REV	DESCRIPTION	DATE	APPR
A	INITIAL RELEASE PER PCN		

PRELIMINARY

Document #_____

Eng. _____ Mkt. _____

PRO-LOG CORPORATION

JIM MC DONALD 5-8-80

LITERATURE, USER'S MANUAL - 7301

RS-232 AND TTY DRIVER/RECEIVER CARD

A	106420	REV A	SHT 1 OF 62

TABLE OF CONTENTS

SECTION 1 — Product Overview
- Block Diagram

RS-232 Introduction
- Microprocessors and RS-232
- RS-232 Characteristics Summary
- Control Signal Handshake
- Potential System Incompatibilities

SECTION 2 — Functional Description
- Card Address Mapping
- RS-232 Section
- 20 mA TTY Circuit
- 7301 Reset Characteristics
- Spare TTL I/O Lines

SECTION 3 — Card Address Mapping
- I/O Port Address Selection
- Address Decoder Jumper Table

SECTION 4 — RS-232 Signal Options and Programming
- General
- RS-232 Signal Options
- RS-232 Programming
- RS-232 Cable Interlock

SECTION 5 — TTY Programming and Configuration Options
- General
- TTY Programming
- TTY Cable Interlock
- Tape Reader Control (ASR consoles)
- Alternate configurations: Electronic data terminal
  Instrument interface

SECTION 6 — ASCII Code Information
- Format
- ASCII Character Set
- Code Table
- Baud Rate Timing
- Programming Considerations

SECTION 7 — 7301 Electrical Specifications

SECTION 8 — 7301 Mechanical Specifications

SECTION 9 — 7301 Operating Firmware

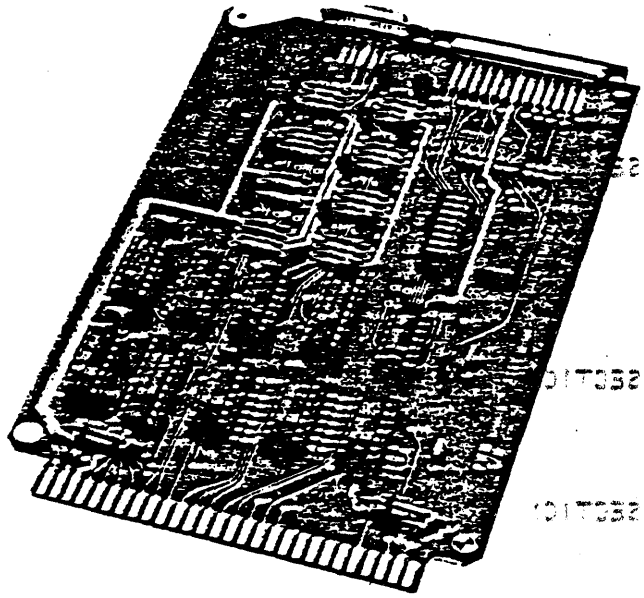
APPENDIX A: ASR-33 Modification for Paper Motion Control

7000 STD BUS 7301 RS-232 & TTY DRV/REC CARD

The 7301 combines the I/O ports and the voltage translation needed to interface a microprocessor to RS-232-C and TTY serial data communications lines. The RS-232-C and TTY circuits feature separate control ports for programming convenience, and cable interlocks which can be tested. The microprocessor card used in conjunction with the 7301 is programmed to provide timing and serial/parallel conversion for both interface circuits.

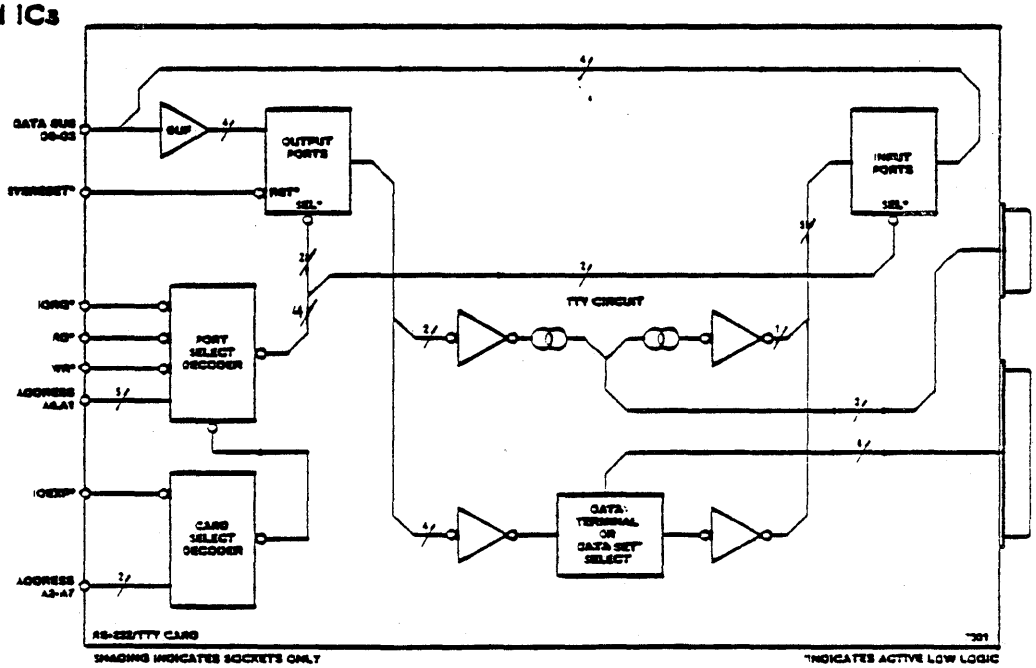
The RS-232-C section of the 7301 provides four line drivers and four line receivers (which are compatible with EIA Standard RS-232-C), and a 25-pin D type cable connector. A pair of DIP sockets on the card enables user-selected Data Terminal or Data Set (modem) configurations by simply inserting a shorting plug (included) into one of the sockets. Operating speed ranges between 0 - 20,000 baud.

The TTY section of the 7301 consists of separate send and receive 20mA current loops, a relay driver for remote control of an ASR-33 type console tape reader, and a 9-pin D type cable connector. Operating speed ranges between 0 - 300 baud.



FEATURES

- **EIA RS-232-C Compatible Interface**
- **ASR/KSR TTY Interface**
- **Simultaneous Full Duplex Operation**
- **On-card Industry Standard Cable Connectors**
- **Program Readable Cable Interlocks**
- **User-selectable Data Terminal/Data Set Configuration**
- **User-selectable Port Addresses**
- **System Processor Generated Baud Rate**
- **Socketed ICs**



PRO-LOG CORPORATION

RS-232 Introduction

EIA Standard RS-232-C (and equivalent international standard CCITT V.24) defines the electrical, mechanical, and procedural aspects of a serial data communication link between two digital systems named the Data Terminal Equipment (DTE, or Data Terminal) and the Data Communication Equipment (DCE, or Data Set).

The Standard specifies thirteen distinct interface configurations and a number of options for each, resulting in a highly flexible data link that is appropriate for a wide variety of applications

For complete familarization with the Standard, the user should obtain the following publications from ELECTRONIC INDUSTRIES ASSOCIATION, Standards Sales Office, 2001 Eye Street N.W., Washington, DC 20006:

a. EIA Standard RS-232-C: Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange

b. Industrial Electronics Bulletin #9: Application Notes for EIA Standard RS-232-C

Microprocessors and RS-232

One classical RS-232 application is a computer terminal (DTE) connected via RS-232 to a modem (DCE) which communicates with a remote computer over telephone lines. A microprocessor can be used in the implementation of any portion of such a system that is within its speed capability.

More frequently, however, a simplified version of RS-232 is used to link a microprocessor with its local peripherals, or to link microprocessors together in a local distributed processing application. The reason for this class of applications is the wide acceptance of RS-232 as a standard interface option for terminals, printers, sensors and actuators, instruments, and potential microprocessor peripherals of all types. The existance of plug-compatible cables and available operating software provide a strong bias in favor of RS-232. Other characteristics which contribute to the popularity of RS-232 are:

a. Many electrical problems avoided by limiting driver/receiver bandwidth to 20,000 bits/second maximum, and by limiting the cable length to 50 feet (15 meters).

b. Electrical compatability is assured by known signal voltages and polarities plus standardized drivers with known loading, short-circuit and power-off characteristics.

c. Standard cable connector pinout and industry-adopted standard D-type connectors assure mechanical compatibility and availability.

d. Within the Standard, RS-232 characteristics are highly adaptable and easy to interface to.

RS-232 Characteristics Summary

The electrical and mechanical standards are explained in detail in RS-232-C, and are summarized in Figures 1 , 2 , and 3 .

RS-232-C defines an interface cable with up to 25 conductors that is up to 50 feet (15 meters) long. The cable's connectors have a fixed pinout, and are used to interconnect a Data Set and a Data Terminal. The 25 possible signals consist of data, control (handshake), and reference (clocks and grounds) signals, not all of which need be used in a given application. Signals not used are ignored at both ends of the cable. The signals that are used are either driven by the Data Terminal and received by the Data Set, or vice versa.

Since the Data Set might be a modem capable of handling telephone line protocol, the Data Set is given precedence over the Data Terminal in some applications, and the majority of available signals are generated by the Data Set. In other applications, the Data Set and Data Terminal are regarded as having equal weight and may operate on a first come-first served basis with many of the potential Data Set signals left unused.

A major distinction between RS-232 systems is whether they are synchronous or asynchronous in nature. In synchronous operation, a timing clock is transmitted along with the serial data. In asynchronous operation, the clock is omitted and data reception and decoding relies on a known, accurate data rate (baud rate) and the inclusion of START and STOP bits in the serial data which allow incoming bits to be framed into characters (Section 6). Most microprocessor peripheral interfaces use the asynchronous mode with ASCII data encoding, and these are the applications the 7301 is optimized for. However, a spare driver/receiver pair (Section 4) can be used to transmit a clock for limited synchronous operation.

Control Signal Handshake

Certain signal protocol must be met before data can be transmitted over the interface. The order of operations is as follows:

a. Data Terminal drives the Data Terminal Ready signal (DTR, pin 20) active to signify that its power is on, it's not in a test mode, and it's ready to participate in communicating with the Data Set.

b. Data Set drives the Data Set Ready signal (DSR, pin 6) active to signify that power is on, it's not in a test mode, and it's ready to communicate with the Data Terminal.

(NOTE: The more complex configurations described by the Standard may require more signalling before the Ready signals are asserted.)

c. Data Terminal now drives the Request To Send (RTS, pin 4) active to condition the Data Set for an imminent transmission of data.

d. Data Set now drives the Clear To Send (CTS, pin 5) active to indicate that conditions have been met to begin transmitting data.

(NOTE: In synchronous systems, clock synchronization occurs before RTS and CTS are asserted.)

e. Serial data can now be transmitted from the Data Terminal over

the Transmitted Data (TD*, pin 2) line or from the Data Set over the Received Data (RD*, pin 3) line. This can be half duplex or full duplex according to the application requirements.

The 7301 provides driver/receiver pairs for the following signal pairs: TD*/RD*, DTR/DSR, RTS/CTS, and Spare/Spare (spares are connected arbitrarily to Carrier Detect and Secondary Transmitted Data). See Section 4 for use and modification information.

Note that the 7301 can be either a Data Terminal or a Data Set. The Signal Configuration Sockets J3 and J4 allow the user to make the selection by inserting a programming plug, which swaps the driven lines and the received lines to the desired configuration.

Potential System Incompatibilities

The user should realize that merely meeting RS-232-C does not guarantee that two systems will be able to communicate as soon as they're interconnected. Besides incompatabilites arising fron the choice of signals, modes, and options outlined in the Standard and the selection of baud rate, the systems may be incompatible due to:

a. Data codes. RS-232-C only specifies binary data. Although ASCII (Section 6) is widely used in English and some other language applications, its use is not mandated in the Standard.

b. Information Protocol. The information contained in the data transmitted (meaning, magnitude, order, etc.) must be intelligible to both systems. For example, both systems must agree on whether the least significant or most significant digit in a large number is transmitted first.

c. Speed compatability. Even if the Data Set and the Data Terminal operate at the same baud rate, the receiving system may not be able to process the data as fast as the transmitting system can send it. The use of the RTS/CTS signal handshake can be used to reduce the transmission rate, as well as the inclusion of extra stop bits. In some cases the receiving system has additional RAM which allows it to accumulate a block of data before processing.

All of these areas of incompatibility are application dependent and require the attention of the system design engineer.

Card Address Mapping

The 7301 is enabled when its jumper-selected address corresponds to the address issued by the processor card during an I/O read or write operation.

The 7301 decodes address lines A1-A7 for card selection, with A0 used to select between the two onboard sequential port addresses. RD* and WR* control the input gating and output latching functions for the ports.

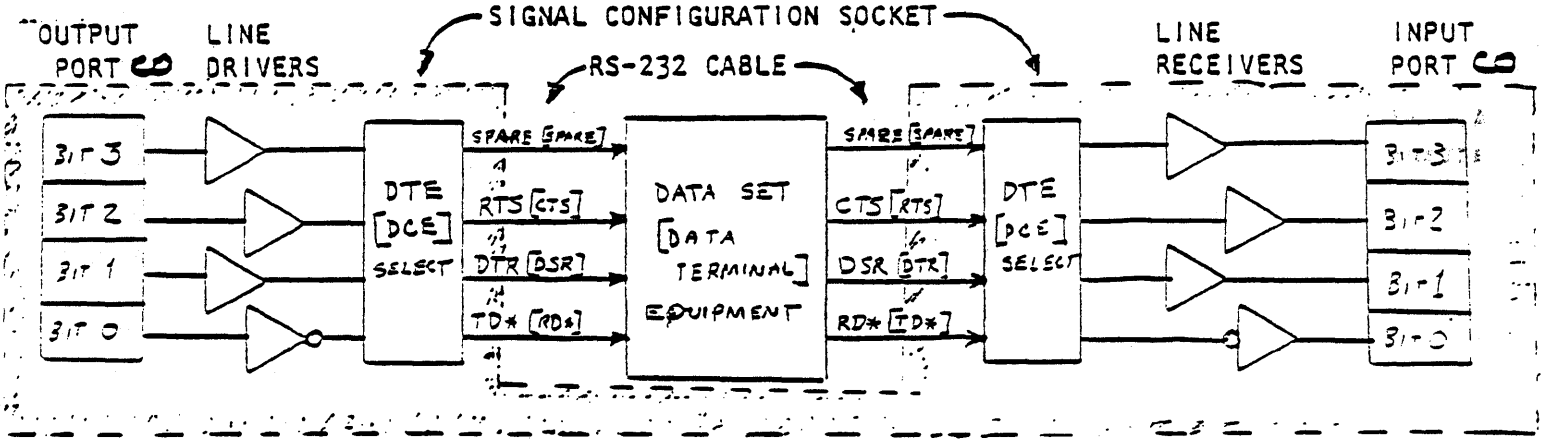
The 7301 is shipped with hexadecimal port addresses C0 and C1 selected by wire jumpers. To change the card address selection see MAPPING, Section 3.

RS-232 Section

The 7301 provides four line drivers and four line receivers that are compatible with EIA Standard RS-232-C. The drivers and receivers are controlled and monitored by one 4-bit output port and one 4-bit input port on the 7301. One additional input port bit provides a program-readable cable interlock which allows the program to determine whether an RS-232 cable is connected to the card. Complete programming information for the RS-232 section of the 7301 is given in RS-232 PROGRAMMING, Section 4.

The 7301 is shipped in a configuration that is suitable for most commercially available computer terminals and modems, Pro-Log's Series 90 PROM Programmers, and for interfacing with Pro-Log's 7304 UART card and UARTs from other manufacturers. The card can occupy either the Data Terminal (DTE) or Data Set (DCE) position in a serial communication system depending on the position of a shorting plug provided with the card. To adapt the 7301 to special applications, refer to RS-232 SIGNAL OPTIONS in Section 4.

Reference information and programming for asychronous ASCII-encoded data applications is found in ASCII CODE INFORMATION, Section 6.



[ ] Denotes user-selectable alternate function
Shaded area represents the 7301's RS232 circuit

Figure 1 - RS-232 Programming Model

(7301 shown in Data Terminal configuration)

20 Milliamp TTY Circuit

The 7301 provides one 20 mA current driver, one 20 mA current receiver, and one open-collector current-sink relay driver for paper tape motion control in modified ASR-33 TTY consoles. The drivers and recievers are controlled and monitored by two output port bits and one input port bit on the 7301. One additional input port bit provides a program-readable cable interlock which allows the program to determine whether a TTY cable is connected to the card. Complete programming information for the TTY section of the 7301 is given in TTY PROGRAMMING, Section 5.

Because of the lack of an industry standard governing 20 mA serial communications, the 7301 is characterized for operation only with model KSR-33 and ASR-33 (or equivalent) electromechanical TTY consoles, or with another 7301. However, a number of strapping options are provided to allow the 7301 to be used reliably in other "20 mA current loop" applications. Refer to TTY SIGNAL OPTIONS in Section 5.

If desired, a user-added modification to the model ASR-33 console allows program control of paper tape motion using the 7301's relay driver. Details of the modification are given in APPENDIX A.

Reference information and programming for ASCII-encoded data applications is found in ASCII CODE INFORMATION, Section 6.

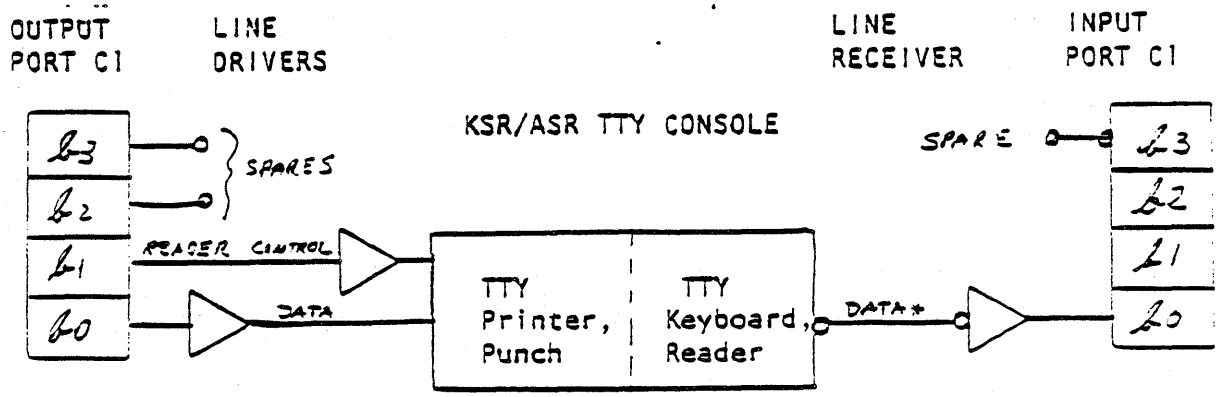


FIGURE 2 : 20 mA TTY PROGRAMMING MODEL

7301 Reset Characteristics

The 7301 is reset by the SYSRESET* signal which is generated by the system processor card at power-on or in response to the PBRESET* input to the processor card.

The SYSRESET* signal has no effect on the 7301's input ports or RS-232 and TTY line receivers.

The SYSRESET* signal clears all of the 7301's output port bits with the following effect on the communications circuits:

a. RS-232 serial data line driver is reset to the Stop Bit (Space) condition; the driver outputs a negative voltage.

b. RS-232 control signal line drivers are reset to the inactive condition; the drivers output negative voltages.

c. TTY transmitter is reset to the Stop Bit (Space) condition such that idle TTY chatter stops; the 20 mA driver is in the ON state.

d. TTY paper tape motion control relay driver is in the OFF state such that the paper tape reader (if present) is not running.

All of the output port bits and corresponding drivers remain in the above states after SYSRESET* until specifically addressed and changed by the program.

Spare TTL I/O Lines

Input Port C1 bit 3 and Output Port C1 bits 2 and 3 are one spare input line and two spare output lines, respectively (Figure 2). Pads are provided for these bits, which are jumper-connected by the user to spare pins at cable connectors J1 and J2 as needed. Refer to the Schematic and Assembly Diagram for the 7301 to determine the location of these signals.

With respect to the system processor's accumulator and the STD Data Bus, the spare input port line is low level active and the spare output port lines are high level active. The spare TTL lines are similar to the TTL lines provided in Pro Log's 7600 Series TTL I/O cards and are suitable for the same applications.

SECTION 3 - MAPPING

I/O Port Address Selection

The 7301 occupies two port addresses with one input port and one output port residing at each address. When shipped, port addresses C0 and C1 hexadecimal are selected by jumper wires. If only one 7301 card is used in the system, or if no existing ports are assigned these addresses, then the address selection is often arbitrary and no change need be made. However, the ports may be assigned any two of four sequential addresses in the range 00-FF hexadecimal as follows:

a. Locate card and port select decoders U3, U4, and U5 (all 74LS42; refer to figure 2 or to the Assembly Diagram 104868).

   Each decoder device has a dual row of pads adjacent to it which forms a decoder output select matrix. Matrix SX is adjacent to U3 and matrix SY is adjacent to U4. Make one and only one connection to each of these matrices. The wire jumper provided when the 7301 is shipped is removed only if a new selection is to be made at that matrix.

   Figure 4 shows where to place jumper straps at SX and SY to obtain any group of four sequential addresses in the range 00-FF. Use the lowest of the group of four addresses desired when determining the jumper positions, and note that this lowest address must end in one of the following hexadecimal digits: 0,4,8, or C (e.g. C0 is the lowest of the group C0, C1, C2, C3).

   Find the most significant hex address digit along the vertical axis of the table in figure 4 , then find the intersection along the horizongal axis that coincides with the desired group of port addresses. At that intersection, read the two connections to be made to SX and SY.

   For example, the card is shipped with jumpers at X6 and Y0, selecting the group C0, C1, C2, C3; the group 28, 29, 2A, 2B is selected with jumpers at X1 and Y2.

b. Next, select two of the four jumpers at matrix SZ adjacent to U5. SZA selects the RS-232 circuit ports, and SZB selects the TTY and cable interlock circuit ports.

   For example, the card is shipped with SZA jumpered to Z0 and SZB jumpered to Z1. This provides the final configuration for the card as shipped, with the RS-232 circuit addressed at hexadecimal address C0 and the TTY/cable interlock circuit at hexadecimal address C1.

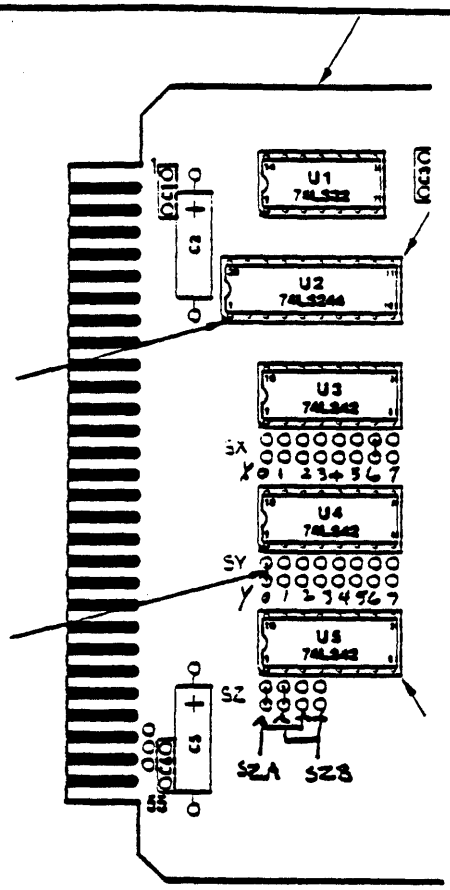


Figure 3 : Location of 7301 Address Decoder Output Select Matrices

MOST SIGNIFICANT HEX ADDRESS	LEAST SIGNIFICANT HEX ADDRESS																JUMPER SELECTION X, Y & Z
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	Z0	Z1	Z2	Z3	Z0	Z1	Z2	Z3	Z0	Z1	Z2	Z3	Z0	Z1	Z2	Z3	← Z
0	X0 Y0				X0 Y1				X0 Y2				X0 Y3				X ANO Y
1	X0 Y4				X0 Y5				X0 Y6				X0 Y7				
2	X1 Y0				X1 Y1				X1 Y2				X1 Y3				
3	X1 Y4				X1 Y5				X1 Y6				X1 Y7				
4	X2 Y0				X2 Y1				X2 Y2				X2 Y3				
5	X2 Y4				X2 Y5				X2 Y6				X2 Y7				
6	X3 Y0				X3 Y1				X3 Y2				X3 Y3				
7	X3 Y4				X3 Y5				X3 Y6				X3 Y7				
8	X4 Y0				X4 Y1				X4 Y2				X4 Y3				
9	X4 Y4				X4 Y5				X4 Y6				X4 Y7				
A	X5 Y0				X5 Y1				X5 Y2				X5 Y3				
B	X5 Y4				X5 Y5				X5 Y6				X5 Y7				
C	X6 Y0				X6 Y1				X6 Y2				X6 Y3				
D	X6 Y4				X6 Y5				X6 Y6				X6 Y7				
E	X7 Y0				X7 Y1				X7 Y2				X7 Y3				
F	X7 Y4				X7 Y5				X7 Y6				X7 Y7				

Figure 4 : 7301 Address Decoder Jumper Selections

SECTION 4 - RS-232 SIGNAL OPTIONS AND PROGRAMMING

General

The 7301 provides four line drivers and four line receivers that are compatible with EIA Standard RS-232-C. These function as voltage translators between the bipolar RS-232 signal levels and the TTL levels required by the STD BUS. The driver/receiver pairs are controlled and monitored by onboard input/output port bits that are in turn controlled by the program.

