TECHNICAL REFERENCE MANUAL
MODEL PK-232 DATA CONTROLLER

ADVANCED ELECTRONICS APPLICATIONS, INC.

(Preliminary Release)

PREFACE TO THE PK-232 TECHNICAL REFERENCE MANUAL

Please read this preface in its entirety. It contains information about how to receive warranty service from AEA, the current software installed in your PK-232 and AEA's software update policy. This information is important; if you do not read it, you may damage your unit.

RF Interference Information To User

This PK-232 has been certified under the limits for Class B computing devices under Subpart J of Part 15 of the FCC rules, and is listed under FCC Identification Number DLX42690056.

This equipment generates and uses radio frequency energy. If it is not installed and used properly, that is, in strict accordance with AEA's instructions, it may cause interference to radio and TV reception. It has been type tested and has been found to comply with the limits of a Class B computing device in accordance with the specifications in Subpart J of Part 15 of the FCC rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio or TV reception, which can be determined by turning the PK-232 on and off, the user is encouraged to try and correct the interference using one or more of the following measures:

- Reorient the antenna of the device receiving interference.
- Relocate the computer with respect to this device.
- Plug the computer into different outlet so the computer and the device are on different branch circuits.

If necessary, the user should consult the dealer or an experienced Radio/TV technician for additional suggestions. The user may find 'How to Identify and Resolve Radio-TV Interference Problems', a booklet prepared by the FCC, helpful.

USE SHIELDED CABLE FOR ALL RS-232 CONNECTIONS

As part of its continuing program of product improvement, AEA reserves the right to make changes in this product's specifications. Changes will be made periodically to the information in this document. These changes will be incorporated in new issues of this manual.

There may be technical in accuracies or typographical errors in this document. Please address comments and corrections to AEA Incorporated, PO Box C2160, Lynnwood, WA 98036-0918. AEA reserves the right to incorporate and issue any information thus supplied in whatever manner it deems suitable without incurring and obligations whatever.

FIRST ISSUE - PRELIMINARY RELEASE (MAY 1987)
I/A/W Software Release Date 21-Jan-87

INTRODUCTION

Your AEA PK-232 Data Controller is the connection between your computer and radios. The PK-232 performs all of the control, modulation, demodulation, coding and encoding function required to establish and maintain data and text communications between your station and any other suitably-equipped station, as well as other communication facilities equipped for digital communications.

The PK-232's packet system software is derived from and compatible with the original TAPR TNCs and presents many of the advanced features of that design, coupled with significant enhancements based on experience gained by thousands of TAPR-equipped amateur packet stations worldwide.

This manual is your reference guide to the technical aspects of the PK-232, its maintenance and repair, as well as special software information for programmers and applications developers in digital Amateur Radio.

BATTERY BACK-UP

Your PK-232 uses batteries to back up the user-programmable values in the system. If you don't install batteries, you'll have to re-enter all of your personal system settings each time you turn on the PK-232. Your PK-232 will operate normally in all modes but will not retain your personalized parameters such as your call sign, until you install three AA-size batteries in the battery holder inside the chassis cover. We recommend that you choose alkaline batteries for this application.

- Remove the four screws from the sides and the two screws from the rear of the chassis. then lift off the PK-232's cover. Take care not to disturb the black or red wires that attach the battery holder to the printed circuit board.

- Find the positive and negative symbols embossed on the inside of the battery holder. Insert each battery, carefully matching the positive symbols on the battery with the positive symbols on the holder.

- Replace the cover and the six screws.

The battery back-up retains all the parameters except the time-of-day clock and the MHEARD (Monitor Heard) list. These two functions are controlled by the microprocessor.

