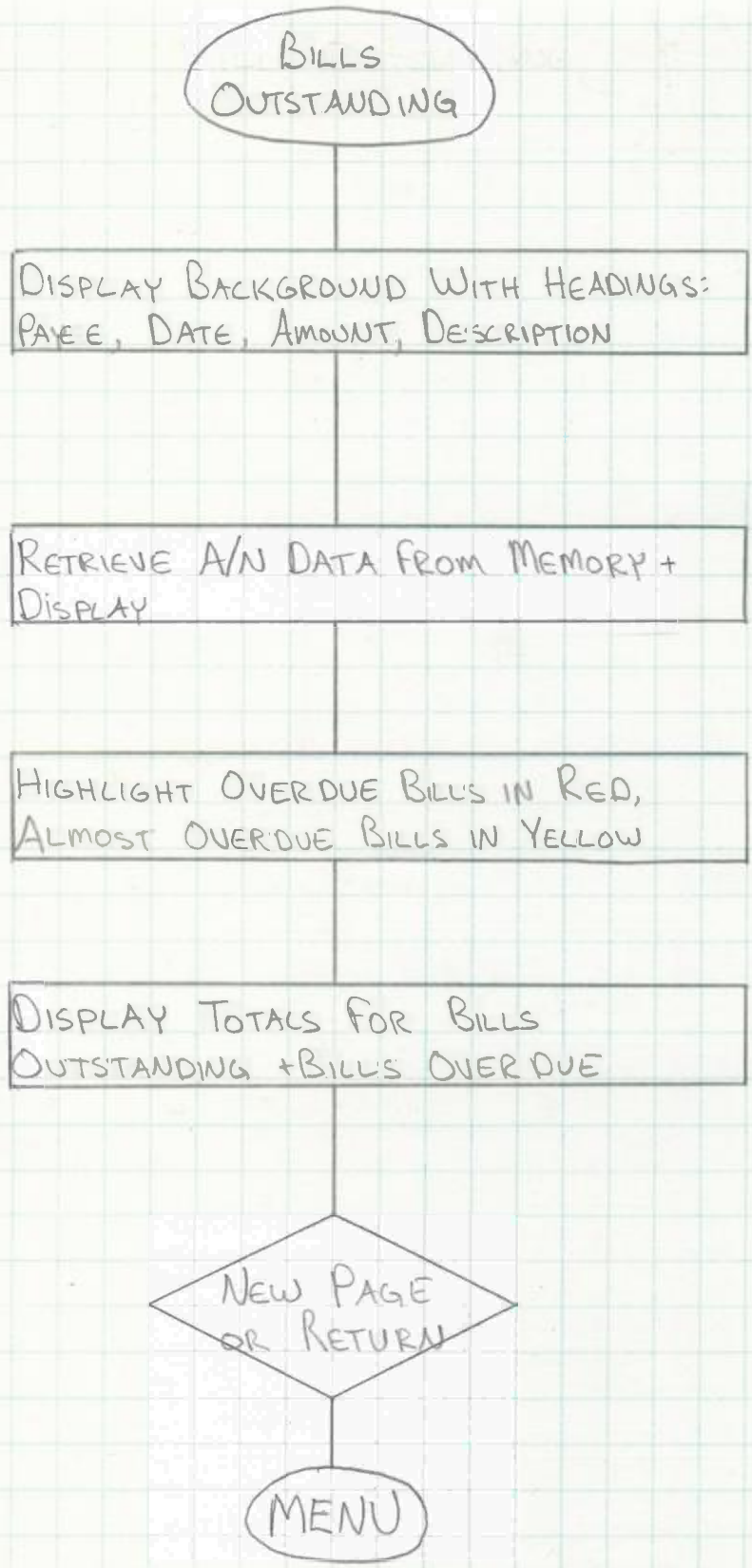USER SELECTS BANKING SERVICE FROM MENU

(BILLS OUTSTANDING)   (BILL PAYMENT)   (FUNDS TRANSFER)   (ACTIVITY REVIEW)   (CHANGE PASSWORD)   (EXIT BANKING)

```
        ┌─────────────┐
        │    BILLS    │
        │ OUTSTANDING │
        └─────────────┘
               │
   ┌───────────────────────────────────────────┐
   │ DISPLAY BACKGROUND WITH HEADINGS:         │
   │ PAYEE, DATE, AMOUNT, DESCRIPTION          │
   └───────────────────────────────────────────┘
               │
   ┌───────────────────────────────────────────┐
   │ RETRIEVE A/N DATA FROM MEMORY +           │
   │ DISPLAY                                   │
   └───────────────────────────────────────────┘
               │
   ┌───────────────────────────────────────────┐
   │ HIGHLIGHT OVERDUE BILLS IN RED,           │
   │ ALMOST OVERDUE BILLS IN YELLOW            │
   └───────────────────────────────────────────┘
               │
   ┌───────────────────────────────────────────┐
   │ DISPLAY TOTALS FOR BILLS                  │
   │ OUTSTANDING + BILLS OVERDUE               │
   └───────────────────────────────────────────┘
               │
           ╱───────────╲
          ╱  NEW PAGE   ╲
          ╲  OR RETURN  ╱
           ╲───────────╱
               │
           ┌────────┐
           │  MENU  │
           └────────┘
```

NOTE:
   THIS MODULE WILL NOT BE
   INCLUDED IN THE INITIAL
   SOFTWARE VERSION

May 24, 1983

Mr. Richard Tuthill
Director
Intellivision Products
Mattel Electronics
5150 Rosecrans Avenue
Hawthorne, California   90250

Dear Richard:

Based upon our most recent discussions and internal planning within
ADP Telephone Computing Service, Inc., (ADP/TCS) I am enclosing an up-
dated schedule which hopefully can serve as the basis for our moving
forward on the home banking product with Mattel Electronics.

The schedule does reflect some delay to the original plan with Mattel,
but is now strategically placed to introduce the first group of 50
terminals to friendly users subsequent to our having successfully placed
the same number of terminals with presentation level protocol (PLP) cap-
abilities in a friendly user test environment (October through November
1983). Keeping to this schedule, a full product roll-out within the HOME
BANKING INTERCHANGE™ (HBI) project and other business opportunities now
being pursued by ADP/TCS would be planned for the March timeframe.

Please review the major milestones and associated timeframes indicated,
and provide me with any comments or any adjustments as required.