Since the 7301 does not employ UART devices, the system program is responsible for generating baud rate timing, serial/parallel conversion of data, and handshake control. Flow diagrams and coding forms in Section 9 show how this is done by manipulating the onboard I/O ports.

RS-232 Signal Options

RS-232-C defines a 25-conductor cable with fixed cable connector pinout that links a Data Terminal (DTE) with a Data Set (DCE). The Data Terminal drives the set of signal lines that are received by the Data Set; the Data Set drives the set of signal lines that are received by the Data Terminal. The 7301's two Signal Configuration Sockets (Figure 1) allow the user to select either the Data Terminal or Data Set configuration for the card by inserting a shorting plug into one of the sockets. The shorting plug swaps all four line drivers and four line receivers simultaneously, readying it for communication with equipment with the complimentary configuration.

In the asynchronous RS-232 applications for which the 7301 is intended, the equipment at either end of the link drives one serial data line and up to two asynchronous handshake lines, as shown in Figure 5 . A spare driver/receiver pair is provided for applications requiring another control or clock signal.

SIGNAL MNEMONIC	SIGNAL NAME	CABLE PIN	DRIVEN BY	RECEIVED BY
TD*	Transmitted Data	2	DTE	DCE
RD*	Received Data	3	DCE	DTE
DTR	DTE Ready	20	DTE	DCE
DSR	DCE Ready	6	DCE	DTE
RTS	Request To Send	4	DTE	DCE
CTS	Clear To Send	5	DCE	DTE
(Spare)		14	DTE	DCE
(Spare)		8	DCE	DTE

FIGURE 5 : 7301 DRIVER/RECEIVER PIN ASSIGNMENTS

As shown in Figure 5 , the Data Terminal drives serial data to the Data Set via the TD* line and uses DTR, RTS, and (Spare) optionally to handshake with the Data Set. The Data Set drives serial data to the Data Terminal via the RD* line and uses DSR, CTS, and (Spare) optionally to handshake with the Data Terminal. If the Data Set wishes to inhibit the Data Terminal, for example, it might do so by dropping CTS to the inactive state.

Note that not all RS-232 compatible equipment drive all of these lines. For example, a CRT terminal might produce the DTR signal by simply routing the DSR signal back to the Data Set by a wire jumper. Or, it might produce DTR by connecting the line to a positive power supply greater than +3 Volts so that DTR is active anytime the terminal is turned on. In either case the Data Set could not inhibit the terminal's operation by dropping its DSR signal since DSR is probably not monitored by the terminal in this instance.

The 7301 provides a matrix of signal option pads near the RS-232 connector (Figure 29) where the user can disconnect three of the four driver/receiver pairs and reconnect them with wire jumpers as needed in specific applications. The positive and negative power supply potentials are available at this matrix so that permanent enable/disable signal levels can be connected to the RS-232 cable in addition to the driver/receiver pairs. Note that reconnections made by the user will be swapped if the Signal Configuration plug is swapped. Refer to the 7301 Schematic and Assembly diagrams for details of this option.

RS-232 Programming

Figures 7 and 8 show programming models of the RS-232 drivers and receivers, respectively. The four output port bits directly control the states of the line drivers, and the states of the line receivers can be read directly from the input port.

Note that the RS-232 serial data lines are low level active (inverted) while the handshake control and spare drivers are high level active. The 7301 provides inverters on both the TD* and RD* lines so that the serial data is high level active in the microprocessor's accumulator.

Figure 6 shows the bit functions of the line driver/receiver control ports according to whether the Data Terminal or Data Set signal configuration is selected.

DATA TERMINAL CONFIGURATION								
PORT CO	BIT FUNCTION							
	7	6	5	4	3	2	1	0
INPUT (RECEIVERS)	1	1	1	1	DSR	CD	CTS	RD*
OUTPUT (DRIVERS)	X	X	X	X	DTR	SCD	RTS	TD*

DATA SET CONFIGURATION								
PORT CO	BIT FUNCTION							
	7	6	5	4	3	2	1	0
INPUT (RECEIVERS)	1	1	1	1	DTR	SCD	RTS	TD*
OUTPUT (DRIVERS)	X	X	X	X	DSR	CD	CTS	RD*

X = Don't Care

FIGURE 6 : PORT CO BIT FUNCTIONS

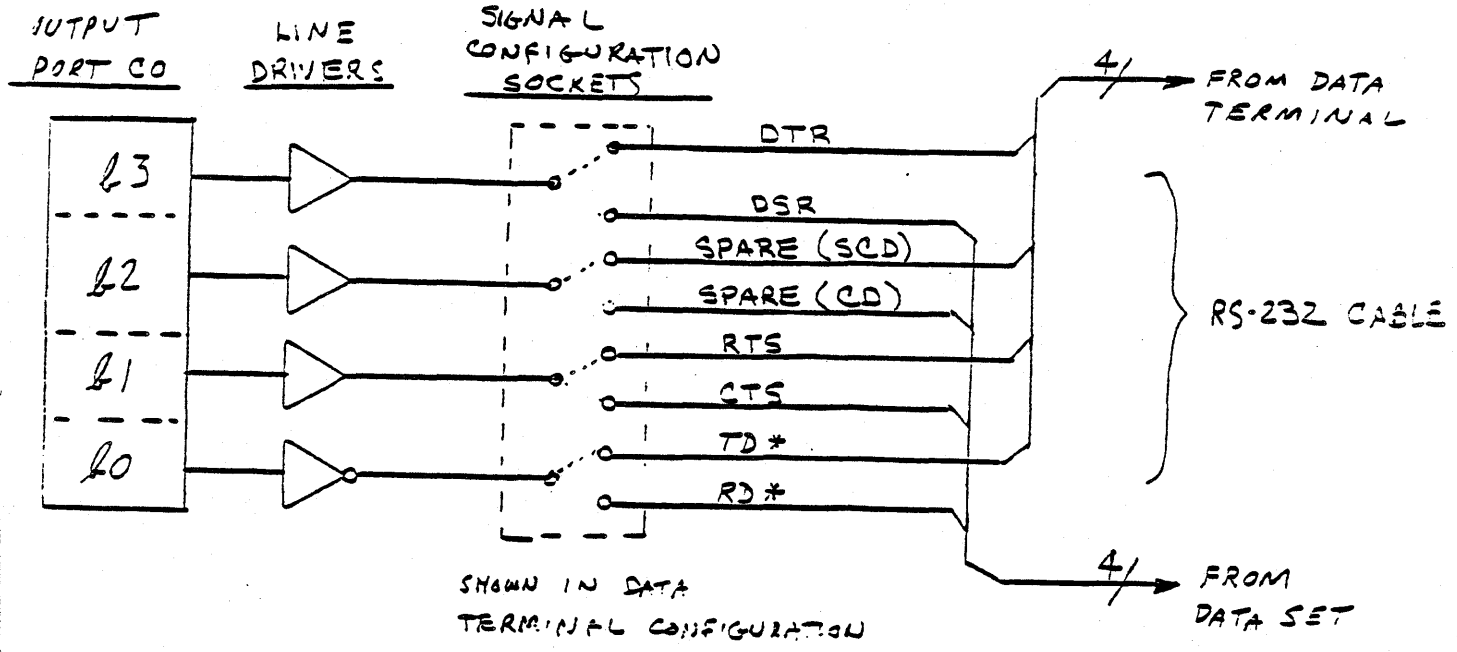


FIGURE 7 : RS-232 DRIVER PROGRAMMING MODEL

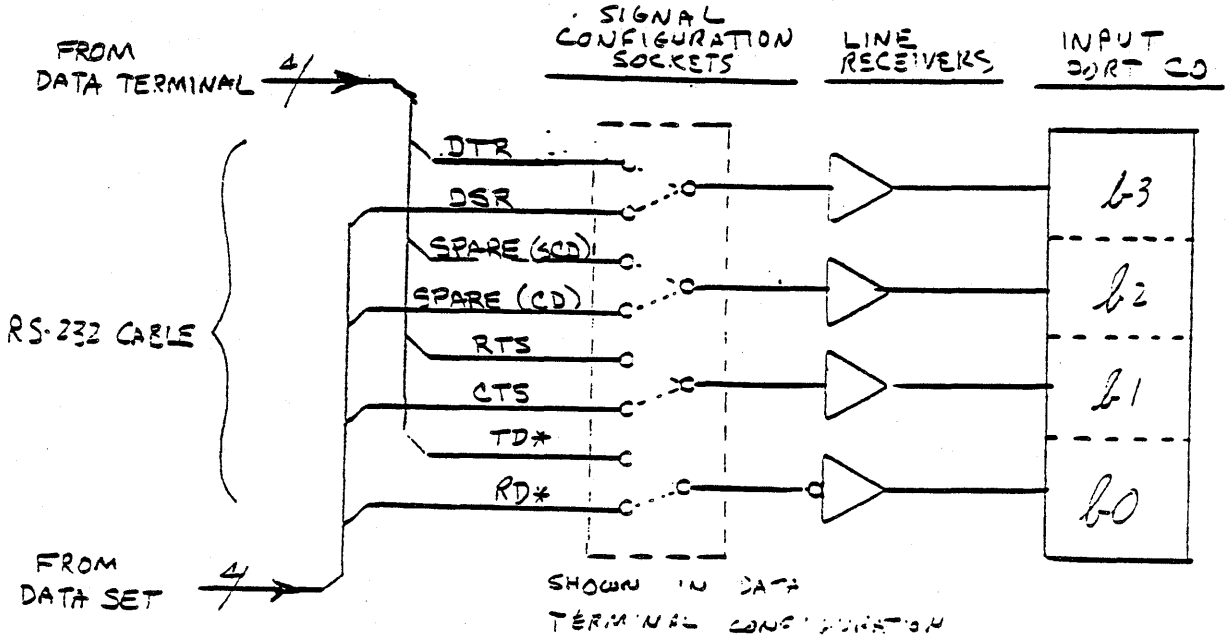


FIGURE 8 : RS-232 RECEIVER PROGRAMMING MODEL

RS-232 Cable Interlock

Input Port C1, used for TTY Receive, provides bit 2 as an RS-232 cable interlock. The purpose of this interlock is to allow the program to determine whether an RS-232 cable, active or inactive, is plugged into the card. This feature can be used in system error-checking or as a convenient technique to determine what system options are present (if the RS-232 option is not in use, the program need not expend time in checking for incoming data).

To use this feature, the user's RS-232 cable assembly should provide a connection between SIGNAL GROUND, pin 7 of J2, and CABLE INTERLOCK*, pin 25 of J2. When the cable is connected, Input Port C1 bit 2 will be in the 1 state when read; if the cable is missing, bit 2 will read 0. (Figure 9).

If J2 pin 25 is in use in the application, unassigned pins 11 and 18 can be used for the cable interlock function. Jumper pads are provided for this purpose in the vicinity of J2 (see Figure 29). Input port C0 bit 3 is available for this purpose.

SECTION 5 - TTY PROGRAMMING AND CONFIGURATION OPTIONS

General

The 7301 provides nominal 20 mA current loop circuitry designed to interface with electromechanical KSR-33 and ASR-33 (or equivalent) consoles, and to Pro Log's Series 90 PROM Programmers. Figure 11 illustrates the intended interface, and Figure 27 shows the recommended wiring for the TTY cable assembly used with the 7301.

The TTY interface circuitry can be altered by the user at pads provided on the 7301 to interface with a variety of other "TTY compatible" instrumentation and communications interfaces. Figures 12 and 13 illustrate possible variations in the interface circuit.

TTY Programming

Figure 9 shows the bit function of the TTY I/O programming port C1. Input Port C1 bit 0 reads the 0 mA/20 mA (mark/space) state of the TTY receiver, which is intended to be driven by a mechanical commutating switch contact. Output Port C1 bit 0 controls the state of the 20 mA driver.

TTY CURRENT LOOP								
PORT C1	BIT FUNCTION							
	7	6	5	4	3	2	1	0
INPUT	1	1	1	1	SPARE*	RS-232 INTLK*	TTY INTLK*	DATA IN
OUTPUT	X	X	X	X	SPARE	SPARE	TAPE CTRL	DATA OUT

X = Don't Care

*Low Level Active

FIGURE 9 : PORT C1 BIT FUNCTIONS

Figure 10 shows the logical meaning of I/O Port C1 bit 0:

	PROGRAMMED STATE	Nominal Current	State Name	ASCII True Data State	Line Voltage	
INPUT PORT C1	bit 0 = 0	0mA	Mark (START)	0	High	7301 RECEIVE
	bit 0 = 1	20mA	Space (STOP)	1	Low	
OUTPUT PORT C1	bit 0 = 0	20mA	Space (STOP)	1	High	7301 TRANSMIT
	bit 0 = 1	0mA	Mark (START)	0	Low	

FIGURE 10 : PORT C1 BIT 0 LOGICAL MEANINGS

A specific programming example with flow diagram and 8080/8085/Z80 coding forms are shown in Section 9. See Section 6 for ASCII code information.

CASE 1. 7301 STANDARD CONFIGURATION

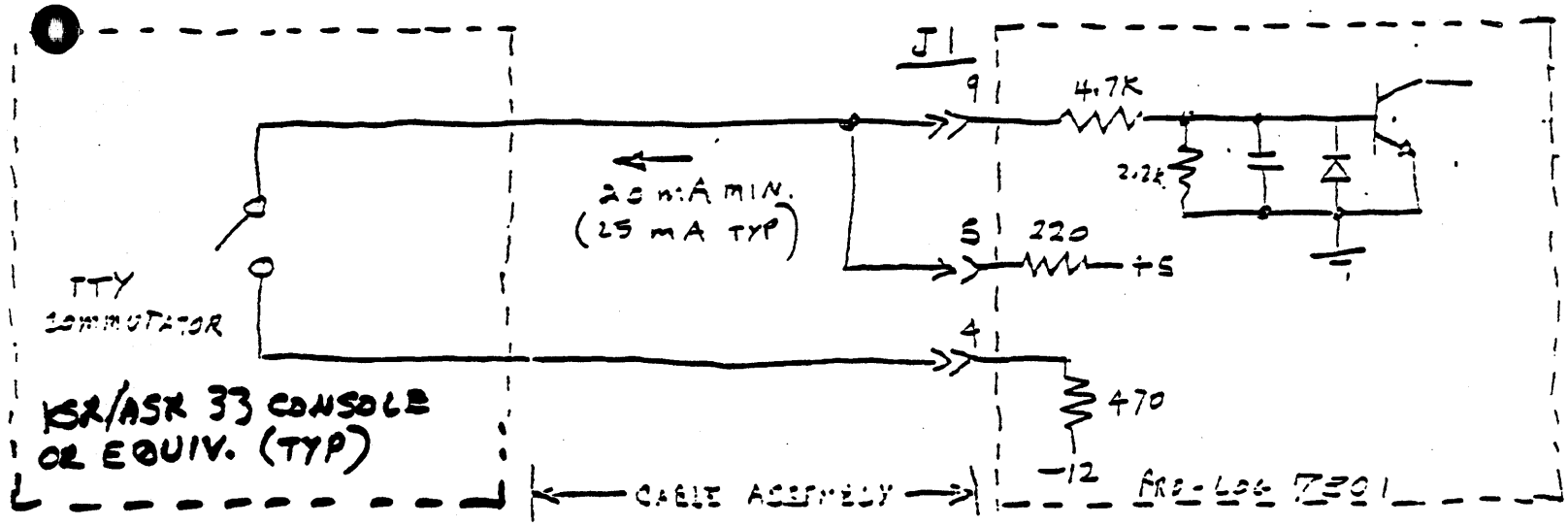


FIGURE 11 A: 7301 CURRENT LOOP RECEIVE
STANDARD TTY CONSOLE

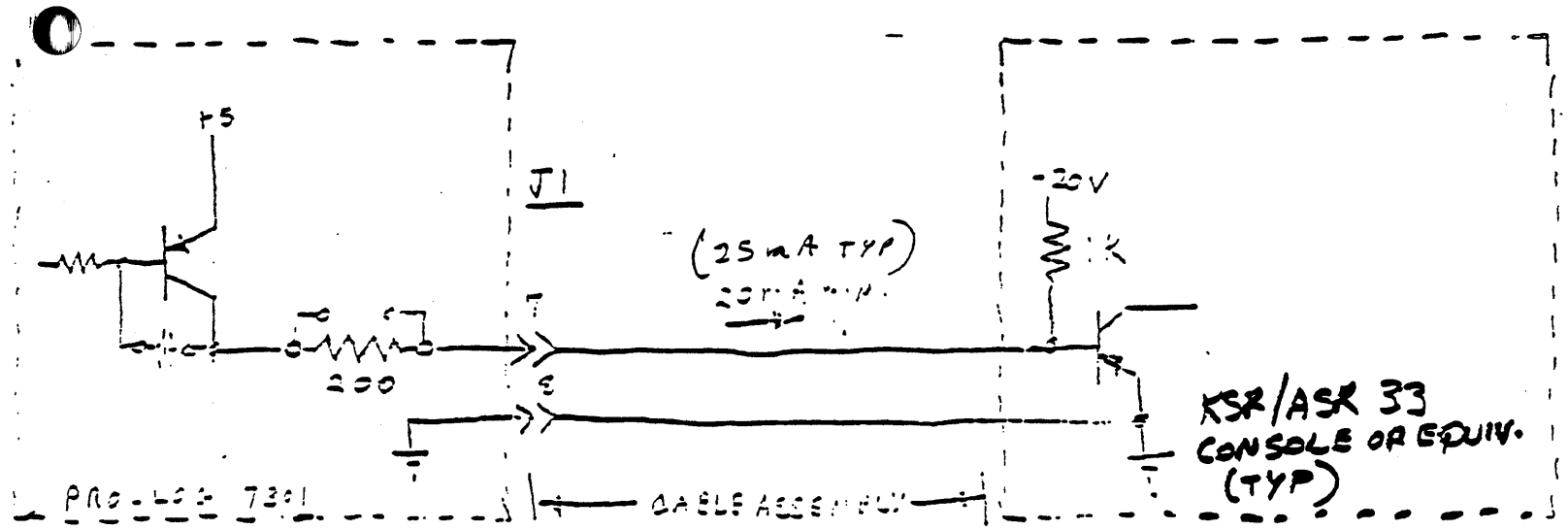


FIGURE 11 B: 7301 CURRENT LOOP TRANSMIT
STANDARD TTY CONSOLE
WITH 20 mA OPTION ONLY

TELETYPE INTERFACE

CASE 2: SUGGESTED USER MODIFICATION FOR ELECTRONIC TERMINAL

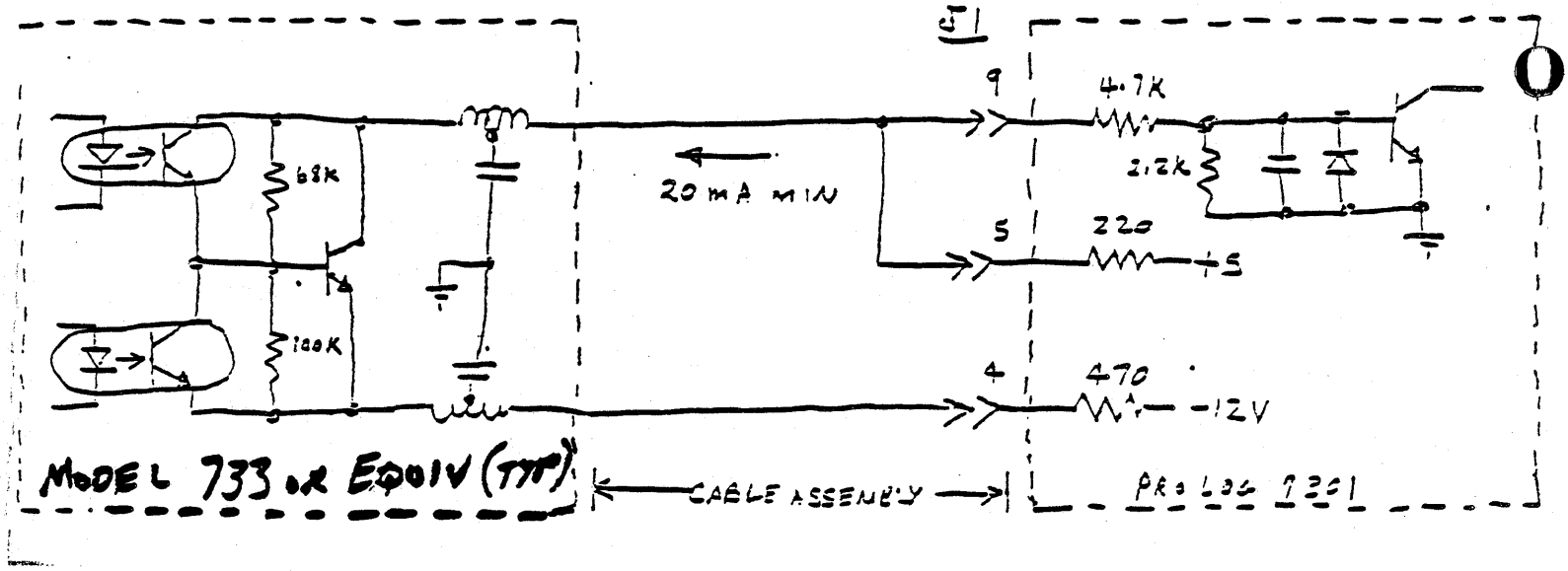


FIGURE 12A: 7301 CURRENT LOOP RECEIVE

TYPICAL ELECTRONIC DATA TERMINAL
(WITH TTY/COMPUTER INTERFACE OPTION)