TABLE OF CONTENTS

CHAPTER 1 – INTRODUCTION

Paragraph		Page	
1.1	Introduction	1-1	7
1.2	Scope	1-1	7
1.3	General Description	1-1	7
1.4	Specifications	1-1	7
1.4.1	Operating Modes	1-2	8
1.4.2	Modem Characteristics	1-2	8
1.4.3	Processor System	1-2	8
1.4.4	Input/Output Connections	1-2	8
1.4.5	Controls and Indicators	1-3	9
1.4.6	General	1-3	9

CHAPTER 2 – FUNCTIONAL DESCRIPTION

2.1	Major Sections	2-1	10
2.1.1	Analog Section	2-1	10
2.1.2	Digital Section	2-1	10
2.1.3	Input/Output Section	2-1	10
2.1.4	Display Section	2-1	10
2.1.5	Power Distribution	2-1	10

CHAPTER 3 – THEORY OF OPERATION

3.1	System Diagrams	3-1	11
3.1.1	Block Diagrams	3-1	11
3.1.2	Logic Diagrams	3-1	11
3.1.3	Schematic Diagrams	3-1	11
3.2	Analog Subsystem	3-2	12
3.2.1	Receive Function	3-2	12
3.2.1.1	Receive Circuits	3-2	12
3.2.2	Transmit Function	3-3	13
3.2.2.1	Transmit Circuits	3-3	13
3.3	Digital Subsystem	3-3	13
3.3.1	Z80A Central Processing Unit	3-3	13
3.3.2	Memory	3-4	14
3.3.3	8536 CIO Counter/Timer and Parallel I/O Unit	3-4	14
3.3.4	8530 SCC Serial Communications Controller	3-6	16

CHAPTER 4 – HOST MODE AND SPECIAL APPLICATIONS

Paragraph		Page	
4.1	Introduction to Host Mode	4-1	18
4.1.1	Why Do We Need a Host Mode	4-1	18
4.1.2	How does Host Mode Help Us?	4-1	18
4.1.3	Entering Host Mode	4-2	19
4.1.4	Leaving Host Mode	4-2	19
4.1.5	The Host Mode Dialog	4-2	19
4.1.6	Host Mode Recovery	4-3	20
4.2	Host Computer Commands	4-3	20
4.2.1	Unsupported Commands	4-4	21
4.2.2	Host Mode Mnemonic Indicators	4-4	21
4.2.3	CONNECT and DISCONNECT	4-5	22
4.2.4	ON/OFF Booleans or Switches	4-5	22
4.2.5	TXDELAY	4-5	22
4.2.6	SENDPAC	4-5	22
4.2.7	Interrogation or Query Commands	4-5	22
4.3	PK-232 Responses	4-5	22
4.3.1	Responses to Interrogation or Query Commands	4-6	23
4.3.2	OPMODE Response	4-7	24
4.3.3	Link Status Request Response	4-7	24
4.3.4	Call Sign Formats	4-7	24
4.4	Sending Data to the PK-232	4-8	25
4.4.1	Data Polling	4-8	25
4.4.2	The HPOLL Command	4-8	25
4.4.3	Special Case in AMTOR	4-9	26
4.4.4	Link Messages	4-9	26
4.5	Host Mode and Special Packet Applications	4-9	26
4.6	Raw HDLC	4-9	26
4.7	"KISS" TNC Asynchronous Packet Protocol	4-10	27
4.7.1	Starting "KISS" TNC Operation	4-11	28
4.7.2	"KISS" TNC Special Characters	4-11	28
4.7.3	"KISS" TNC Frame Structure	4-11	28
4.7.4	"KISS" TNC Commends	4-11	28
4.7.4.1	TXDELAY: CTL = $01	4-11	28
4.7.4.2	PERSISTENCE: CTL = $02	4-12	29
4.7.4.3	SLOTTIME: CTL = $03	4-12	29
4.7.4.4	TXTAIL: CTL = $04	4-12	29
4.7.4.5	FULLDUP: CTL = $05	4-13	30
4.7.4.6	HOST OFF: CTL = $FF	4-13	30
4.7.4.7	DATA: CTL = $00	4-13	30
4.8	Maximum Block Size	4-14	31
4.9	MEMORY, I/O and ADDRESS Commands	4-14	31
4.9.1	MEMORY Command	4-14	31
4.9.2	ADDRESS Command	4-14	31
4.9.3	I/O Command	4-15	32
4.10	Converse and Transparent Modes	4-15	32
4.11	MHEARD Command in Host Mode	4-15	32
4.12	Software Release Date Code	4-16	33
4.13	Product Type Code	4-16	33