ADP/TCS and the financial institutions participating in our home banking
project place a high value in Mattel's participation, and our commitments
must be fully understood pursuant to a contract agreement between Mattel
and ADP/TCS at this time. In this regard, I have requested that
Ms. Pamela Stock send out the initial draft of the agreement now under-
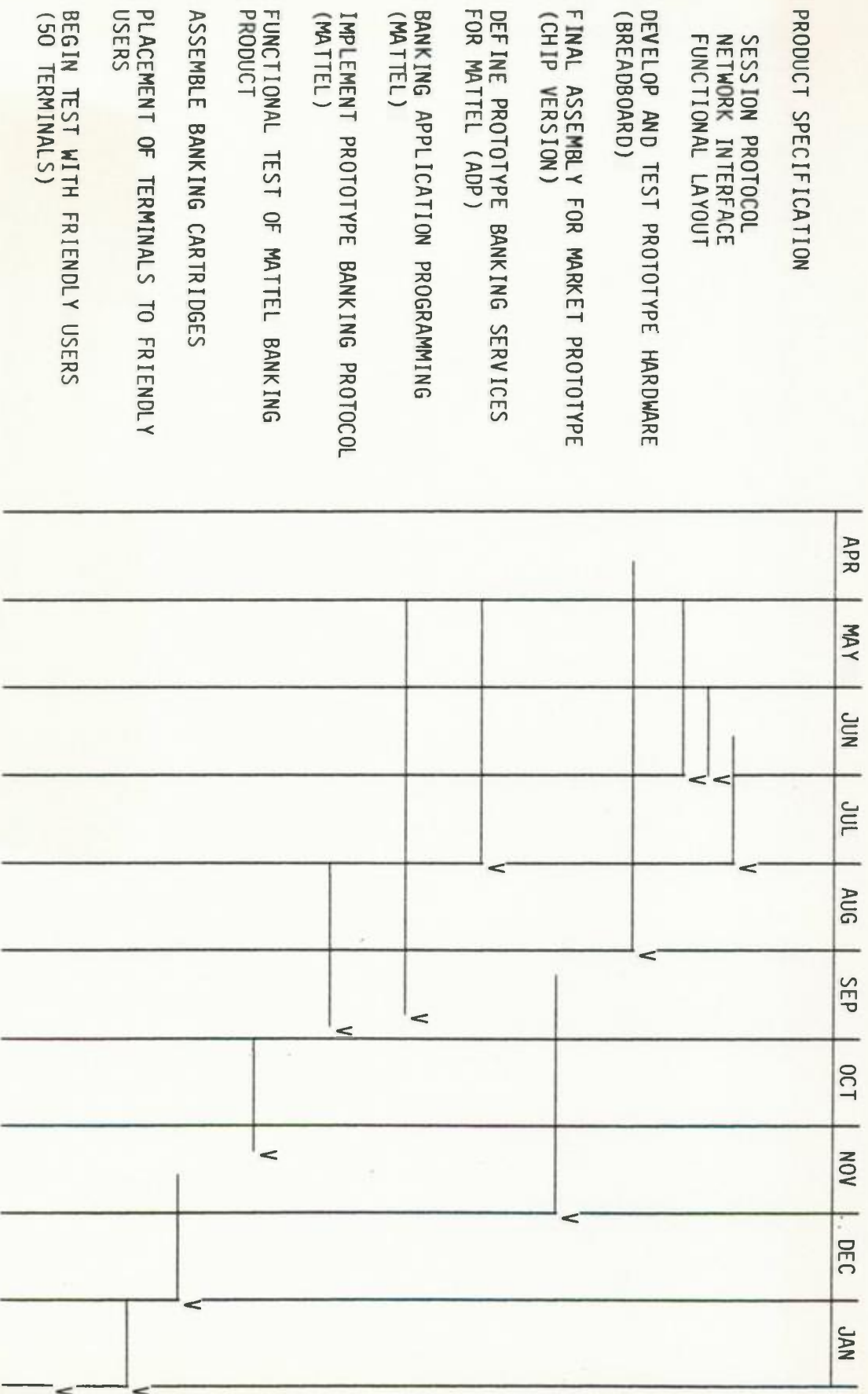stood to be prepared.

Regards,

A. L. Ayres
Vice President

ALA/st
Enclosure
cc: Dr. David P. Chandler
     Mr. Jim Marziani

ADP/MATTEL HOME BANKING PRODUCT IMPLEMENTATION

PRODUCT SPECIFICATION

SESSION PROTOCOL
NETWORK INTERFACE
FUNCTIONAL LAYOUT

DEVELOP AND TEST PROTOTYPE HARDWARE
(BREADBOARD)

FINAL ASSEMBLY FOR MARKET PROTOTYPE
(CHIP VERSION)

DEFINE PROTOTYPE BANKING SERVICES
FOR MATTEL (ADP)

BANKING APPLICATION PROGRAMING
(MATTEL)

IMPLEMENT PROTOTYPE BANKING PROTOCOL
(MATTEL)

FUNCTIONAL TEST OF MATTEL BANKING
PRODUCT

ASSEMBLE BANKING CARTRIDGES

PLACEMENT OF TERMINALS TO FRIENDLY
USERS

BEGIN TEST WITH FRIENDLY USERS
(50 TERMINALS)

APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	JAN