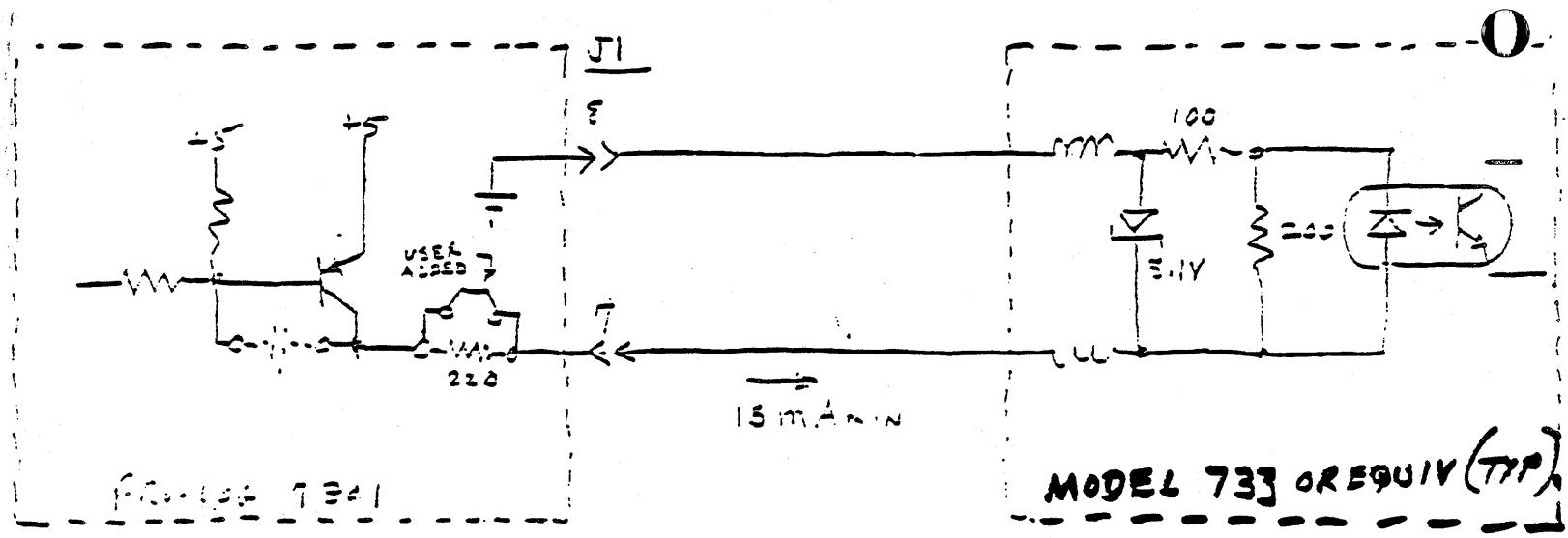


FIGURE 12B 7301 CURRENT LOOP TRANSMIT

ELECTRONIC DATA TERMINAL
(WITH TTY/COMPUTER INTERFACE OPTION)

PAGE 3: SUGGESTED USER MODIFICATION FOR INSTRUMENT

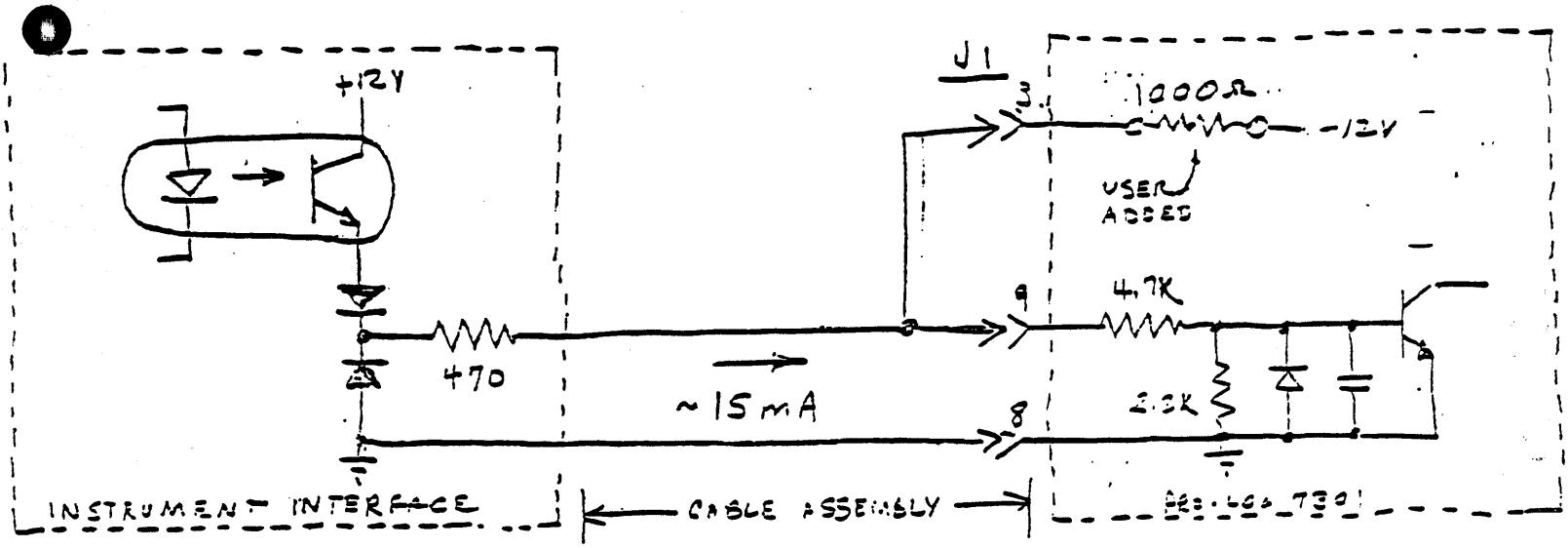


FIGURE 13A: 7301 CURRENT LOOP RECEIVE:
USER-ADDED PULLDOWN RESISTOR
AT J1 PIN 2 (SPARE)

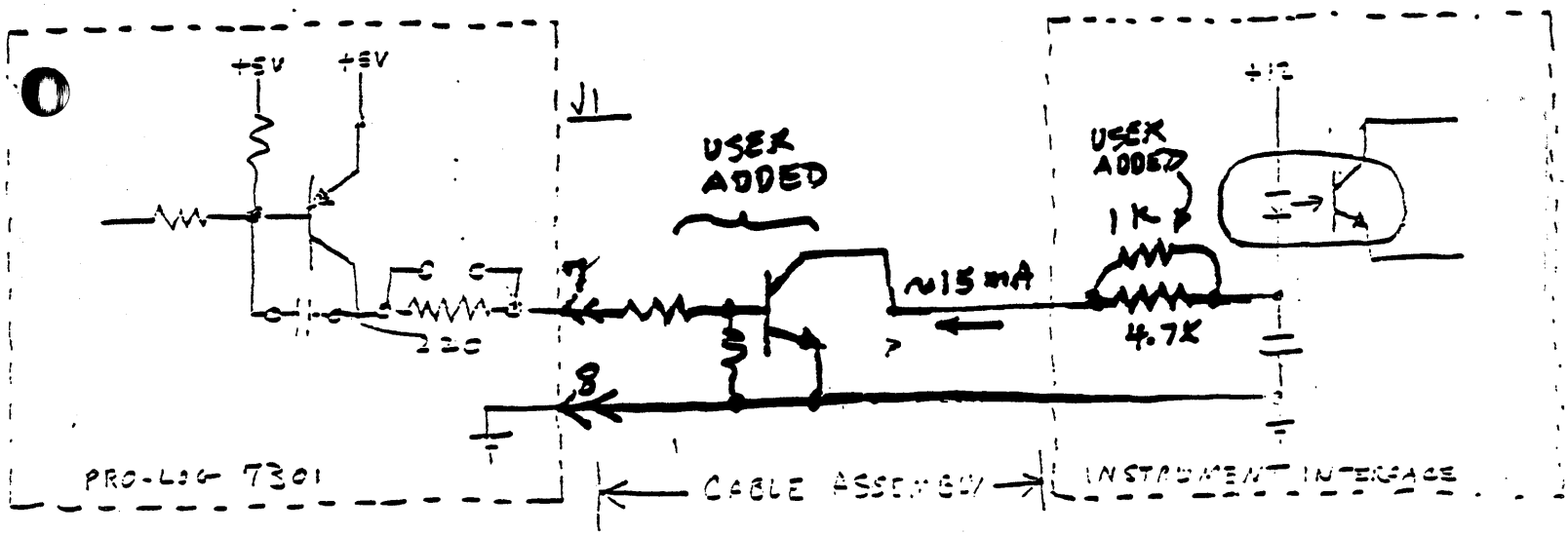


FIGURE 13B: 7301 CURRENT LOOP [illegible]

USER MODIFIED INSTRUMENT INTERFACE

NOTE: SELECT RESISTOR APPROPRIATE TO INPUT LED'S MAY LIMIT INTERFACE TO LAB APPLICATION AND SHORT CABLE)

TTY Cable Interlock

Input Port C1 provides bit 1 as a TTY cable interlock status bit. The purpose of this interlock is to allow the program to determine whether a TTY cable, active or inactive, is connected to the card. This feature can be used in system error-checking, or as a convenient technique to determine what system options are present (if the TTY option is not in use, the program need not expend time in checking for incoming data).

To use this feature, the user's TTY cable assembly should provide a connection between LOGIC GROUND (J2 pin 8) and CABLE INTERLOCK* (J2 pin 2). When the cable is connected, Input Port C1 bit 1 will be in the 1 state when read; if the cable is missing, bit 1 will read 0.

TTY Tape Reader Control

Appendix A describes a modification to the ASR-33 console which allows a user-added relay to stop and start the paper tape reader automatically via the processor's program. The 7301 provides a switchable current source and -12VDC relay supply which power the relay and allow it to be switched according to the state of Output Port C1 bit 1 as shown in Figure 14. Figure 15 shows a typical hardware configuration.

	PROGRAMMED STATE	NOMINAL CURRENT	RELAY	TAPE MOTION
OUTPUT PORT C1	BIT 1 = 0	0mA	Open	Stopped
	BIT 1 = 1	60mA	Closed	Moving

FIGURE 14 : ASR-33 PAPER TAPE MOTION CONTROL

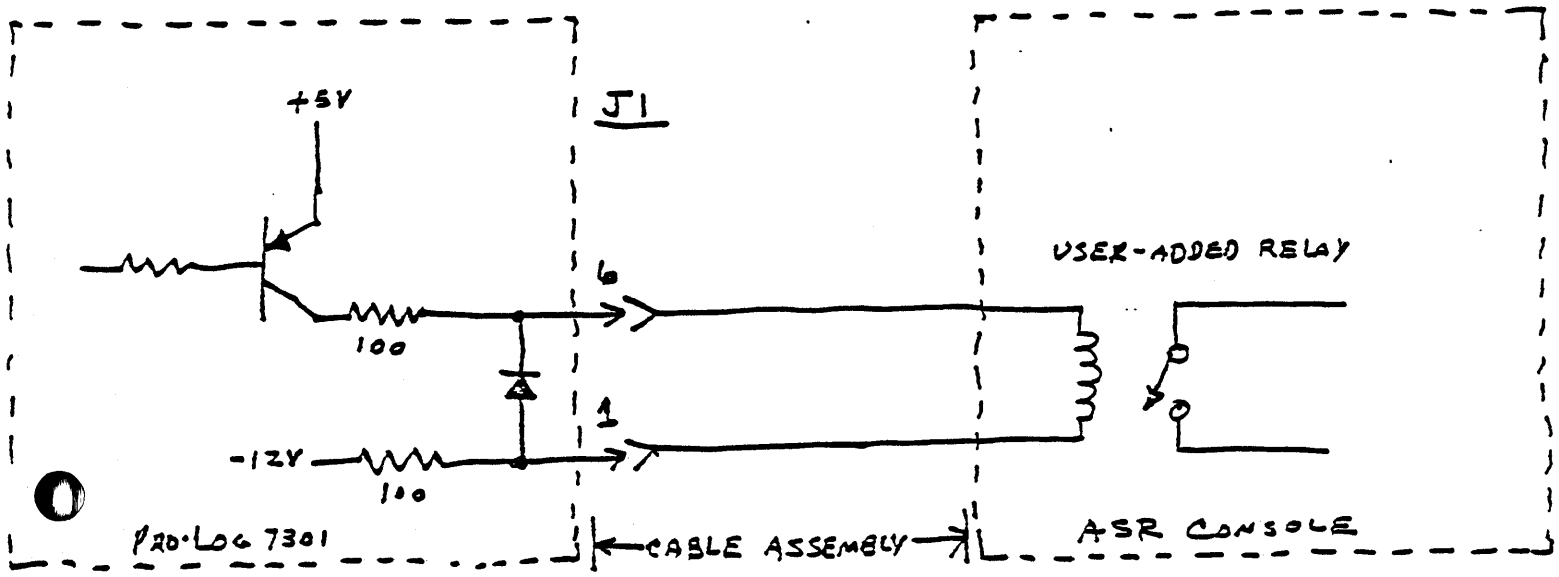


FIGURE 15: RECOMMENDED PAPER TAPE MOTION CONTROL RELAY USER MODIFICATION FOR ASR TTY CONSOLES

RELAY CHARACTERISTICS: CONTACTS – 15VA
COIL – 600Ω MIN (12V-15V)

SUGGESTED SOURCES: CORNELL-DUBILIER # 603-12V
GUARDIAN # 1345-1C-12D
SIGMA # 65F1A-12DC

SECTION 6 - ASCII CODE INFORMATION

ASCII (American Standard Code for Information Interchange) is an 8-bit code for data transmission. The code consists of 7 bits for information, allowing a 128-character set to be encoded, and one bit for error-checking (parity).

8-bit microprocessors can conveniently handle the ASCII characters as one parallel byte of information. In serial data applications such as RS-232, one START bit and one or more STOP bits are added to the ASCII code for asychronous operation. These bits, in conjunction with an accurate serial bit timing rate (baud rate) and the convention that the least significant bit is transmitted first, allow the receiving system to frame the received bit stream into ASCII characters and to decode them accurately.

Format

Figure 16 shows the serialized ASCII character's voltage waveform as it would be seen on the TD* or RD* lines (RS-232 pins 2 or 3), consisting of:

a. High level active START bit which begins the character and triggers the receive operation;

b. Seven low level active data bits, beginning with the least significant bit (b0) and progressing to the most significant bit (b6); (b7)

c. Low level active parity bit/which, if used, causes the data field (b0 through b7) to have an even number of active bits ("even parity") or an odd number of active bits ("odd parity");

d. Low level active STOP bit, which lasts one or more bit times and which represents the idle condition when no data transmission is taking place.

Note that in some applications the following variations may be encountered:

a. Data field may be reduced to 5 or 6 bits, reducing the size of the character set but increasing the character transmission rate;

b. Parity bit may be held in one state or omitted;

c. 1, 1.5, 2 or more STOP bits may be needed, and special timing for certain mechanical printer operations such as carriage return, line feed, and top-of-form may be implemented as extra STOP bits.

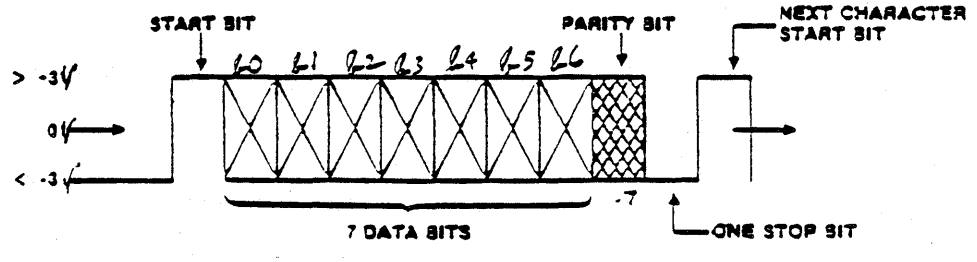


Figure 16 : RS-232/ASCII Serial Character Format

Inverters on the 7301 allow the programmer to use positive true logic. ASCII data in internal processor registers is operated on and communicated to the ports as true data. Internally the START bit is a logic 0 and the STOP bit is a logic 1, and the ASCII character appears as the 2-digit hex value shown in Figure 17.

ASCII Character Set

Figure 17 shows the 2-digit hexadecimal number required to produce a binary bit pattern equivalent to the ASCII code for the standard TTY keyboard character set. Locate the character desired, then find the least significant hexadecimal digit of the code along the left axis and the most significant digit across the top of the table.

Parity adds or subtracts hexadecimal 80 from the code, depending on the odd, even, suppressed, set, or clear state of the parity bit required by the user's system.

Control Characters are the codes produced when the keyboard's CONTROL (CTRL) key is depressed while one of the alphabet or [, \,], ^, _, or @ keys is depressed. The standard meaning of the control characters is shown below the code table. Their advantage in keyboard control applications is that two simultaneous keystrokes are required to produce them, so operator errors are reduced. Note that a control character can be produced arithmetically by subtracting hexadecimal 40 from the equivalent alphabet or special character.

Some equipment, particularly intelligent terminals, monitor and react to certain control characters while generating them. Consult the equipment user's manual for special control character considerations.

Lower Case alphabet characters are frequently omitted in low cost terminals and printers, with the SHIFT key action limited to the subset of punctuation and special characters. Lower case characters are generated arithmetically by adding hexadecimal 20 to the equivalent upper case character.

HEX				MSD	p = 1	8	9	A	B	C	D	E	F
					p = 0	0	1	2	3	4	5	6	7
	BITS				b7	p	p	p	p	p	p	p	p
					b6	0	0	0	0	1	1	1	1
					b5	0	0	1	1	0	0	1	1
LSD	b3	b2	b1	b4 / b0		0	1	0	1	0	1	0	1
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p	
1	0	0	0	1	SOH	DC1	!	1	A	Q	a	q	
2	0	0	1	0	STX	DC2	"	2	B	R	b	r	
3	0	0	1	1	ETX	DC3	#	3	C	S	c	s	
4	0	1	0	0	EOT	DC4	$	4	D	T	d	t	
5	0	1	0	1	ENQ	NAK	%	5	E	U	e	u	
6	0	1	1	0	ACK	SYN	&	6	F	V	f	v	
7	0	1	1	1	BEL	ETB	'	7	G	W	g	w	
8	1	0	0	0	BS	CAN	(	8	H	X	h	x	
9	1	0	0	1	HT	EM	)	9	I	Y	i	y	
A	1	0	1	0	LF	SUB	*	:	J	Z	j	z	
B	1	0	1	1	VT	ESC	+	;	K	[	k	{	
C	1	1	0	0	FF	FS	,	<	L	\	l	|	
D	1	1	0	1	CR	GS	-	=	M	]	m	}	
E	1	1	1	0	SO	RS	.	>	N	^	n	~	
F	1	1	1	1	SI	US	/	?	O	_	o	DEL	

CONTROL CHARACTERS

NUL	Null	FF	Form Feed	CAN	Cancel
SOH	Start of Heading	CR	Carriage Return	EM	End of Medium
STX	Start of Text	SO	Shift Out	SUB	Substitute
ETX	End of Text	SI	Shift In	ESC	Escape
EOT	End of Transmission	DLE	Data Link Escape	FS	File Separator
ENQ	Enquiry	DC1	Device Control 1	GS	Group Separator
ACK	Acknowledge	DC2	Device Control 2	RS	Record Separator
BEL	Bell (audible or attention signal)	DC3	Device Control 3	US	Unit Separator
BS	Backspace	DC4	Device Control 4 (Stop)	DEL	Delete (Rubout)
HT	Horizontal Tabulation (punched card skip)	NAK	Negative Acknowledge		
LF	Line Feed	SYN	Synchronous Idle		
VT	Vertical Tabulation	ETB	End of Transmission Block		

FIGURE 17 : ASCII Character Code Table

Baud Rate Timing

The baud rate is the rate at which bits are transmitted over the serial data line, expressed as bits per second.

The character rate is the number of characters transmitted per second, or the baud rate divided by the number of bits per character. For example, standard serial ASCII characters are 10 bits long, so 110 baud should produce eleven characters per second. However, electromechanical KSR/ASR-33 consoles require 2 stop bits for 11 bits per character, so their character rate at 110 baud is 110/11 = 10 characters per second.

Figure 18 shows the standard baud rates, bit times, and standard ASCII character rates for most of the RS-232 equipment used in the United States.

BAUD RATE	TIME FOR ONE BIT	CHARACTERS PER SECOND (ASCII w/one stop bit)
50	20.00 ms	5
75	13.33 ms	7.5
110	9.09 ms	11
134.5	7.44 ms	13.5
150	6.66 ms	15
200	5.00 ms	20
300	3.33 ms	30
600	1.66 ms	60
1200	833.3 us	120
1800	555.5 us	180
2400	416.6 us	240
4800	208.3 us	480
9600	104.2 us	960
19200	52.08 us	1920

FIGURE 18 : BAUD RATE TIMING TABLE

ASCII Programming Considerations

ASCII transmit and receive operations via RS-232 require the following:

a. Parallel-to-serial conversion by the processor for transmit, and serial-to-parallel conversion for receive; and

b. Accurate bit timing within the processor's program.

Parallel/serial conversion is generally performed by a combination of shift or rotate commands which isolate or accumulate individual bits, and the OR, AND, and Exclusive OR logic commands which allow individual bits to be masked, set, cleared, and complemented. The routines in Section 9 illustrate these techniques, along with ASCII-to-hexadecimal conversion and related operations.

Timing depends on the program's ability to generate a constant time delay based on the time required to manipulate the data (usually a small fraction of the bit time) plus a programmed time delay which adds to the processing time for each bit to produce the bit time required by the baud rate. Accuracy of one programmed bit time is important because the one-bit error is cumulative for all the bits in the character. For example, a 1% error in the width of the START bit would be a 9% error at the Parity bit if the same time delay routine is used for all nine bits (the START bit, seven data bits, and parity bit).

The routines provided in Section 9 employ a technique where the basic timing unit is 1/2 the bit time required for the baud rate. This causes the processor to read the state of the incoming data line in the nominal center of the received data bits (Figure 19), allowing up to a 50% error to accumulate by the time the Parity bit is received. The maximum allowable error in bit timing is then 50% / 9 = 5.5%, or 2.7 % for the 1/2-bit time delay. While the crystal-controlled microprocessor can achieve much more precise delays for low and moderate baud rates (Table 1), the accumulated error for an electromechanical terminal may be as high as 20%.

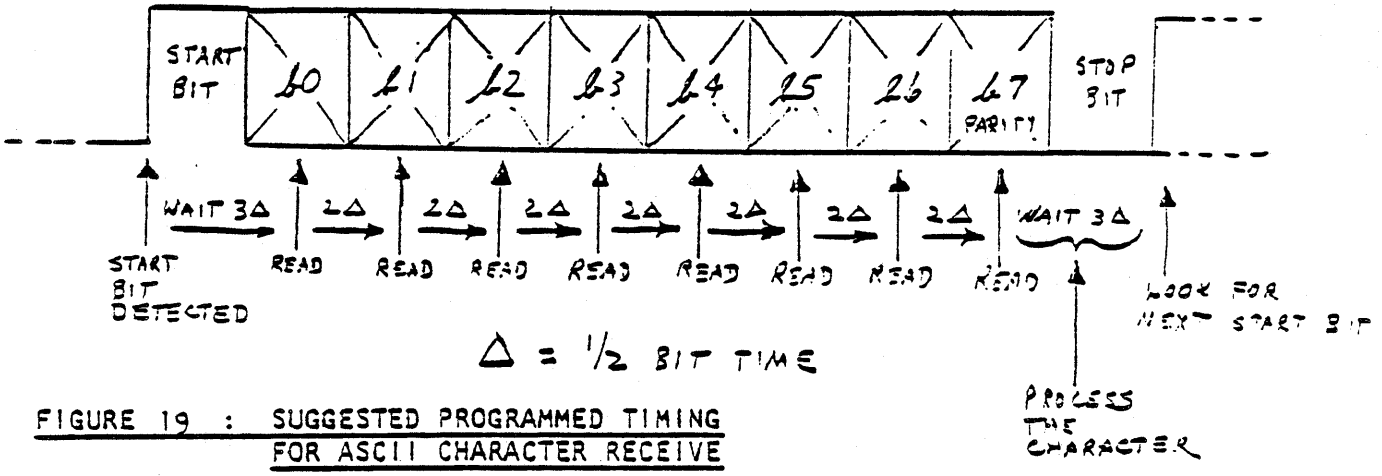


FIGURE 19 : SUGGESTED PROGRAMMED TIMING FOR ASCII CHARACTER RECEIVE

Note that system noise immunity can be improved by rereading the START bit 1/2 bit time after its initial detection. If the START bit is not still present at that time, the receive operation was initiated by a noise spike and can be terminated without introducing a system error.

SECTION 7 - 7301 ELECTRICAL SPECIFICATIONS

ELECTRICAL

- VCC = +5V, ±5%
- ICC = 525mA maximum (380mA typical)
- AUX -V = -12V ±5%
- AUX I = -200mA maximum
- Address, data, and control buses meet all STD BUS general electrical specifications.