CHAPTER 5 – MECHANICAL ASSEMBLY AND DIS ASSEMBLY

Paragraph		Page	
5.1	Cover Removal	5-1	34
5.2	Circuit Board Removal	5-	
5.3	Circuit Board Assembly	5-2	35
5.4	Cover Assembly	5-	

CHAPTER 6 – ADJUSTMENTS

6.1	Preliminary Setup	6-1	36
6.2	Calibration Procedure	6-	
6.2.1	AFSK Generator (Transmit) Adjustments	6-2	37
6.2.2	Demodulator (Receive) Adjustments	6-3	38
6.3	Functional Tests	6-	

CHAPTER 7 – TROUBLESHOOTING

7.1	Introduction	7-1	40
7.2	General Tests	7-	
7.2.1	Power Supply	7-	
7.2.2	Obvious Problems	7-2	41
7.2.3	Assembly Problems	7-	
7.2.4	Cabling Problems	7-	
7.3	Specific Symptoms	7-	
7.3.1	Symptom: Controller appears dead	7-	
7.3.2	Symptom: Modem cannot be calibrated	7-4	43
7.3.3	Symptom: Transmitter cannot be keyed	7-	
7.3.4	Symptom: Transmitter signals not copyable by other stations	7-	
7.3.5	Symptom: Received signals not copyable	7-	
7.4	Terminal Interface Troubleshooting	7-5	44
7.4.1	Symptom: Controller does not communicate with the terminal	7-	
7.4.2	Symptom: Controller signs on with mutilated data	7-	
7.4.3	Symptom: Controller does not respond or accept commands	7-6	45

APPENDIX A – AX.25 LEVEL 2 PROTOCOL	A	46
APPENDIX B – "KISS" TNC SPECIFICATION	B	70
APPENDIX C – DRAWINGS	C	75
APPENDIX D – WAVEFFORMS	D	83
	Last Page	101

CHAPTER 1 – INTRODUCTION

1.1 Introduction

This Technical Reference Manual will assist you to maintain, repair and adjust your PK-232 Data Controller. Please use this manual in conjunction with AEA's Operating Manual for the PK-232, Revision D.

The PK-232 Operating Manual contains complete and detailed information on controller operating procedures, controller commands and syntax, general installation methods, terminal and radio connections, as well as connector wiring diagrams, printed circuit board schematics, board layout drawings and parts list.

1.2 Scope

This manual includes information on theory of operation, hardware descriptions and troubleshooting instructions and charts. In addition, detailed information on the PK-232's Host mode is presented for the benefit of programmers and software application developers.

1.3 General Description

AEA's Model PK-232 is a multi-mode protocol converter and data controller that includes self-contained modems for all modes.

The PK-232 sends and receives Morse Baudot and ASCII RTTY, facsimile, AMTOR/SITOR and AX.25 packet. The PK-232 converts these signals to ASCII data end sends the data to your terminal via an EIA standard RS-232 (CCITT V.24/V.28) serial port, and to your printer via a special Centronics-type parallel interface. Except for the printer interface, the reverse procedures convert ASCII data typed at your terminal to the signals and protocols required for transmission to other stations.

All necessary tone generation and demodulation (modem) functions are built in; however, external modems can be connected to the PK-232 via dedicated rear-panel connectors. All decoding, encoding, protocol and transmitter control routines are stored in the PK-232's firmware in PROM and internal program memory.

Operating modes, speeds, modem tone pairs, filter bandwidths and system protocols are selected by typing commands on your computer or data terminal. Any communications or terminal emulator normally used with a telephone line modem can be used with the PK-232.

1.4 Specifications

As part of its program of product improvement, AEA reserves the right to make changes in this product's specifications. Changes will be made to the information in this document and incorporated in revisions to this manual. Specifications are subject to change without notice.

1.4.1 Operating Modes

The PK-232 provides operation in Morse, Baudot, ASCII, AMTOR/SITOR, half- or full-duplex Packet Radio in accordance with AX.25 protocols, facsimile, SIAM and Host and "KISS" TNC modes for use with packet protocols other than AX.25.