* See Figure 28 in Section 8 for STD BUS loading

RS-232 Electrical Specifications

	PARAMETER	MIN	MAX	UNIT	NOTE
VOH	High Level Output Voltage	+3		V	1
VOL	Low Level Output Voltage	-3		V	
IOS	Output Short Circuit Current, Output Shorted To ±12V (Duration 1 Second Maximum)		85	mA	
CL	Output Load Capacitance		2500	pF	2
Tr Tf	Output Rate Of Change		30	V/μs	
VIH	Input High Level Voltage	+3	+25	V	3
VIL	Input Low Level Voltage	-25	-3	V	
IIH	Input High Level Current	+0.5	+8.5	mA	
IIL	Input Low Level Current	-8.5	-0.5	mA	
EL	Output Termination Bias	-2.0	+2.0	V	
	BAUD Rate (Bits/Second)	0	20,000	BAUD	
	Recommended Cable Length		50 16	feet meters	
VO	Driver Open Circuit Output Voltage	-12	+5	V	
RO	Driver Output Resistance, Power Off	300		ohm	
RL	Input Load Resistance	3K	6K	ohm	3

NOTES:
1. Minimum load resistance 3K ohms
2. Includes cable and terminator capacitance
3. Except Cable Interlock, which is 4 LSTTL input loads

FIGURE 20 : 7301 RS-232 Connector J2

Typical RS-232 Line Voltages

Figure 21 shows the VIH/VIL range of the 7301's RS-232 receivers to the left of the center axis, and typical VOH/VOL performance with other than the specified 3K load for the 7301's RS-232 drivers on the right.

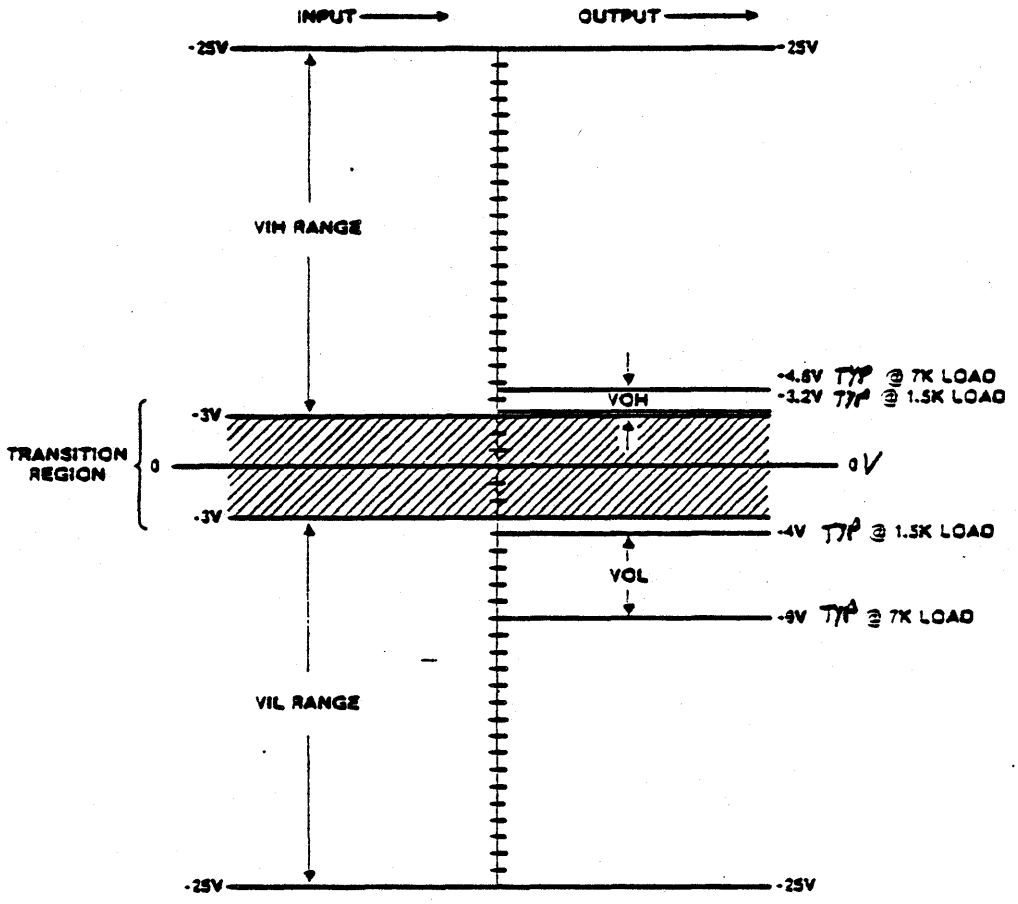
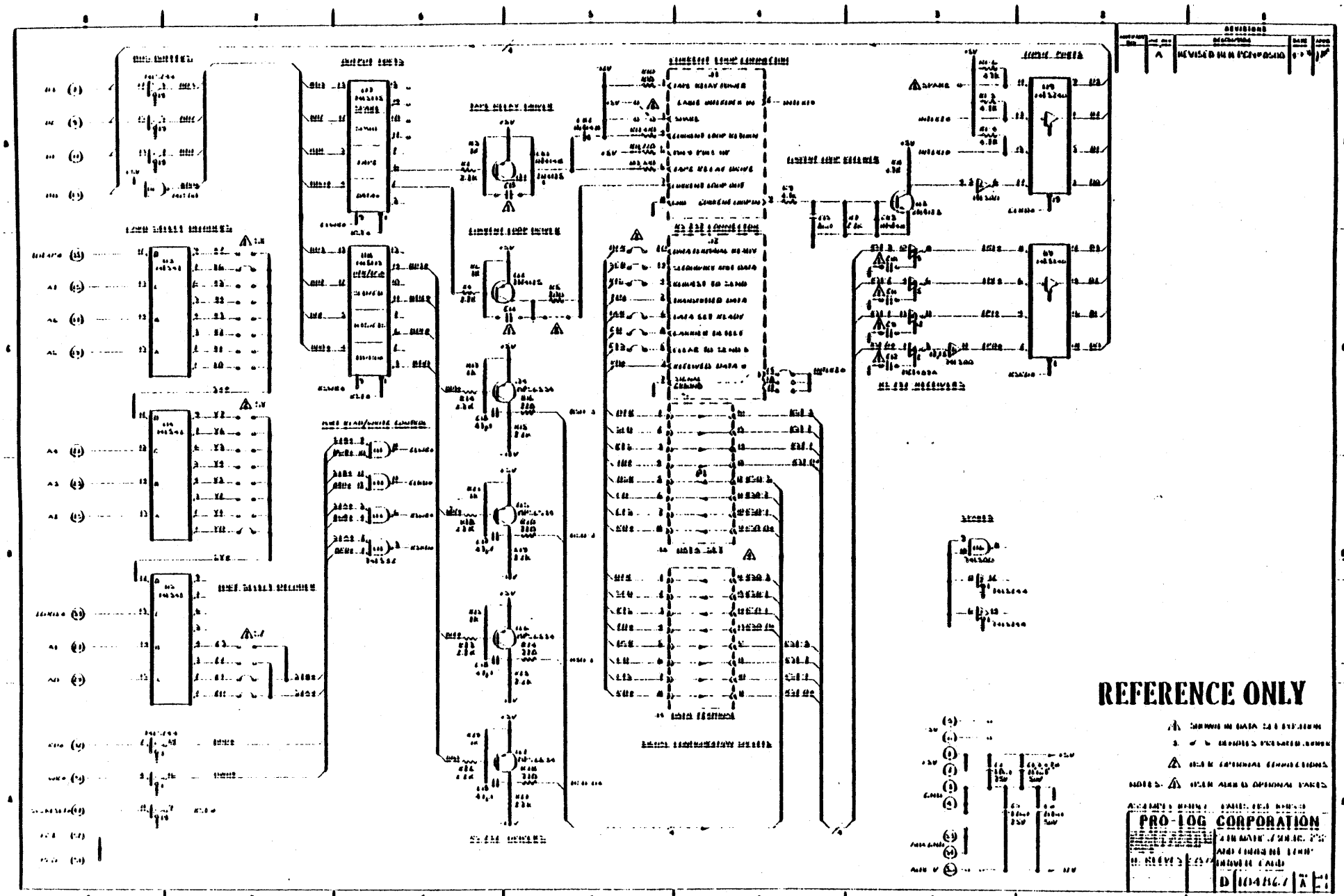


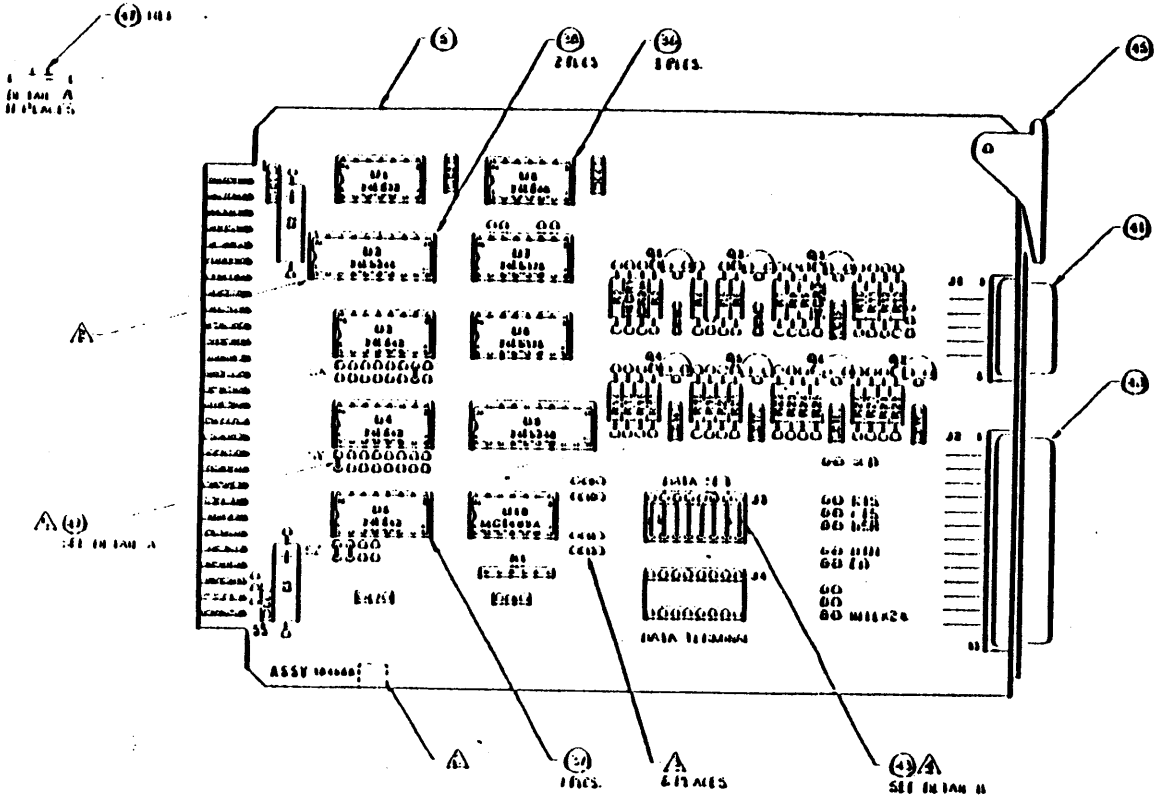
FIGURE 21 : Typical 7301 RS-232 Driver/Receiver Performance

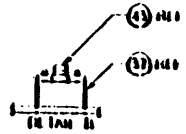
TTY Electrical Soecifications

J1 PIN	FUNCTION
1	Paper tape motion control relay coil power source; provides -12V through a 100 ohm, 1/4 Watt, 5% resistor (diode clamp to pin 6)
2	Interface cable interlock input; active low logic; use is optional. Designed to be connected to logic ground (pin 8) by the interface cable when the cable is plugged into 7301 J1. Read into the processor as input port C1 bit 1; a logic 1 is returned when pin 2 is grounded, or a logic 0 when open. Input loading 4 LSTTL loads.
3	Spare pin; connected to a pad on the 7301 adjacent to pads supplying +5V or -12V. User may connect a jumper wire or a 1/2 Watt resistor to either voltage for special applications.
4	Current loop return (7301 receive); supplies -12V through a 470 ohm, 1/2 Watt, 5% resistor supplying 20 mA return and pin 9 voltage pulldown when the TTY commutator is closed (spacing). Not required in some instrument interfaces.
5	20 mA source (7301 receive); supplies +20 mA from +5V through a 220 ohm, 1/4 Watt, 5% resistor when the TTY commutator is closed (spacing) and pulls up pin 9 when the commutator is open (marking).
6	Paper tape control relay drive; supplies up to +50 mA through a switching transistor and 100 ohm, 1/4 Watt, 5% resistor when output port C1 bit 1 is at logic 1 (+5V current source). When output port C1 bit 1 is at logic 0 or the 7301 is reset, output leakage current is 0.1 mA maximum (pin 6 at 0V).
7	20 mA source (7301 transmit); provides +20 mA minimum (measured with pin 7 at 0V) from the +5V supply through a switching transistor and a 200Ω, 1/4 Watt, 5% resistor when output port C1 bit 0 is at logic 0. Output leakage current (pin 7 at 0V) with output port C1 bit 0 at logic 1 is 0.1 mA maximum. Note that this output supplies +20 mA when the 7301 is reset. Pads are provided for jumpering across the 200Ω resistor in instrument interface applications requiring additional drive voltage.
8	Logic ground out; connects to LOGIC GROUND (STD BUS pin 3, 4).
9	Current Loop Receive (7301 receive); a transistor base input through a 4.7K ohm, 1/4 Watt, 5% resistor and negative current clamp diode. A voltage greater than +2.5V applied to pin 9 is returned to the processor as input port C1 bit 0 at logic 1. A voltage lower than +0.8V is read as logic 0.

FIGURE 22: **J1 Current Loop Connector Specifications**


REFERENCE ONLY
PRO-LOG CORPORATION





REFERENCE ONLY

PRO-LOG CORPORATION

SECTION 8 - 7301 MECHANICAL SPECIFICATIONS

The 7301 meets all STD BUS general mechanical specifications with the exception of the TTY connector J1 and the RS-232 connector J2 which protrude from the card front 0.375 inches (0.953 cm). With interface cable connected, two card slot widths may be required.

Cable Connector Pinouts

Figures 23 and 24 show the RS-232-C Standard pinout for 25-pin cable connectors, and for the 9-pin TTY connector, respectively. Figure 25 gives vendor model numbers for J1 and J2 mating connectors.

RS-232 CONNECTOR PIN LIST					
PIN NUMBER	SIGNAL ORIGIN		PIN NUMBER	SIGNAL ORIGIN	
SIGNAL					SIGNAL
PROTECTIVE GROUND		1	14	T□	SEC. TRANSMITTED DATA*
TRANSMITTED DATA*	T■	2	15	S	TRANS SIGNAL TIMING
RECEIVED DATA*	S■	3	16	S	SEC. RECEIVED DATA*
REQUEST TO SEND	T□	4	17	S	RECV. SIGNAL TIMING
CLEAR TO SEND	S□	5	18		(UNASSIGNED)
DATA SET READY	S□	6	19	T	SEC. REQUEST TO SEND
SIGNAL GROUND	OUT	7	20	T□	DATA TERMINAL READY
CARRIER DETECT	S□	8	21	S	SIGNAL QUALITY DETECT
(TEST)		9	22	S	RING INDICATOR
(TEST)		10	23	T/S	DATA SIGNAL RATE SEL
(UNASSIGNED)		11	24	T	TRANS. SIGNAL TIMING
SEC. CARRIER DETECT	S	12	25	INC	CABLE INTERLOCK*
SEC. CLEAR TO SEND	S	13			

*Designates Active Low Level Logic

LEGEND:
T - Signal Originates At Data Terminal (DTE)
S - Signal Originates At Data Set (DCE)
■ - Permanent Connection On 7301
□ - Jumper Connection On 7301; May Be Relocated By User

FIGURE 23 : **RS-232 Connector Pin List (J2)**

TTY CONNECTOR PIN LIST					
PIN NUMBER	SIGNAL FLOW		PIN NUMBER	SIGNAL FLOW	
SIGNAL					SIGNAL
-12V/100Ω SOURCE	OUT	1	6	OUT	SWITCHED RELAY SINK*
CABLE INTERLOCK*	IN	2	7	OUT	CURRENT LOOP TRANS*
(SPARE)		3	8	OUT	LOGIC GROUND
-12V/470Ω SOURCE	OUT	4	9	IN	CURRENT LOOP RECV*
+5V/220Ω SOURCE	OUT	5			

*Designates Active Low Level Logic

FIGURE 24 : **TTY Connector Pin List (J1)**

MANUFACTURER	CONNECTOR MODEL	
	RS-232-C (25-PIN D)	TTY (9-PIN D)
AMPHENOL	17-90250-16	17-90090-16
ITT - CANNON	DBM-25P	DEM-9P
TRW - CINCH	DBM-25P	DEM-9P

FIGURE 25: **Mating Connector Information**

Figures 26 and 27 show the end-on pin numbering used by the RS-232 connector J2 and the TTY connector J1, respectively, as seen from the card front.

The recommended wiring diagram shown for the TTY connector J1 results in KSR/ASR-33 console compatibility, and compatability with Pro Log's Series 90 PROM Programmers with 9102 and 9112 TTY options.

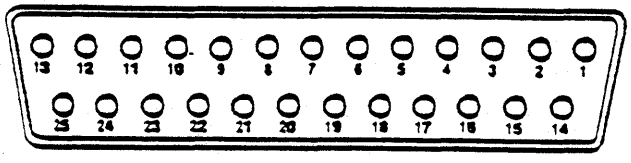


FIGURE: 26 **RS-232 Connector Pins; Female (7301 J2 Front View)**

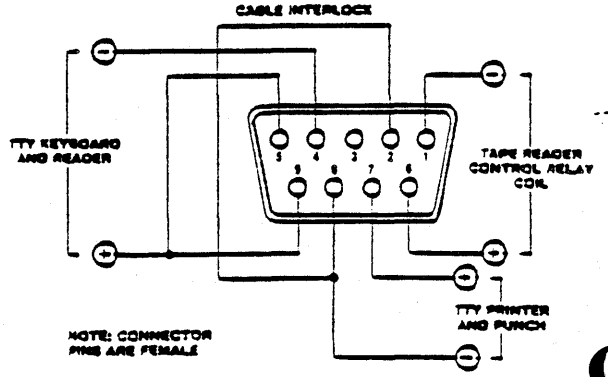


FIGURE:27 **Recommended TTY Hookup (J1 Front View)**

7301 STD BUS EDGE CONNECTOR

STD/7301 EDGE CONNECTOR PIN LIST							
PIN NUMBER				PIN NUMBER			
OUTPUT (DRIVE) Ⓐ						OUTPUT (DRIVE) Ⓑ	
INPUT (LOADING) Ⓐ						INPUT (LOADING) Ⓑ	
MNEMONIC							MNEMONIC
-5 VOLTS	IN		2	1		IN	-5 VOLTS
GROUND	IN		4	3		IN	GROUND
-5V			6	5			-5V
D7			8	7	55	1	D3
D6			10	9	65	1	D2
D5			12	11	56	1	D1
D4			14	13	66	1	D0
A15			16	15		1	A7
A14			18	17		1	A6
A13			20	19		1	A5
A12			22	21		1	A4
A11			24	23		1	A3
A10			26	25		1	A2
A9			28	27		1	A1
A8			30	29		1	A0
RD*	1		32	31		1	WR*
MEMRQ*			34	33		1	IORQ*
MEMEX*			36	35		1	IOEXP*
MCSYNC*			38	37			REFRESH*
STATUS 0*			40	39			STATUS 1*
BUSRQ*			42	41			BUSAK*
INTRQ*			44	43		1	INTAK*
NMIRQ*			46	45			WAITRQ*
PBRESET*			48	47		1	SYSRESET*
CNTRL*			50	49			CLOCK*
PC1	IN		52	51	OUT		PC0
AUX GND	IN		54	53	IN		AUX GND
AUX-V (+12V)	IN		56	55			AUX -V

*Designates Active Low Level Logic
Ⓐ DESIGNATES LSTTL LOADS

FIGURE 28: Connector Pin List

Strapping Option Locations

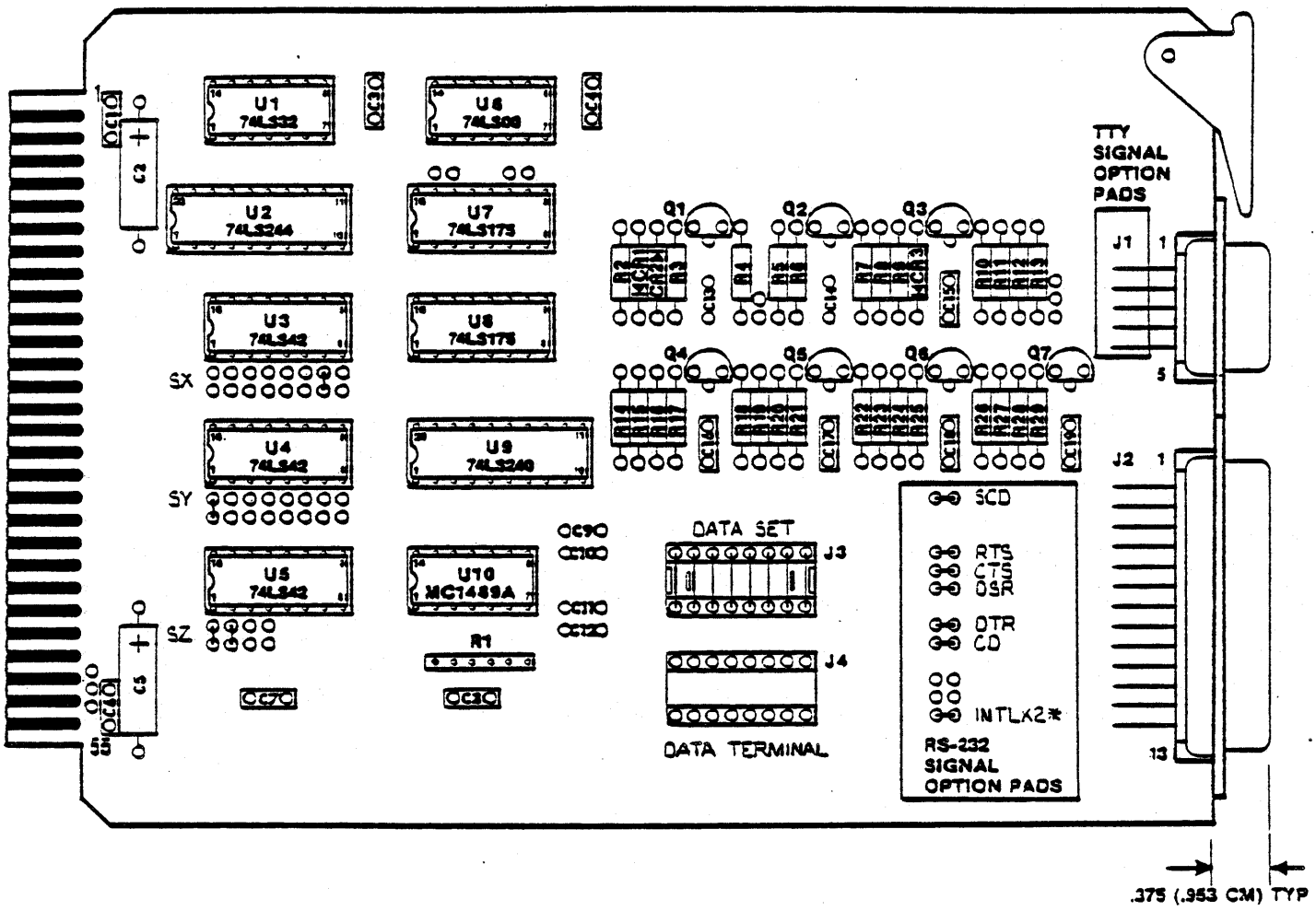


FIGURE 29 : 7301 USER OPTION LOCATIONS

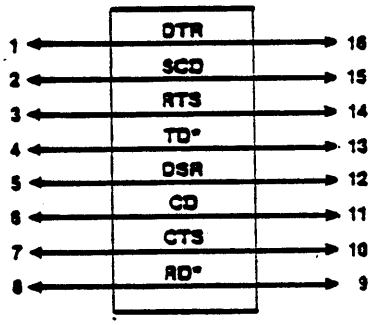
⊶⊷ Designates as-shipped jumper location

NOTE: Signal Configuration Select Plug P1 shown in Data Set (DCE) select position (J3).

Signal Configuration Select Plug (RS-232 Circuit)

Signal Configuration Select Plug P1 is used to establish the 7301 card's position as Data Terminal or Data Set in an RS-232 communication link as shown in Section 2. .

The plug's shorting links are exposed to allow for test points. The signals shown in Figure 30 are independent of the plug's position in the Data Terminal Select Socket J4 or the Data Set Select Socket J3, though the source of the signals changes with the plug's position. If the user elects to change the driven and received lines by changing jumpers at the RS-232 signal option pads, the test points at P1 reflect these changes.



Data Terminal/Data Set Select Plug P1 Wiring
Shorting Bars Exposed for Scope Test Points

FIGURE 30

SECTION 9 - 7301 OPERATING FIRMWARE

The following routines are provided for the 7301 user. They will operate in either Z80 or 8085 systems; the flow diagrams which describe them allow easy conversion to another processor's assembly language, although timing constants generally require modification according to variations in instruction execution time between processors and in crystal oscillator frequency.

These routines may be used without licensing from Pro Log, or they may be used as models for modification. Although they have been tested and are believed to be correct, they are not represented to be appropriate to any specific application or free from errors or copyright infringement.

The subroutine modules are devided into two groups. One group is dedicated to effecting a full RS-232-C interface, and uses input and output port C0 of the 7301 card. The other group is dedicated to a 29mA current loop interface and uses input and output port C1. The user can implement an RS-232-C interface with no control lines, by changing the current loop routine port address to C0. All RS-232-C routines are independent of the 7301 signal configuration, i.e. they may be used if the 7301 is set up for DATA SET or DATA TERMINAL.

Upon entry, all transmit routines require that the data to be transmitted be in the accumulator or A register. Upon exit from all receive routines, the received data is in the accumulator or A register. All registers used by the routines are saved and unaltered upon completion of the routines.

All routines use a common delay subroutine called ($\frac{1}{2}$ BITΔT). This counting loop retrieves a 16 bit delay count, N, from two adjacent memory locations, then down-counts N to zero. Those memory locations, designated in the listing by the label

DELAY COUNT, can be two RAM locations or two ROM locations. If they are in RAM, the BAUD rate can be varied under program control. If in ROM, the BAUD rate will remain fixed. If in RAM, these locations must be loaded with N prior to the first access of the serial routines.

All listings include the object code of all instructions in Hex. All address dependent locations are blank and can be filled in by the user, permitting assignment of the routines to any area of memory desired. The Comment section of the Program Assembly Forms should be read carefully prior to using these routines. Certain locations may be modified to further allow the user to adapt the routines to specific needs.

The DELAY COUNT, N, can be obtained from Table 1. provided herein. The table is arranged to provide a count, N, for the BAUD rate, the microprocessor, and at the clock rates provided on PRO-LOG processor cards.

MODULE DESCRIPTIONS

(RCV & ECHO) This routine receives one character each time it is accessed, and echoes the character as it is received. CTS/RTS is raised high prior to reading the start bit and dropped low during the reception of the stop bit. FOR RS-232-C USE ONLY.

(SEND) This routine transmits one character each time it is accessed. RTS/CTS is raised during transmission of the start bit and dropped during transmission of the stop bit. If the 7301 is used as a DATA TERMINAL, the ANAI instruction following the label FIND DTR must be modified as shown. FOR RS-232-C USE ONLY.

(AUTOBAUD) This routine, upon receiving an ASCII CR (carriage return), measures the width of the start bit and determines the BAUD rate. A count, N, is generated which is subsequently stored in the two adjacent RAM memory locations specified by the label DELAY COUNT. The routine must precede any transfer of data, since it automatically adjusts the system BAUD rate to that of the sending device. FOR RS-232-C USE ONLY.

(PTRD & PRNT) This routine reads and echoes one character from the paper tape reader. FOR USE WITH 20mA CURRENT LOOP ONLY.

(RD & PRNT) This routine reads and echoes one character from a 20mA current loop interface, or from an RS-232-C interface that uses only the transmit and receive data lines (TD, RD).

(PTRD ONLY) This routine reads one character from the paper tape reader without echoing it. FOR USE WITH 20mA CURRENT LOOP ONLY.

(RD ONLY) This routine reads one character from a 20mA current loop interface, or from an RS-232-C interface that uses only the transmit and receive data lines (TD, RD). Data is not echoed.

(BAUD RATE) This routine, upon receiving an ASCII CR (carriage return), measures the width of the start bit and determines the BAUD rate. A count, N, is generated which is subsequently stored in the two adjacent RAM memory locations specified by the label DELAY COUNT. The routine must preceded any transfer of data, since it automatically adjusts the system BAUD rate to that of the sending device. FOR USE WITH 20mA CURRENT LOOP ROUTINES, OR RS-232-C INTERFACES THAT USE ONLY THE RECEIVE AND TRANSMIT DATA LINE.

(RD-PT) This routine activates the paper tape reader for the transmission of one character only, then deactivates it.

(2 BITΔT) This routine implements an approximate 2 bit delay at the current BAUD rate.

(BIT Δ T) This routine implements an approximate 1 bit delay at the current BAUD rate.

(½ BIT Δ T) This routine implements an approximate ½ bit delay at the current BAUD rate. The current BAUD rate is dependent upon the 16 bit count, N, stored in the two adjacent memory locations identified as DELAY COUNT. N can be found from Table 1., or set by the (BAUD CHK) or (BAUD RATE) routines.

(DELAY) This routine implements a time delay dependent upon some count N, determined and preset by the user prior to entry. The delay can be found by the relationship:

$$T = aNt,$$

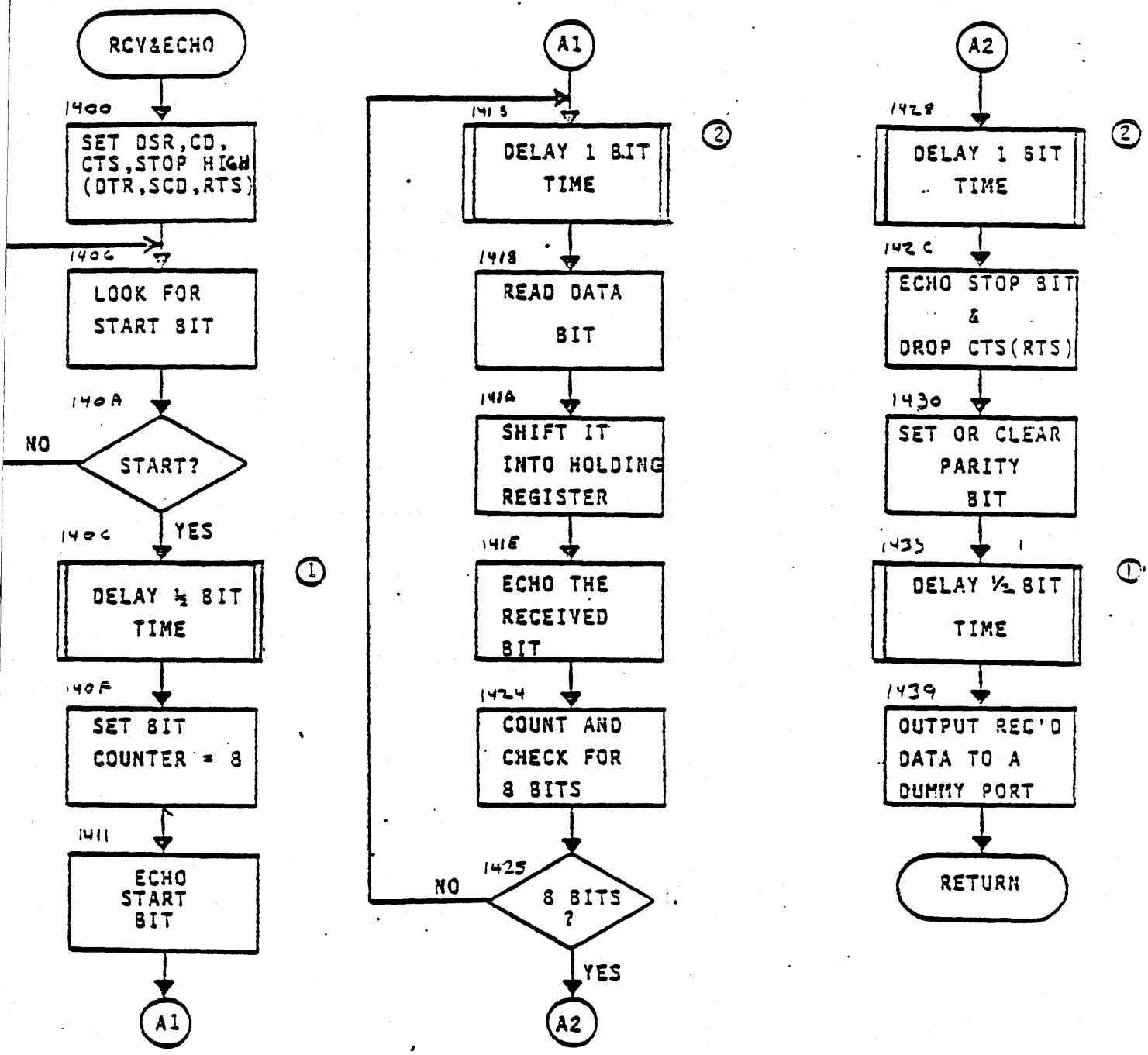
and the values of a and t can be found at the bottom of Table 1.

BAUD RATE	PERIOD µs	HEXADECIMAL TIME DELAY CONSTANT							
		Z-80		Z-80A		8085A		8085A	
		0.4µs	Error	0.25µs	Error	0.32µs	Error	0.20µs	Error
50	20000.00	040E	.05%	067F	.03%	0513	.002%	0820	.01%
75	13333.33	02B3	.04%	0454	.007%	0361	.001%	056A	.013%
110	9090.91	01D6	.06%	02F2	.05%	024D	.03%	03B0	.001%
134.5	7434.94	0180	.01%	0268	.06%	01E1	.003%	0303	.05%
150	6666.67	0158	.01%	0228	.07%	01AF	.008%	02B3	.06%
200	5000.00	0101	.09%	019D	.12%	0142	.14%	0206	.04%
300	3333.33	00AA	.24%	0113	.15%	00D6	.02%	0158	.05%
600	1666.67	0054	.43%	0088	.21%	0069	.41%	00AB	.26%
1200	833.33	0028	.49%	0042	.37%	0033	.35%	0054	.30%
1800	555.56	001A	.86%	002B	.24%	0021	.30%	0037	.33%
2000	500.00	0017	.55%	0026	1.16%	001D	1.53%	0031	.04%
2400	416.67	0013	2.19%	0022	1.32%	0018	.24%	0028	.79%
3600	277.78	000C	4.74%	0014	.17%	000F	.12%	001A	.43%
4800	208.33	0008	3.01%	000E	1.01%	000B	3.54%	0013	1.64%
7200	138.89	0004	.64%	0008	3.46%	0006	.22%	000B	2.88%
9600	104.17	0002	4.58%	0006	5.52%	0004	3.98%	0008	1.91%
19200	52.08	0001	35.22% ①	0001	3.65%	0001	16.53% ①	0002	7.16% ①
TIME STATES		48N + 153				46N + 147			
TIME (µs)		19.2N+61.2		12N+38.25		15.36N+47.04		9.6N+29.4	

HEXIDECIMAL TIME DELAY CONSTANTS FOR RS232 AND CURRENT LOOP SUBROUTINES
(For use with 7301 card)

TABLE 1.

NOTE ①: NOT RECOMMENDED DUE TO EXCESSIVE ERROR



RS232 RECEIVE ROUTINE

PRO-LOG CORPORATION 8085A/Z80 PROGRAM ASSEMBLY FORM

TITLE Z301 RS232 ROUTINES DATE 3-6-80

14	0	0	E5	(RCV-ECHO)	PSP	HL
		1	D5		PSP	DE
		2	3E		LDAI	
		3	0F		-	0F
		4	D3		OPA	
		5	C0		-	C0
	0	6	DB	START	IPA	
		7	C0		-	C0
		8	1F		RRAC	
		9	DA		JP	C1
		A	06		-	START
		B	14		-	
		C	CD		JS	
		D	E6		-	(½BITAT)
		E	15		-	
		F	16		LDDI	
14	1	0	08		-	COUNT 8
		1	3E		LDAI	
		2	0E		-	0E
		3	D3		OPA	
		4	C0		-	C0
	1	5	CD	RD:ECHO	JS	
		6	E3		-	(BITAT)
		7	15		-	
		8	DB		IPA	
		9	C0		-	C0
		A	1F		RRAC	
		B	7B		LDA	E
		C	1F		RRAC	
		D	5F		LDE	A
		E	17		RLAC	
		F	3E		LDAI	
14	2	0	07		-	07
		1	17		RLAC	
		2	D3		OPA	
		3	C0		-	C0
		4	15		JDCD	
		5	C2		JP	20
		6	15		-	RD:ECHO
		7	14		-	
		8	CD		JS	
		9	E3		-	(BITAT)
		A	15		-	
		B	3E		LDAI	
		C	0D		-	0D
		D	D3		OPA	
		E	C0		-	C0
		F	7B		LDA	E
14	3	0	F6		ORAI	
		1	80		-	PARITY
		2	5F		LDE	A
		3	CD		JS	
		4	E6		-	(½BITAT)
		5	15		-	
		6	7B		LDA	E
		7	D1		PLP	DE
		8	E1		PLP	HL
		9	D3		OPA	
		A	XX		-	XX
		B	C9		RTS	
		C				
		D				
		E				
		F				

((RECEIVE & ECHO 1 CHARACTER. UPON EXIT, CHARACTER IS IN REGISTER A))

SET DSR, CD, CTS ACTIVE WITH STOP BIT. (OR DTR, SCD, RTS, STOP)

LOOK FOR START BIT

DELAY ½ BIT TIME

SET BIT COUNTER = 8

ECHO START BIT

DELAY 1 BIT TIME

READ DATA BIT

SHIFT IT INTO REGISTER E

ECHO DATA BIT

COUNT AND CHECK FOR 8 BITS

ECHO STOP BIT, DROP CTS (RTS)

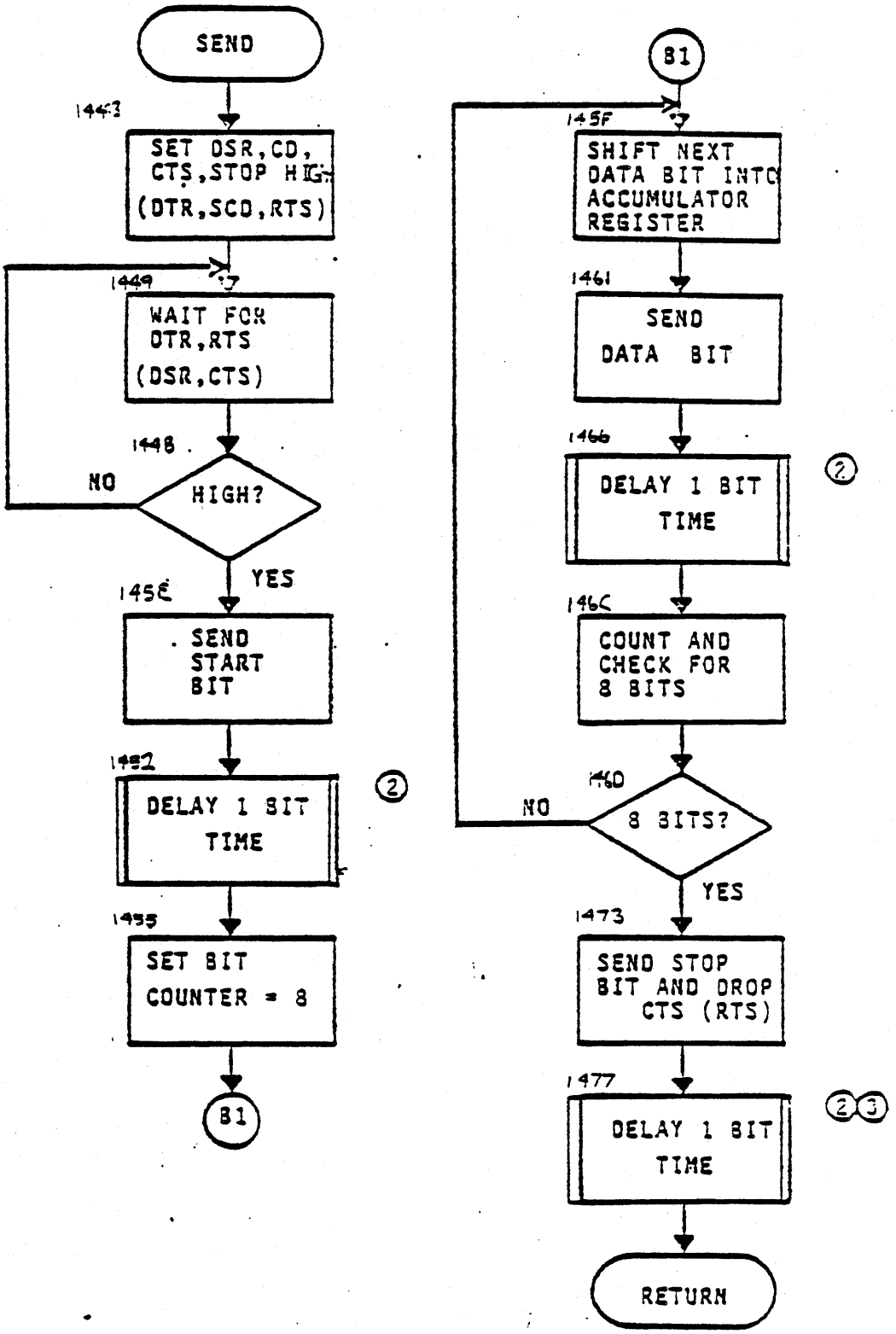
SET PARITY BIT (OR MASK PARITY BIT WITH ANAI 7F)

DELAY ½ BIT TIME

RESTORE REGS

OUTPUT TO DUMMY PORT FOR VIEWING ON ANALYZER.

EXIT WITH RECEIVED DATA IN REGISTER A.



RS232 SEND ROUTINE

TITLE ((SEND ONE CHARACTER. UPON ENTRY, CHARACTERMUST BE IN REGISTER A.)) DATE

14	4	0	E5	(SEND)	PSP	HL
		1	D5		PSP	DE
		2	5F		LDE	A
		3	3E		LDAI	
		4	0F		-	0F
		5	D3		OPA	
		6	C0		-	C0
		7	DB	FIND DTR	IPA	
		8	C0		-	C0
		9	E6		ANAI	
		A			-	02 (OR 0A)
		B	CA		JP	21
		C	47		-	FIND DTR
		D	14		-	
		E	3E		LDAI	
		F	0E		-	0E
14	5	0	D3		OPA	
		1	C0		-	C0
		2	CD		JS	
		3	E3		-	(BIT AT)
		4	15		-	
		5	16		LDDI	
		6	08		-	COUNT 8
		7	16		LDDI	
		8	08		-	08
		9	16		LDDI	
		A	08		-	08
		B	00		NOP	
	5	C	3E	SND·NXT	LDAI	
		D	00		-	00
		E	7B		LDA	E
		F	1F		RRAC	
14	6	0	5F		LDE	A
		1	3E		LDAI	
		2	07		-	07
		3	17		RLAC	
		4	D3		OPA	
		5	C0		-	C0
		6	CD		JS	
		7	E3		-	(BIT AT)
		8	15		-	
		9	3E		LDAI	
		A	00		-	00
		B	00		NOP	
		C	15		DCD	
		D	C2		JP	20
		E	5C		-	SND·NXT
		F	14		-	
14	7	0	F5		PSP	AF
		1	F1		PLP	AF
		2	00		NOP	
		3	3E		LDAI	
		4	0D		-	0D
		5	D3		OPA	
		6	C0		-	C0
		7	CD		JS	
		8	E3		-	(BIT AT)
		9	15		-	
		A	D1		PLP	DE
		B	E1		PLP	HL
		C	C9		RTS	
		D				
		E				
		F				

T SET DSR, CD; CTS ACTIVE, WITH STOP BIT (OR DTR, SCD, RTS, STOP)

T LOOK FOR RTS (CTS) AND/OR DTR (DSR)
NOTE: IF 7301 USED AS DATA TERMINAL USE 02; IF DATA SET USE CA.
← CA OR 02

T IF NOT THERE, WAIT FOR IT

T SEND START BIT

T DELAY 1 BIT TIME

T SET BIT COUNTER = 8

T (DELAY ADJUST)

T (DELAY ADJUST)

T SHIFT DATA BIT INTO REGISTER A

T SEND DATA BIT

T DELAY 1 BIT TIME

T (DELAY ADJUST)

T COUNT AND CHECK FOR 8 BITS

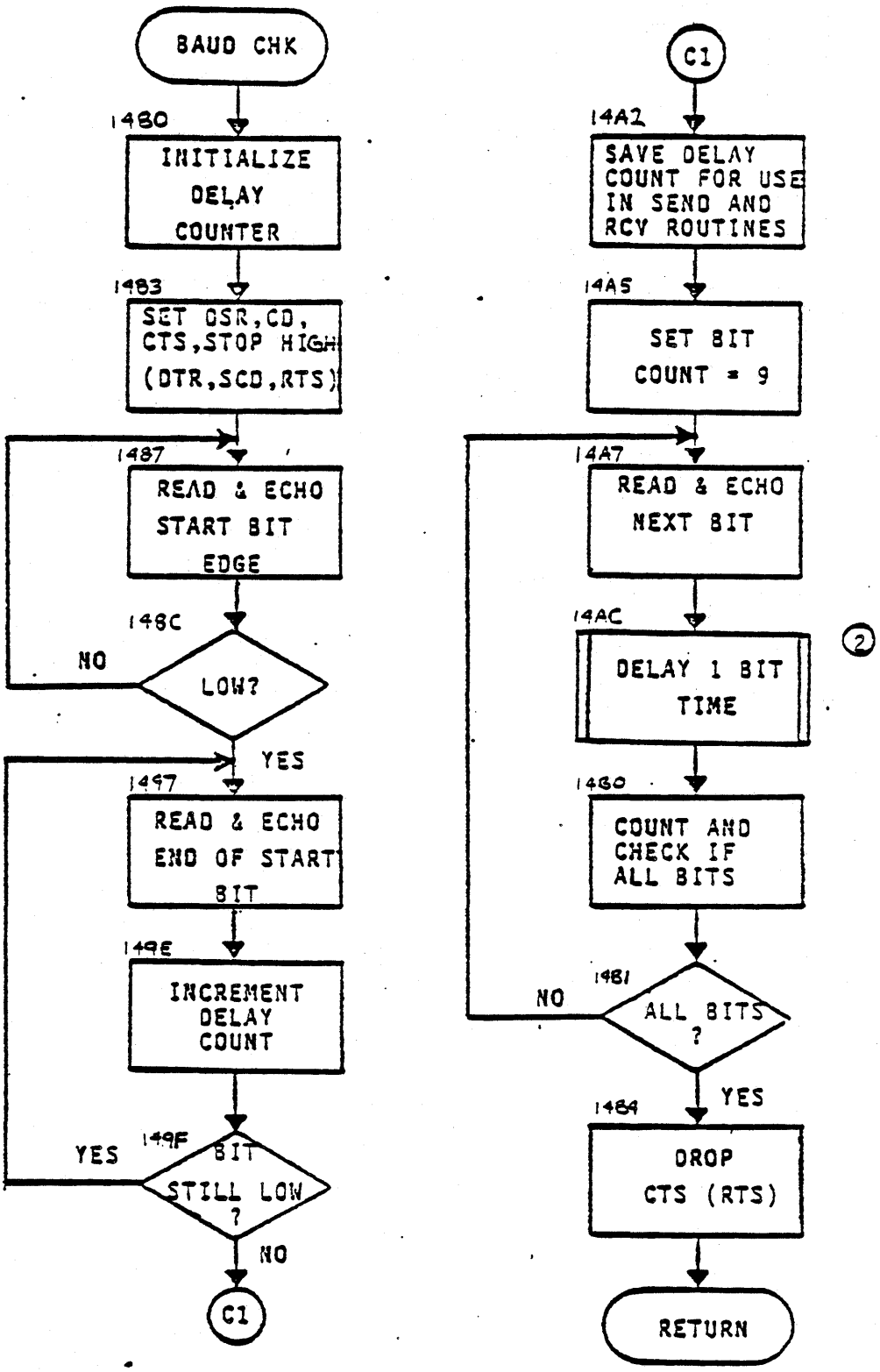
T (DELAY ADJUST)

T SEND STOP BIT, DROP CTS (RTS)

T DELAY 1 BIT TIME (FOR 2 STOP BITS, JUMP TO (2BIT AT))

T RESTORE REGISTERS

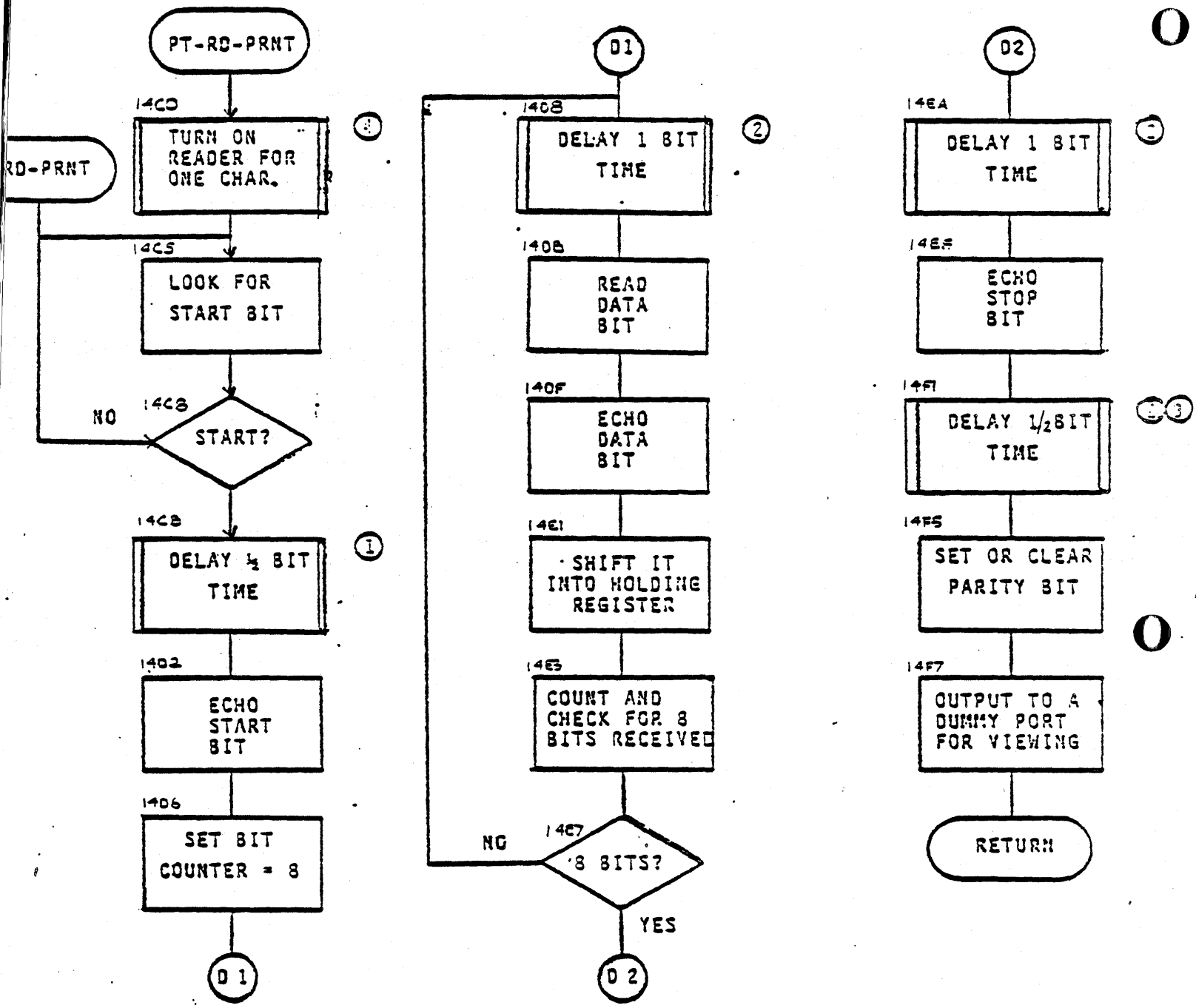
EXIT



RS232 BAUD RATE DETERMINATION ROUTINE

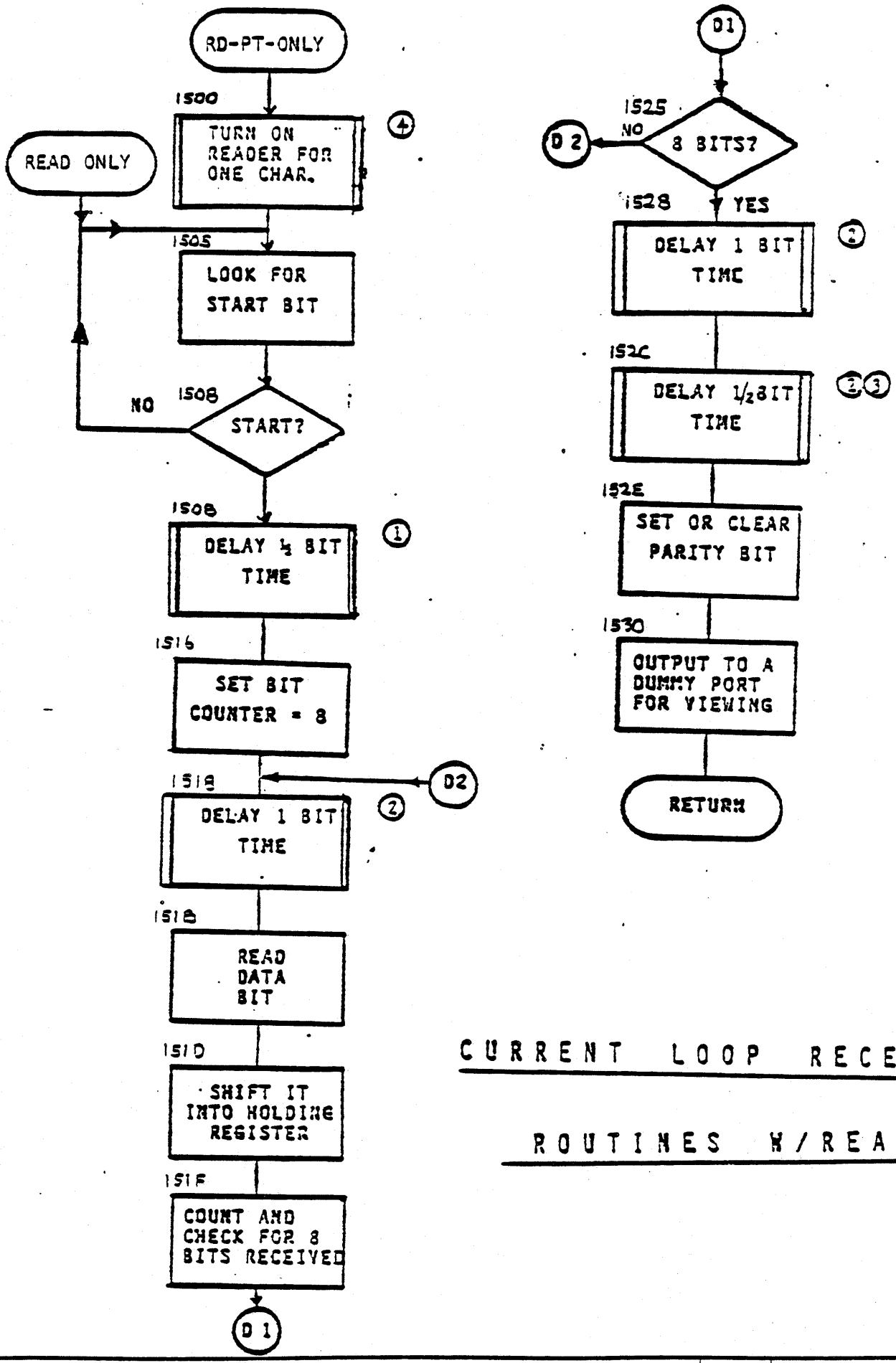
TITLE BAUD RATE CHECK-RS232 DATE 3-6-80

14	8	0	21	(BAUD CHK)	LDPI	HL	INITIALIZE COUNTER=-1
		1	FF		-	FF	
		2	FF		-	FF	
		3	3E		LDAI		SET DSR, CD, CTS, STOP ACTIVE
		4	OF		-	OF	(OR DTR, SCD, RTS, STOP)
		5	D3		OPA		
		6	CO		-	CO	
		7	DB	START CHK	IPA		LOOK FOR START BIT EDGE
		8	CO		-	CO	
		9	D3		OPA		(ECHO DATA)
		A	CO		-	CO	
		B	1F		RRAC		
		C	DA		JP	C1	
		D	87		-	START CHK	
		E	14		-		
		F	3E		LDAI		8085 Z80
14	9	0	07		-	07	(t = 1 time state) 148t (151t)
		1	3D	AT PAD	DCA		
		2	C2		JP	30	
		3	91		-	AT PAD	
		4	14		-		
		5	03		ICP	BC	
		6	0B		DCP	BC	
		7	DB	STOP CHK	IPA		LOOK FOR STOP BIT EDGE
		8	CO		-	CO	
		9	D3		OPA		(ECHO DATA)
		A	CO		-	CO	
		B	00		NOP		* REPLACE WITH DCP BC 4TNS
		C	00		NOP		FOR Z80
		D	1F		RRAC		
		E	23		ICP	HL	COUNT TIMES THRU LOOP N=(HL)
		F	D2		JP	C9	
14	A	0	97		-	STOP CHK	
		1	14		-		
		2	22		STPD	HL	SAVE LOOP COUNT AS TIME DELAY
		3			-	DELAY COUNT	COUNT FOR (SEND), (RCV & ECHO) ROUTINES
		4			-		
		5	3E		LDAI		SET BIT COUNT = 9
		6	09		-	09	
		7	F5	ALL BITS	PSP	AF	WAIT FOR END OF CHARACTER
		8	DB		IPA		
		9	CO		-	CO	
		A	D3		OPA		
		B	CO		-	CO	
		C	CD		JS		
		D	E3		-	(BITAT)	
		E	15		-		
		F	F1		PLP	AF	
14	B	0	3D		DCA		
		1	C2		JP	E0	
		2	A7		-	ALL BITS	
		3	14		-		
		4	3E		LDAI		DROP CTS (RTS)
		5	OD		-	OD	
		6	D3		OPA		
		7	CO		-	CO	
		8	C9		RTS		EXIT
		9					
		A					
		B					
		C					
		D					
		E					
		F					



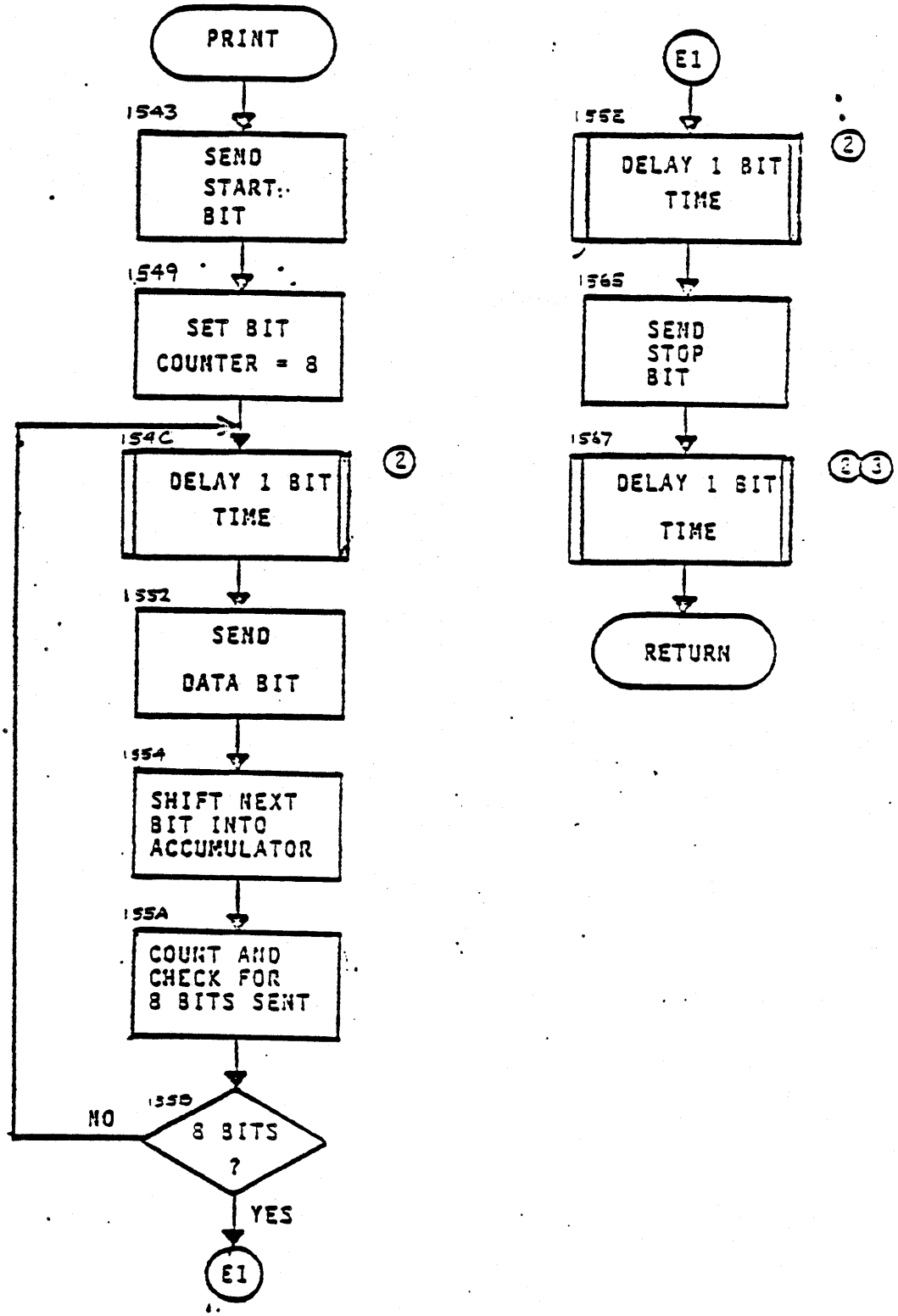
CURRENT LOOP RECEIVE ROUTINES W/READER

						TITLE 7301 CURRENT LOOP ROUTS DATE 3-10-80
14	C	0	CD	(PT-RD-PRNT)	JS	START_READER
		1	C0		-	(RD-PT)
		2	15		-	
		3	E5	(RD-PRNT)	PSP	HL // READ & ECHO 1 CHARACTER UPON //
		4	D5		PSP	DE // EXIT CHARACTER IS IN REGISTER A //
		5	DB	RD-START	IPA	LOOK FOR START BIT
		6	C1		-	C1 (CØ)
		7	1F		RRAC	
		8	DA		JP	C1
		9	C5		-	RD-START
		A	14		-	
		B	CD		JS	DELAY ½ BIT TIME
		C	E6		-	(½ BIT AT)
		D	15		-	
		E	E3		XCPT	HL (DELAY ADJUST)
		F	E3		XCPT	HL
14	D	0	23		INCP	HL
		1	2B		DCP	HL
		2	3E		LDAI	ECHO START BIT
		3	00		-	00
		4	D3		OPA	
		5	C1		-	C1 (CØ)
		6	1E		LDEI	SET BIT COUNTER = 8
		7	80		-	80
		8	CD	RD-8-BITS	JS	DELAY ½ BIT TIME
		9	E3		-	(BIT AT)
		A	15		-	
		B	DB		IPA	READ DATA BIT
		C	C1		-	C1 (CØ)
		D	E6		ANAI	ECHO IT
		E	01		-	01
		F	D3		OPA	
14	E	0	C1		-	C1 (CØ)
		1	1F		RRAC	SHIFT IT INTO REGISTER E
		2	7B		LDA	E
		3	1F		RRAC	
		4	5F		LDE	A
		5	23		INCP	HL (DELAY ADJUST)
		6	2B		DCP	HL
		7	D2		JP	C0 CHECK FOR 8 BITS
		8	D8		-	RD-8-BITS
		9	14		-	
		A	CD		JS	DELAY 1 BIT TIME
		B	E3		-	(BIT AT)
		C	15		-	
		D	3E		LDAI	ECHO STOP BIT
		E	01		-	01
		F	D3		OPA	
14	F	0	C1		-	C1 (CØ)
		1	CD		JS	STOP BIT DELAY
		2	E6		-	(½ BIT AT)
		3	15		-	
		4	7B		LDA	E SET PARITY BIT (USE ANAI 7F TO
		5	F6		ORAI	CLEAR IT, IF DESIRED)
		6	80		-	80 (PARITY)
		7	D3		OPA	OUTPUT CHARACTER TO DUMMY PORT
		8			-	XX FOR VIEWING ON ANALYZER
		9	D1		PLP	DE
		A	E1		PLP	HL
		B	C9		RTS	EXIT WITH CHARACTER IN REGISTER A
		C				
		D				
		E				
		F				



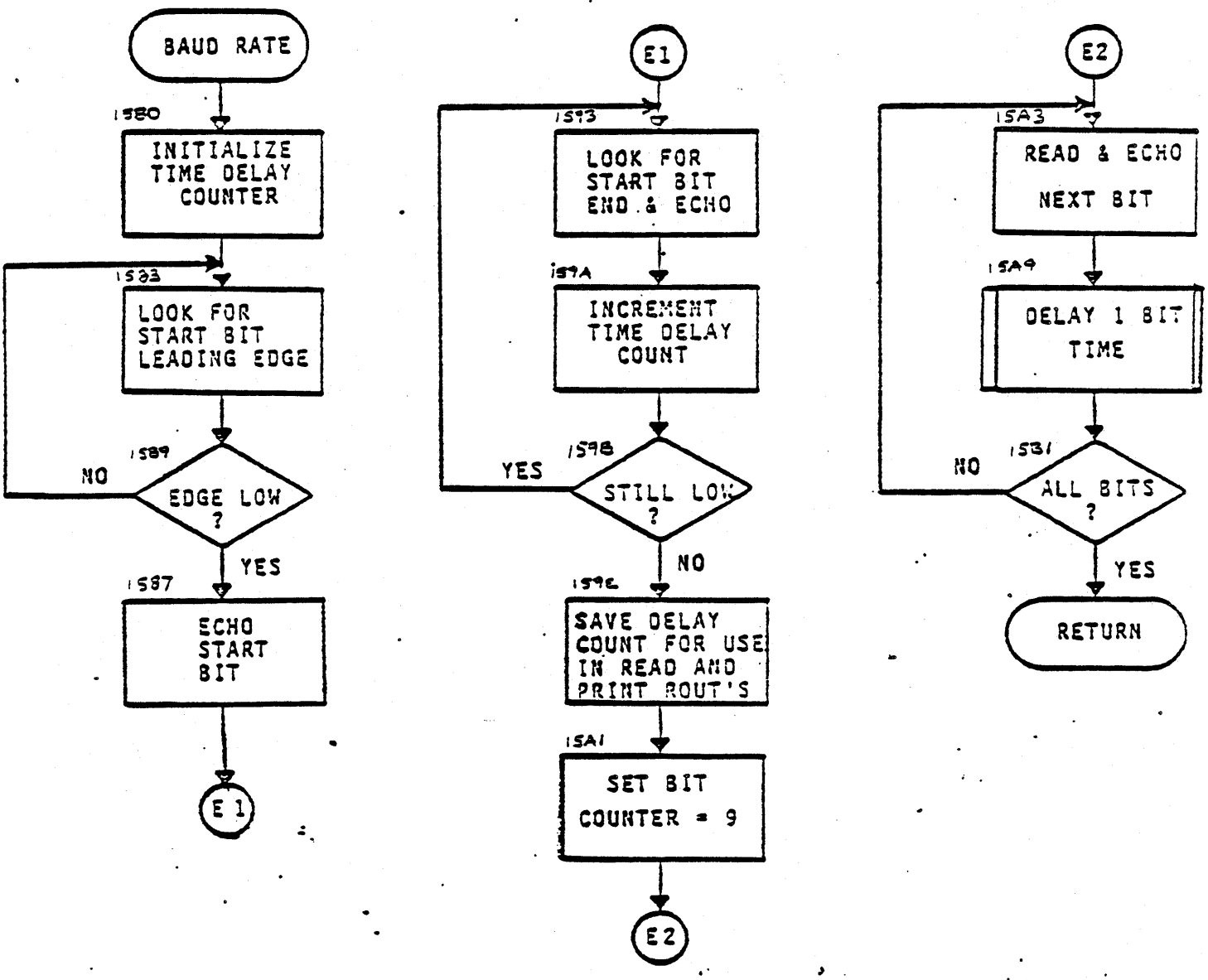
CURRENT LOOP RECEIVE ROUTINES W/READER

PAGE	ADDR		LABEL	INSTR	OPERAND	TITLE / COMMENTS
						TITLE DATE
15	00	CD	(RDPT ONLY)	JS		START READER
	1	C0		-	(RD-PT)	
	2	15		-		
	3	E5	(RD ONLY)	PSP	HL	//READ 1 CHARACTER - UPON EXIT//
	4	D5		PSP	DE	//CHARACTER IS IN REGISTER A//
	5	DB	FIND SBIT	IPA		
	6	C1		-	C1 (OR C8)	LOOK FOR START BIT
	7	1F		RRAC		
	8	DA		JP	C1	
	9	05		-	FIND SBIT	
	A	15		-		
	B	CD		JS		DELAY ½ BIT TIME
	C	E6		-	(½ BIT AT)	
	D	15		-		
	E	3E		LDAI		(DELAY ADJUST)
	F	00		-	00	
15	10	3E		LDAI		
	1	04		-	04	
	2	3C	DELY	DCA		
	3	C2		JP	20	
	4	12		-	DELY	
	5	15		-		
	6	1E		LDEI		SET BIT COUNTER = 8
	7	80		-	80	
	8	CD	RD-EIGHT	JS		DELAY 1 BIT TIME
	9	E3		-	(BIT AT)	
	A	15		-		
	B	DB		IPA		READ DATA BIT
	C	C1		-	C1 (OR C8)	
	D	1F		RRAC		SHIFT IT INTO REGISTER E
	E	7B		LDA	E	
	F	1F		RRAC		
15	20	5F		LDE	A	
	1	23		ICP	HL	(DELAY ADJUST)
	2	2B		DCP	HL	
	3	3E		LDAI		
	4	00		-	00	
	5	D2		JP	C0	CHECK FOR ALL 8 BITS
	6	18		-	RD-EIGHT	
	7	15		-		
	8	CD		JS		DELAY 1 BIT TIME
	9	E3		-	(BIT AT)	
	A	15		-		
	B	7B		LDA	E	SET PARITY BIT (USE ANAI 7F
	C	E6		ORAI		TO CLEAR, IF DESIRED)
	D	80		-	80 (PARITY)	
	E	D3		OPA		OUTPUT TO DUMMY PORT FOR
	F			-	XX	VIEWING ON ANALYZER
15	30	CD		JS		STOP BIT DELAY
	1	E6		-	(½ BIT AT)	
	2	15		-		
	3	D1		PLP	DE	
	4	E1		PLP	HL	
	5	C9		RTS		EXIT WITH CHARACTER IN REGISTER A
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					



CURRENT LOOP SEND ROUTINE

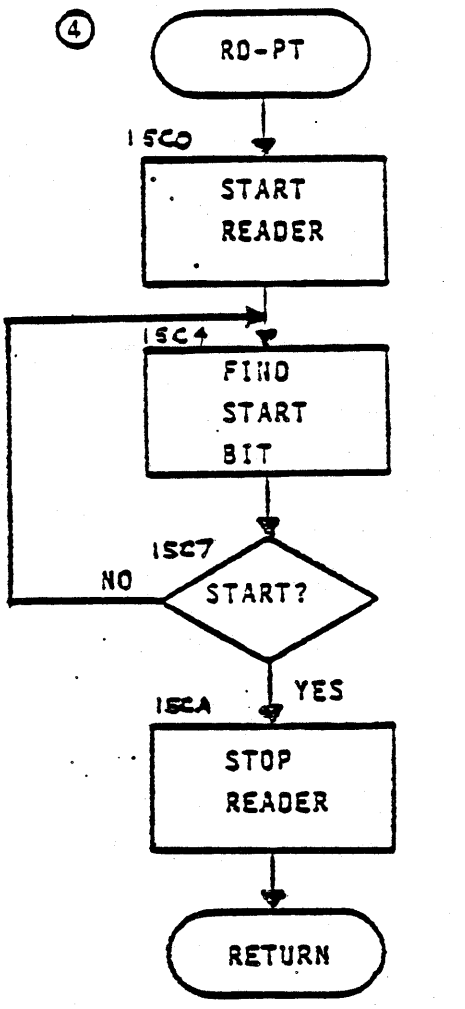
PAGE ADDR	LINE ADDR	OBJECT CODE	LABEL	MNEMONIC	OPERAND	TITLE / COMMENTS
						TITLE DATE
15	40	E5	(PRINT)	PSP	HL	//PRINT CHARACTER IN REGISTER A//
	1	D5		PSP	DE	
	2	5F		LDE	A	
	3	3E		LDAI		SEND START BIT
	4	00		-	00	
	5	D3		OPA		
	6	C1		-	C1 (CØ)	
	7	F5		PSP	AF	(DELAY ADJUST)
	8	E1		PLP	AF	
	9	16		LDDI		SET BIT COUNTER = 8
	A	08		-	08	
	B	00	SND-8-BITS	NOP		(DELAY ADJUST)
	C	CD		JS		DELAY 1 BIT TIME
	D	E3		-	(BIT AT)	
	E	15		-		
	F	3E		LDAI		SEND DATA BIT
15	50	01		-	01	
	1	A3		ANA	E	
	2	D3		OPA		
	3	C1		-	C1 (CØ)	
	4	1F		RRAC		SHIFT NEXT BIT INTO REGISTER A
	5	7B		LDA	E	
	6	1F		RRAC		
	7	5F		LDE	A	
	8	23		ICP	HL	(DELAY ADJUST)
	9	00		NOP		
	A	15		DCD		COUNT AND CHECK FOR 8 BITS
	B	C2		JP	20	
	C	48		-	SND-8-BITS	
	D	15		-		
	E	CD		JS		DELAY 1 BIT TIME
	F	E3		-	(BIT AT)	
15	60	15		-		
	1	00		NOP		(DELAY ADJUST)
	2	00		NOP		
	3	3E		LDAI		SEND STOP BIT
	4	01		-	01	
	5	D3		OPA		
	6	C1		-	C1 (CØ)	
	7	CD		JS		STOP BIT DELAY
	8	E3		-	(BIT AT)	USE (2 BIT AT) FOR 2 STOP BITS
	9	15		-		
	A	D1		PLP	DE	
	B	E1		PLP	HL	
	C	C9		RTS		EXIT
	D					
	E					
	F					
15	70					
	1					
	2					
	3					
	4					
	5					
	6					
	7					
	8					
	9					
	A					
	B					
	C					
	D					
	E					
	F					



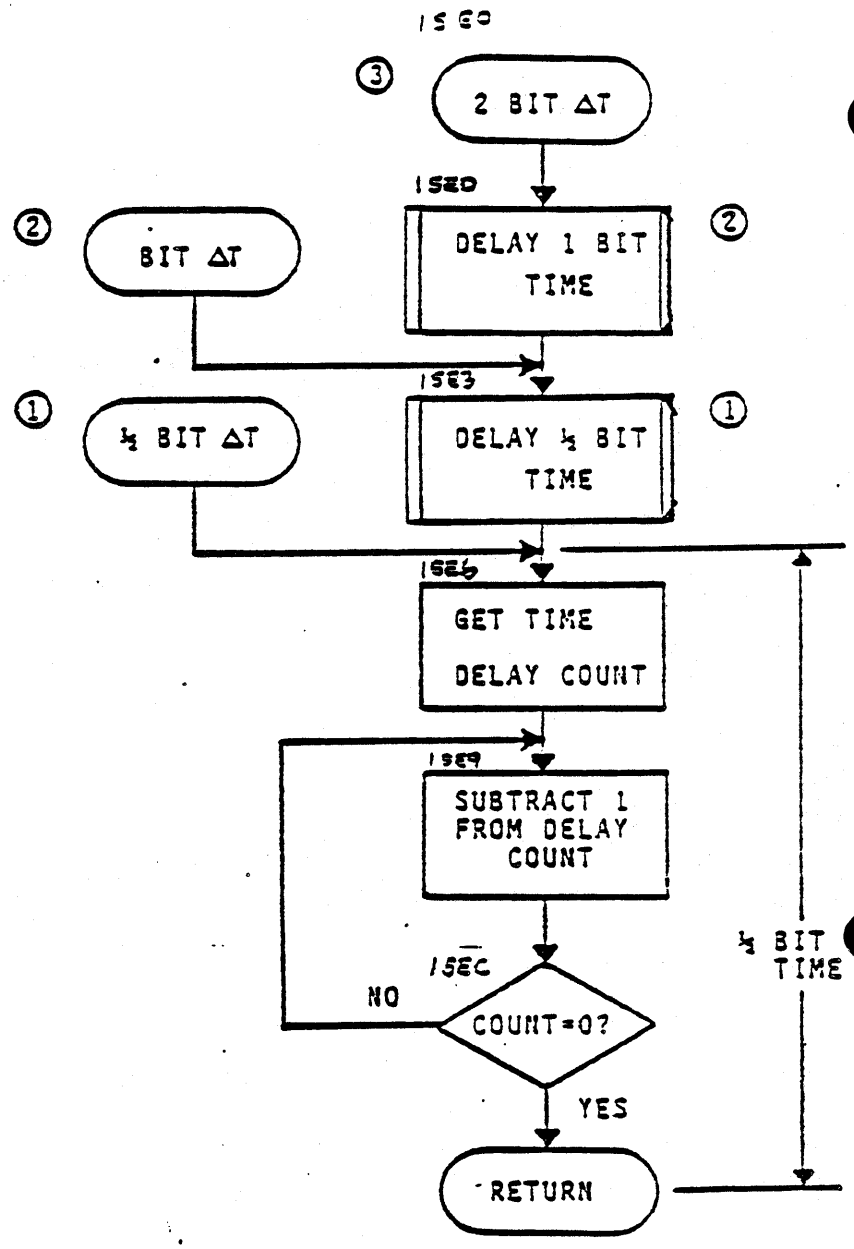
CURRENT LOOP BAUD RATE DETERMINATION

TITLE BAUD_RATE_CHECK - CUR_LOOP DATE 3-10-80

PAGE	LINE	LOC	OBJ CODE	LABEL	MNEMONIC	OPERAND	COMMENTS
15	90	21		(BAUD RATE)	LDPI	HL	; INITIALIZE DELAY COUNTER = 1
		1	FF		-	FF	
		2	FF		-	FF	
		3	DB	START EDGE	IPA		; LOOK FOR START BIT LEADING
		4	C1		-	C1 (CØ)	; EDGE
		5	E6		ANAI		; ECHO IT
		6	01		-	01	
		7	D3		OPA		
		8	C1		-	C1 (CØ)	
		9	C2		JP	30	
		A	83		-	START EDGE	
		B	15		-		; t=1 time state (8085) 149t (280) 154t
		C	3E		LDAI		
		D	06		-	06	
		E	00		NOP		
		F	3C	WAIT	DCA		
15	9 0		C2		JP	30	
	1		8F		-	WAIT	
	2		15		-		
	3		DB	END EDGE	IPA		; LOOK FOR START BIT TRAILING
	4		C1		-	C1 (CØ)	; EDGE
	5		E6		ANAI		; ECHO IT
	6		01			01	
	7		D3		OPA		; N=(HL) (8085) 47N± (280) 47N±
	8		C1		-	C1 (CØ)	
	9		00		NOP		
	A		23		ICP	HL	; INCREMENT TIME DELAY COUNT
	B		CA		JP	31	
	C		93		-	END EDGE	
	D		15		-		
	E		22		STPD	HL	; SAVE DELAY COUNT FOR USE
	F				-	DELAY COUNT	; IN (RD.PRNT) & (PENTIR-OUTIVES
15	A 0				-		
	1		0E		LDCI		; READ & ECHO REMAING BITS
	2		09		-	09	
	3		DB	XTRA BITS	IPA		; READ BIT
	4		C1		-	C1 (CØ)	
	5		E6		ANAI		; ECHO IT
	6		01		-	01	
	7		D3		OPA		
	8		C1		-	C1 (CØ)	
	9		CD		JS		; DELAY 1 BIT TIME
	A		E3		-	(BIT AT)	
	B		15		-		
	C		23		ICP	HL	; (DELAY ADJUST)
	D		23		ICP	HL	
	E		2B		DCP	HL	
	F		2B		DCP	HL	
15	B 0		0D		DCC		; CHECK FOR ALL BITS
	1		C2		JP	30	
	2		A3		-	XTRA BITS	
	3		15		-		
	4		C9		RTS		; EXIT
	5						
	6						
	7						
	8						
	9						
	A						
	B						
	C						
	D						
	E						
	F						



CURRENT LOOP READER CONTROL



DELAY ROUTINE SERIAL I/O

TITLE DATE

	C0	3E	(RD-PT)	LDAI	// READ 1 CHARACTER FROM READER //
	1	03		- 03	↓START READER
	2	D3		OPA	
	3	C1		- C1	↓
	4	DB	FIND START	IPA	↓FIND START BIT
	5	C1		- C1	
	6	1F		RRAC	
	7	DA		JP C1	
	8	C4		- FIND START	
	9	15		-	↓
	A	3E		LDAI	↓STOP READER
	B	01		- 01	
	C	D3		OPA	
	D	C1		- C1	↓
	E	C9		RTS	EXIT.
	F				
	0				
	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	A				
	B				
	C				
	D				
	E				
	F				
15	F0	CD	(2 BIT AT)	JS	↓DELAY 2 BIT TIMES
	1	E3		- (BIT AT)	
	2	19		-	
	3	CD	(BIT AT)	JS	↓DELAY 1 BIT TIME
	4	E6		- (½ BIT AT)	
	5	19		-	
	6	2A	(½ BIT AT)	LDPD HL	↓DELAY ½ BIT TIME
	7			- DELAY COUNT	
	8			-	
	9	2B	(DELAY)	DCP HL	↓VARIABLE DELAY
	A	7C		LDA H	ΔT = 24 N t WHERE N = (HL)
	B	B5		ORA L	AND t = 1/f ; f = μP Clock Freq.
	C	C2		JP 20	
	D	E9		- (DELAY)	
	E	15		-	↓
	F	C9		RTS	EXIT
	0				
	1				
	2				
	3				
	4				
	5				
	6				
	7				
	8				
	9				
	A				
	B				
	C				
	D				
	E				
	F				

APPENDIX A

TTY INTERFACE

The TTY connects to the 7301 Card via the 9 pin Serial Interface connector mounted below the card ejector. This interface within the 7301 is a three circuit six wire connection. The three, two-wire circuits are:

Data to TTY, 20 milliamp neutral loop receive.
Data from TTY, 20 milliamp neutral loop send.
Reader Control to TTY, 15 volt DC neutral loop reader control.

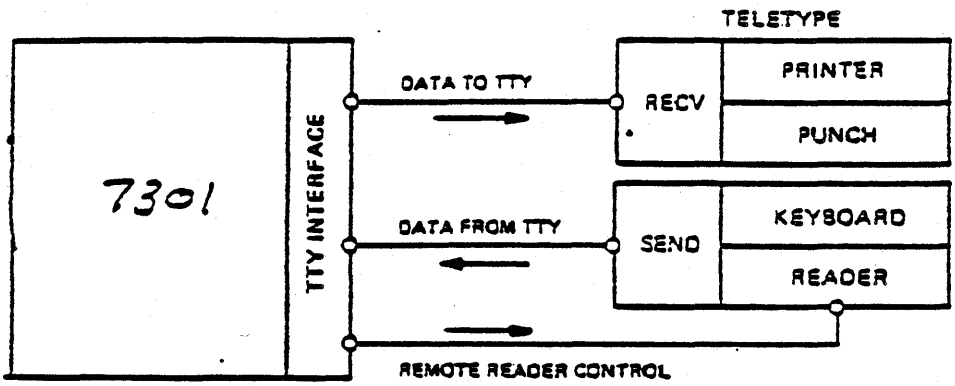


FIGURE A-1 TTY INTERFACE

This interface allows full-duplex send and receive, and remote reader control. The separate send and receive of full-duplex allows the system processor to edit the input data before printing and/or punching the output data. Remote reader control allows the system processor to operate the TTY reader at a rate compatible with the application.

TTY OPTIONS AND MODIFICATIONS

The TTY interface requires the following options and modification in the teletype unit:

1. 20 milliamp current loop option
2. Full duplex option
3. Remote reader control modifications

The 20 milliamp current loop and the full duplex connection are options available on the TTY unit. The remote reader control requires the addition of a 12 to 15 volt DC relay capable of switching the high voltage reader circuit inside the TTY.

A number of mini-computers specify a similar interface thus many TTY consoles have the required configuration. Detailed instructions for modifying a TTY to this configuration are given in the following paragraphs.

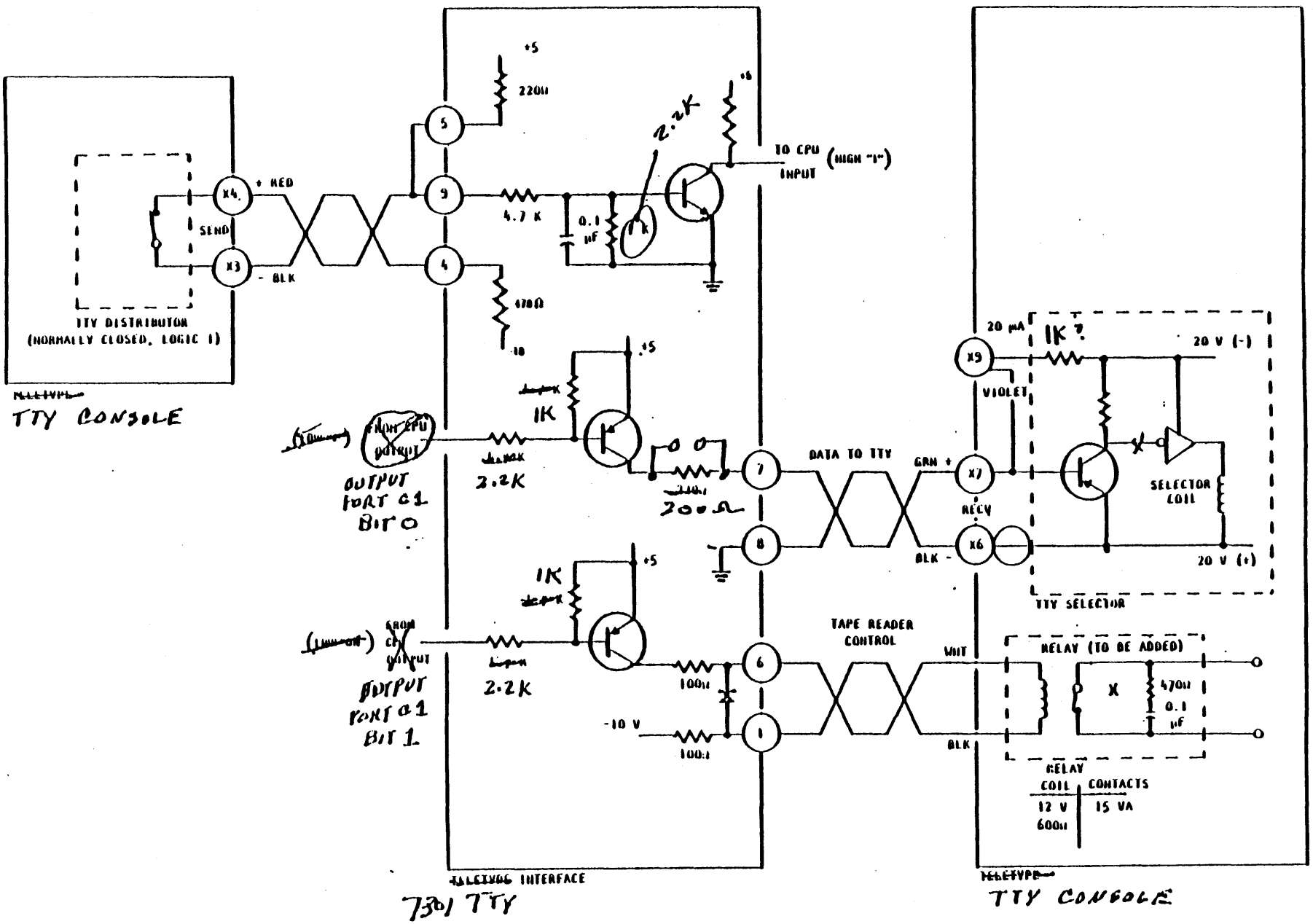


FIGURE A-2 – TTY Interface Circuit

REMOVING THE TTY HOUSING

It is necessary to remove the TTY housing to inspect or modify the TTY options.

1. Unplug the TTY from any power source.
2. Remove the roll of TTY printer paper from its cradle.
3. Remove the manual paper feed knob by pulling firmly.
4. Remove the mode select knob located on the right front by pulling firmly.
5. Remove the metal trim panel behind the mode select knob by prying downward.
6. Remove the 4 screws under the metal trim panel.
7. Remove the screw on the left side of the paper tape reader housing.
8. Remove the four knurled knobs along the lower rear edge of the housing.
9. Lift upward on the housing to remove, being careful of the controls on the paper tape reader as they clear their openings in the housing.

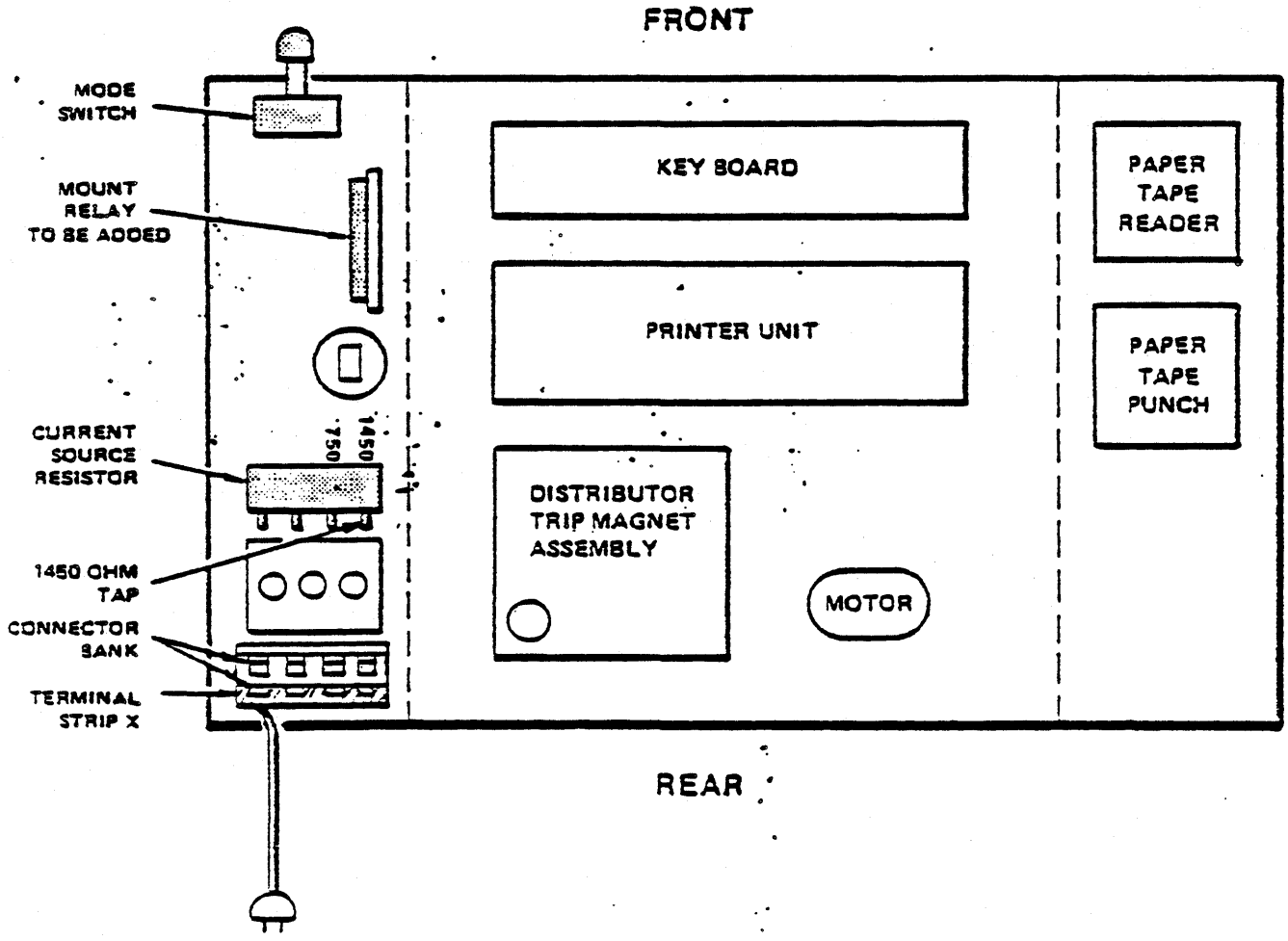


FIGURE A-3 TOP VIEW- TTY with Housing Removed

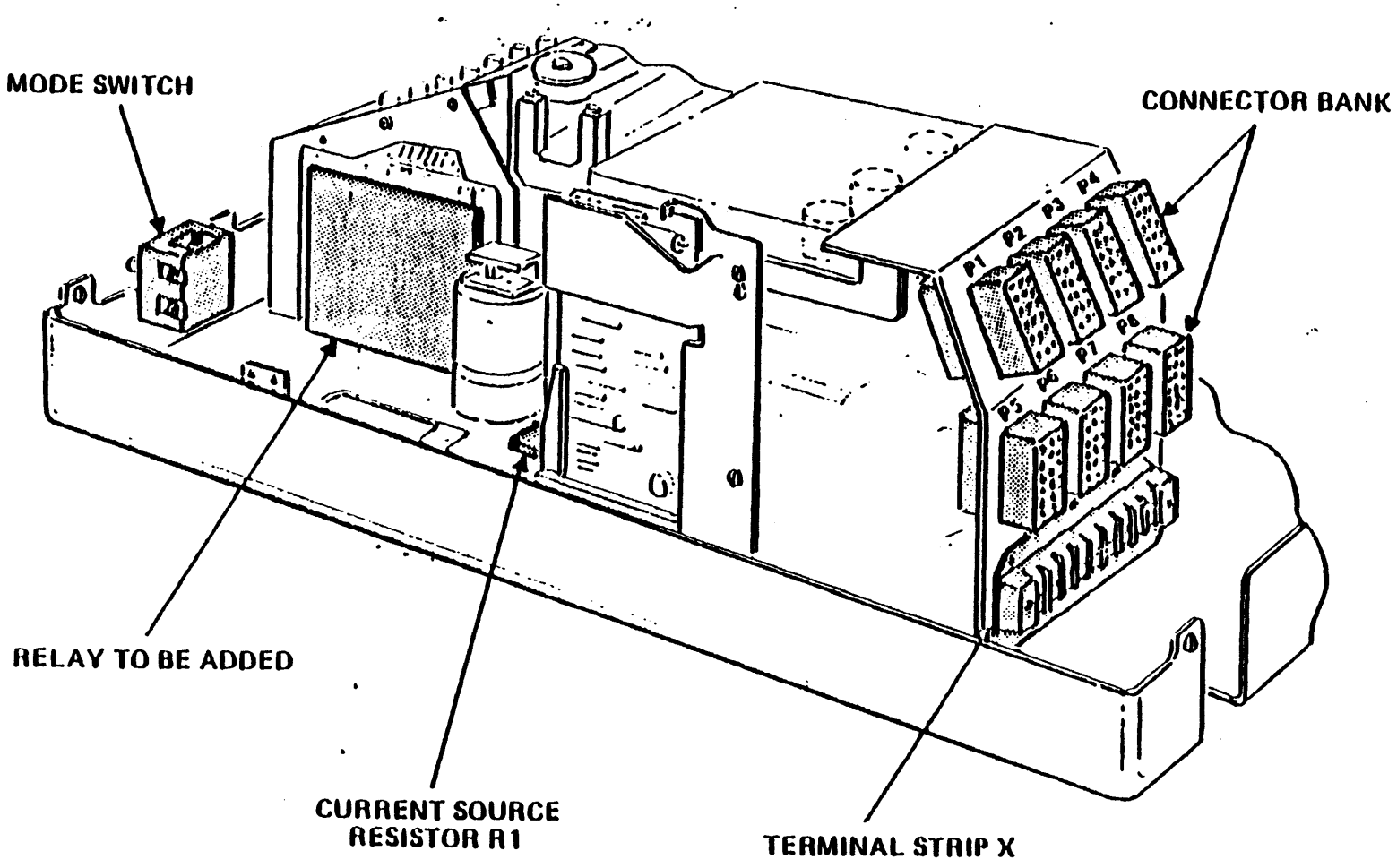


FIGURE A-4. SIDE VIEW- TTY with Housing Removed

CURRENT LOOP OPTION

The TTY send and receive current loop can be optionaly selected to work from either 20 milliamp or 60 milliamp. When the selection is made both the internal current source and the selector drive current bias must be modified to be compatible.

INTERNAL CURRENT SOURCE

The internal current source is set to 20 milliamp by putting the blue wire on the 1450 ohm tap of power resister R1 located on the right side of the TTY.

SELECTOR DRIVE CURRENT BIAS

The selector drive current bias is set to 20 milliamp by optional wiring on terminal strip X located below the connector bank in the right rear corner of the TTY. In making this change various wiring configurations may be encountered as shown in figure A-5, depending on whether the unit has an elapsed time meter.

TTY WITHOUT ELAPSED TIME METER

A TTY without an elapsed time meter may be wired either as 1A or 1B of figure A-5 To modify for 20 milliamp:

If wired as 1A:
Do nothing; this is the correct connection for 20 milliamp without an elapsed time meter.

If wired as 1B:
Remove the violet wire form terminal X8 and move it to terminal X9 with the yellow wire.

TTY WITH ELAPSED TIME METER

A TTY with an elapsed time meter may be wired as 1A, 1B, 2A or 2B. To modify for 20 milliamp:

If wired as 1A:
Remove the black/green wire from X8, tape the exposed end and tie-back into the wire bundle. Locate a black wire and a blue wire on terminal X5. Move both wires from X5 to terminal X8.

If wired as 1B:
Remove the violet wire from X8 and move it to X9. Remove the black/green wire from X8; tape the exposed end and tie-back into the wire bundle. Locate a black wire and a blue wire connected on terminal X5. Move both wires from X5 to X8.

If wired as 2A:
Do nothing; this is the correct connection for 20 milliamp with an elapsed time meter.

If wired as 2B:
Remove the black wire and blue wire from X9. Remove the violet wire and black/green wire form X8. Connect the black wire and blue wire to X8. Connect the violet wire to X9. Locate the yellow wire taped back in the wire bundle. Connect the yellow wire to X9. Tape the exposed end of the black/green wire and tie-back into wire bundle.

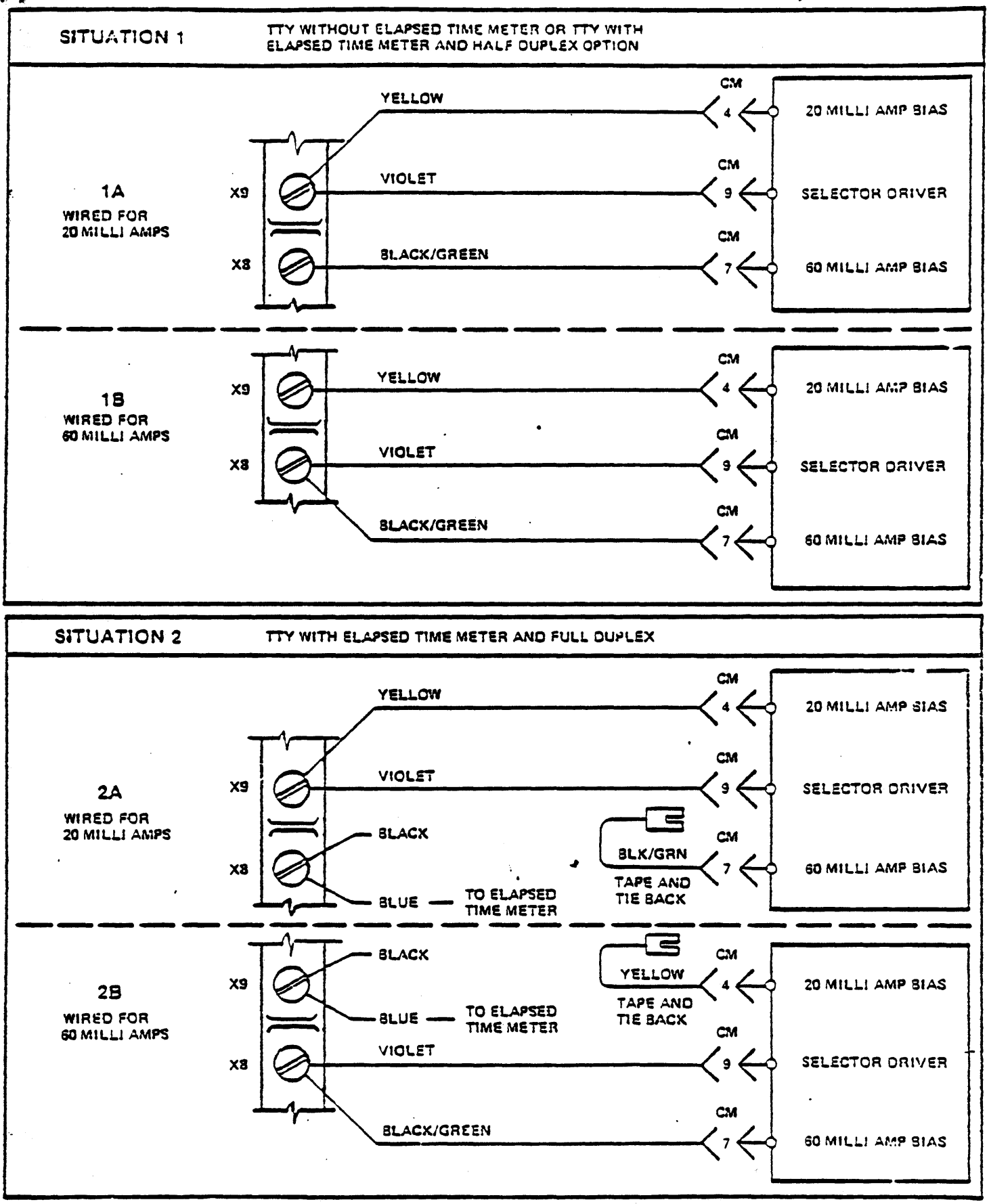


FIGURE A-5 = Current Loop Option

FULL DUPLEX OPTION

The full duplex option is wired into the TTY on terminal strip X located below the connector bank in the right rear corner of the unit.

If the TTY is wired for half-duplex, terminal strip X should appear as in Figure A-6

If the TTY is wired for full-duplex, terminal strip X should appear as in Figure A-7

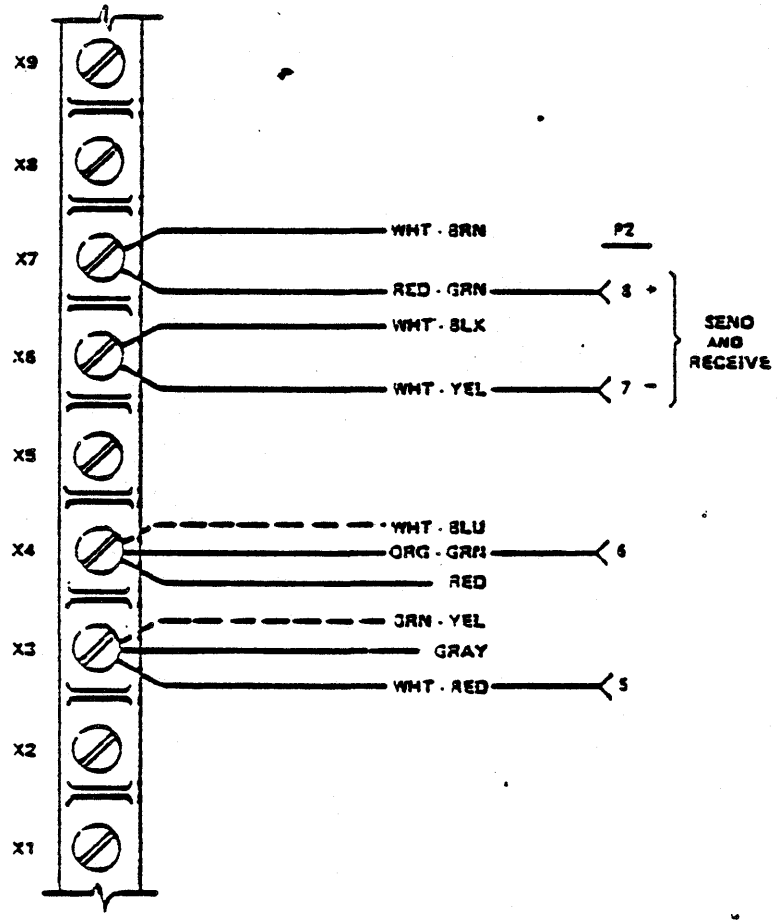


Figure A-6 TTY Half-Duplex Option

To convert from half-duplex to full-duplex:

1. Confirm that screw lug X5 has no wires connected. If there is a BLACK wire and a BLUE wire on X5 an elapsed time meter is installed. Refer to the current LOOP option for instructions on moving the black wire and blue wire from X5 to X8.
2. Move the white-blue wire form screw lug X4 to X5.
3. Move the brown-yellow wire from screw lug X3 to X5.

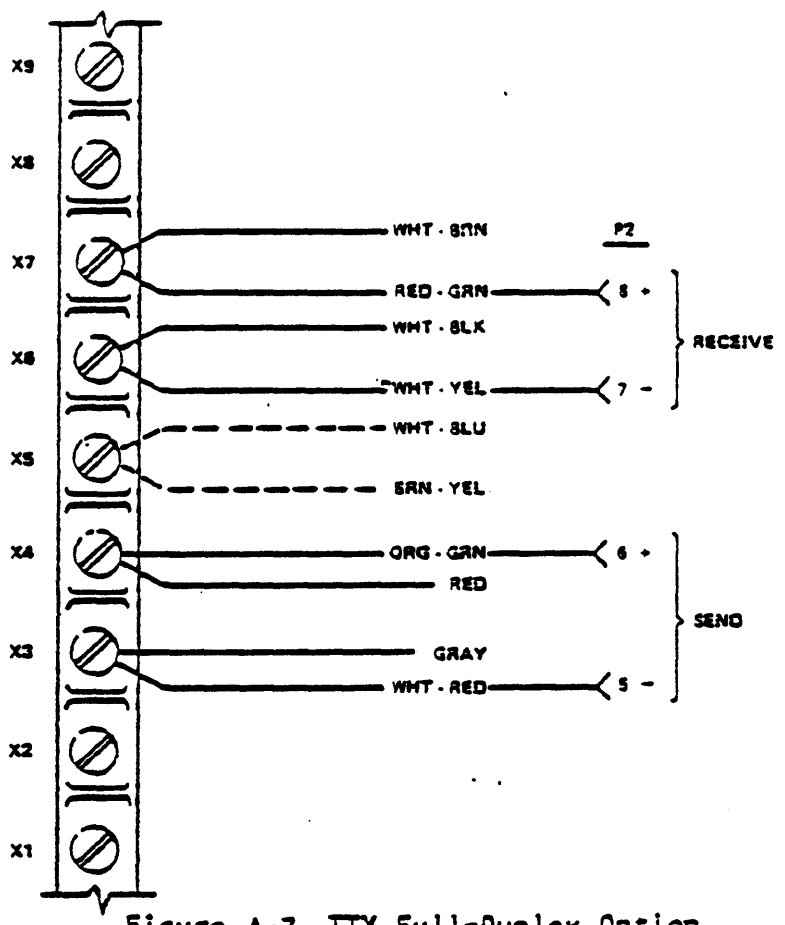


Figure A-7 TTY Full-Duplex Option

REMOTE TTY READER CONTROL

The wiring of standard teletype does not allow the TTY paper tape reader to be used remotely as a stand-alone input device. By modifying the distributor trip circuit for remote operation the TTY reader can be advanced one character at a time for total reader control.

TTY circuits operate from 115 VAC or 48VDC requiring remote logic control circuits to be relay buffered. Two basic reader circuits will be encountered in TTY reader modification, Manual and Automatic. The Manual and Automatic readers are identified by the reader control switch located on the Paper Tape reader. The Manual reader has a three position switch labeled ON, OFF, FREE. The Automatic reader has a four position switch labeled MANUAL START, AUTO, MANUAL STOP, FREE.

MANUAL READER OPERATION

The 115 VAC manual reader circuit is operated in either the LINE or LOCAL modes by the reader ON-OFF switch located on the reader.

The manual reader circuit can be controlled remotely by adding a relay to control the reader trip coil in the LINE mode. Modifying the reader circuit as shown in Figure A-8 allows normal operation in the LOCAL mode and remote control in the LINE mode.

MANUAL READER MODIFICATION

Locate Plug P4.
Locate the blue wire connecting P4 pin 3 and P4 pin 11.
Locate the orange wire on screw lug L1 of the mode switch.
Locate L2 of the mode switch (no wires)
Locate the orange/gray wire on screw lug 1 of the mode switch.
If the wire colors do not agree, do not proceed with this modification unless the connections can be verified to agree with those of Figure A-8 .

2. Cut the blue wire near P4 pin 3
3. Splice wire A from the new relay to the portion of the blue wire still connected to P4 pin 11.
4. Connect wire B to screw lug L2 of the mode switch.
5. Connect wire C to screw lug 1 of the mode switch.

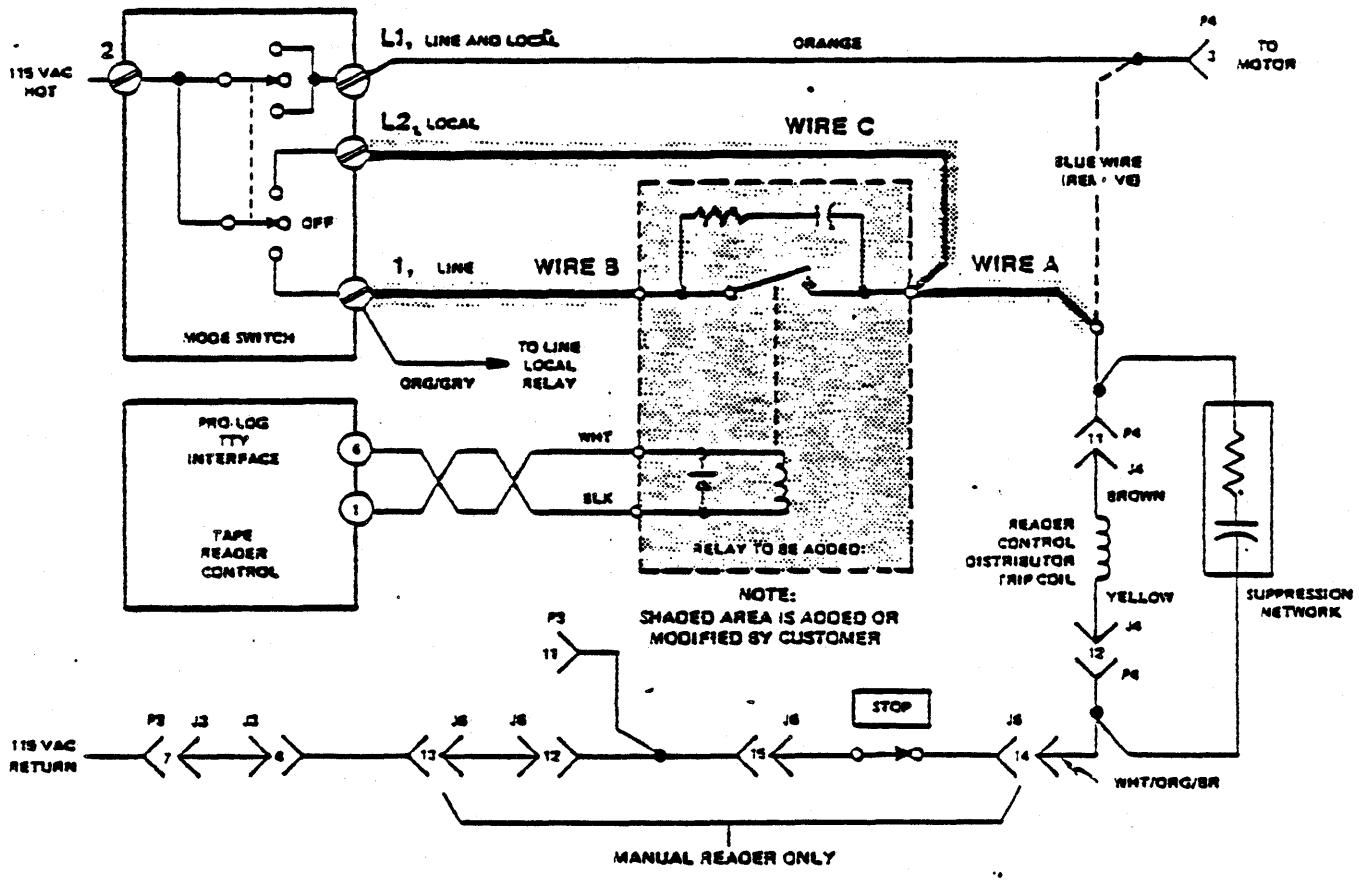


Figure A-8 TTY Modification (Manual Reader)

AUTO READER OPERATION

The 115 VAC automatic reader circuit is normally open due to the TDC relay contact. The 48 VDC TDC relay can be operated by the momentary MANUAL START switch on the reader or by the DC1 data function. Once the TDC relay operates, it holds itself energized until the momentary MANUAL STOP switch on the reader is activated. The DC3 and ENQ data functions will also stop the reader.

The automatic reader circuit can be controlled remotely by adding a relay to control a contact closure in parallel across the TDC relay contact as shown in Figure A-9..

AUTO READER MODIFICATION

1. Locate Jack J6 connected to plug P6.
   Locate the yellow/green wire at J6-13.
   Locate the blue wire at J6-14.
   If the wire colors do not agree, do not proceed with this modification unless the connection can be verified to agree with those of Figure A-9
2. Connect wire A from the new relay to the blue wire at J6-14.
3. Connect wire B from the new relay to the yellow/green wire at J6-13.

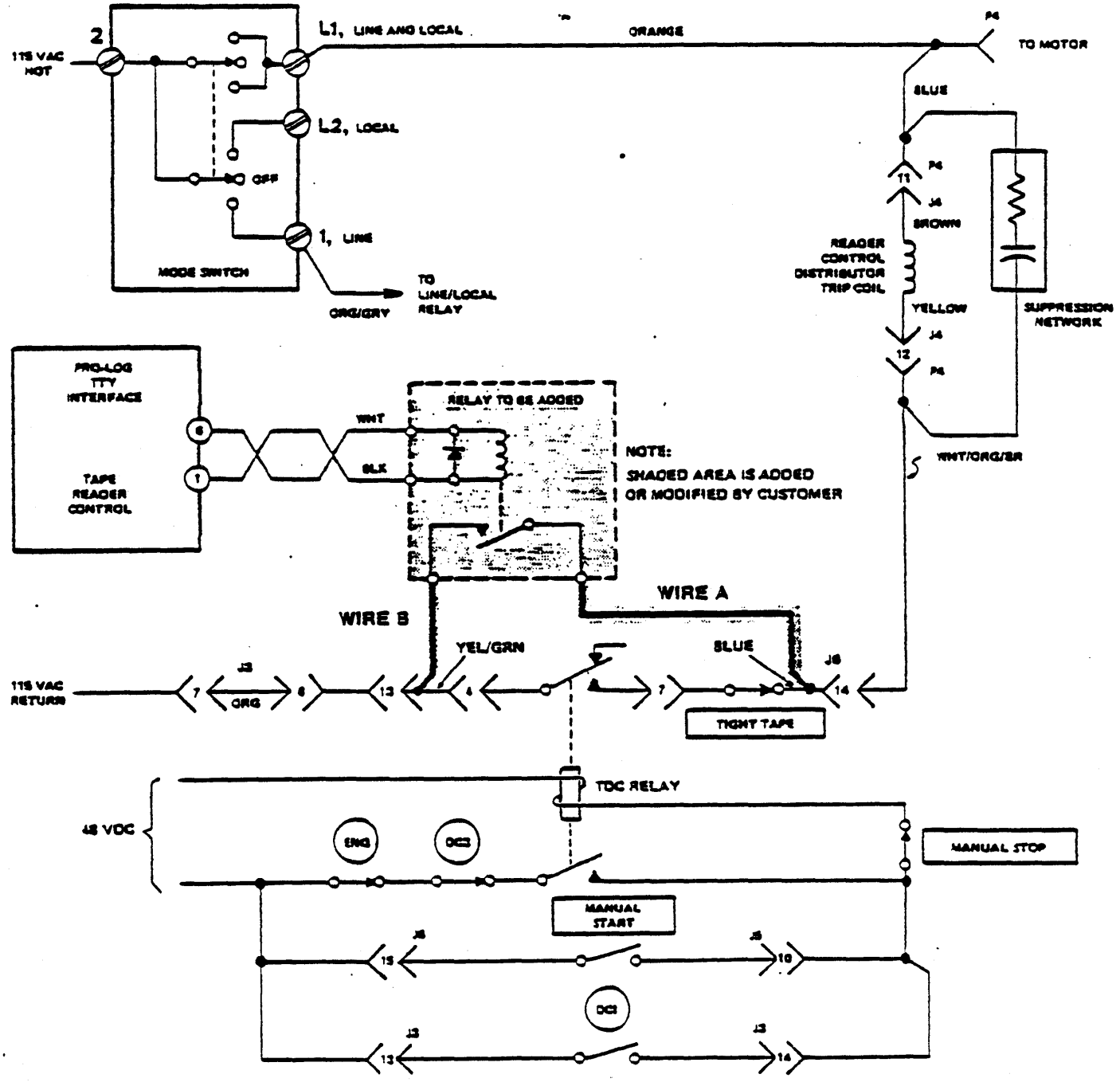


Figure A-9 TTY Modification (Auto Reader)

PRO-LOG
CORPORATION 2411 Garden Road Monterey, California 93940 Telephone (408) 372-4593
TWX: 910-360-7082