1.4.2 Modem Characteristics

Demodulator:	Limiter-discriminator type, preceded by an eight-pole Chebyshev 0.5 dB ripple bandpass filter
Receive bandpass:	Automatically switched by operating mode
VHF packet:	Center frequency 1700 Hz, bandwidth 2600 Hz
HF (except CW):	Center frequency 2210 Hz, bandwidth 450 Hz
CW:	Center frequency 800 Hz, bandwidth 200 Hz
Modulator:	Low-distortion AFSK sine wave function generator, phase-continuous AFSK
Output Level:	5 to 100 millivolts RMS, adjustable by rear-panel control

1.4.3 Processor System

Protocol conversion:	Zilog Z-80 microprocessor
RAM:	16 kilobytes
ROM:	Up to 48 kilobytes of ROM may be used
Hardware HDLC:	Zilog 8530 SCC

1.4.4 Input/Output Connections

Radio Interface:	Two five-pin TTL connectors, selectable on the front panel
Input/output Lines:	Receive audio Transmit audio Push-To-Talk (PTT) External squelch input Ground
External modem connector	Five-pin TTL - TXD, RXD, DCD, PTT, Ground
Direct FSK Outputs	Normal and reverse
Oscilloscope Outputs	Mark (Stop) and Space (Start)
CW keying Outputs	Positive: +100 VDC max. at up to 100 mA Negative: -30 VDC max. at up to 20 mA
Terminal Interface:	RS-232C 25-pin DB25 connector
Input/Output	RS-232 with full hardware and software handshake on wires 1-8 and 20
Terminal Data Rates	Auto-baud selection of 300, 1200, 2400, 4800 and 9600 BPS. TBAUD adds 110, 150, 200 and 600 BPS.

1.4.5 Controls and Indicators

Front Panel Controls:	Power Switch Radio Selector Switch Threshold Adjust
Indicators:	Ten-segment discriminator-type bargraph indicator for HF tuning. DCD LED (**D**ata **C**arrier **D**etect)

Status and Mode Indicators:

Mode Group	Status Group
BAUDOT	STBY
ASCII	PHASE
PKT	IDLE
MORSE	ERROR/CONV
CHECK	OVER
FEC	TFC/TRANS
ARQ	RQ/CMD
MODE L	CON
STBY	STA
	MULT
	SEND

1.4.6 General

Power Requirements:	+13 VDC (12 to 16 VDC) at 700 mA
Mechanical:	Overall 11" × 8.25" × 2.5" (279.4 × 209.6 × 63.5 mm) Weight 3 pounds (1.36 kilograms)

CHAPTER 2 – FUNCTIONAL DESCRIPTION

2.1 Major Sections

The PK-232 Data Controller has five major functional blocks:

- Analog
- Digital
- Input/Output (I/O)
- Display
- Power Distribution

Each of these functional blocks has its own distinct functions. The following paragraphs describe each section.

Refer to the PK-232 Functional Block Diagram, Figure 1 in Appendix C, while reading these descriptions.

2.1.1 Analog Section

The analog section consists of active filters, limiters, a threshold detector, DCD (Data Carrier Detect and diode discriminator circuits. Analog switches automatically adjust various circuit characteristics for each operating mode.

2.1.2 Digital Section

The digital section contains a microprocessor, read only memory (ROM) random access memory (RAM), two crystal-controlled clock generators an a variety of "glue" chips to gate and isolate the digital signals.

2.1.3 Input/Output Section

The I/O Section provides a standard serial data path in accordance with EIA RS-232-C and CCITT Recommendations V.24/V.28, between the PK-232 and the associated computer or terminal.

The I/O section contains an HDLC (High-Level Data Link Control) translator; output drivers for AFSK tones, DC logic voltages for direct FSK transmitter keying, CW DC keying signals, a watchdog timer and PTT (push-to-talk) switching voltages for the associated transmitter.

The I/O section also isolates input and output control signal buffers and provides circuits for connecting an external modem.

All control and communications to and from the PK-232 pass through the I/O section.

2.1.4 Display Section

The display section consists of a tuning indicator and its associated driver; status and mode indicators with their drivers and receivers; a DCD indicator and a threshold control.

2.1.5 Power Distribution Section

The Power distribution section contains voltage regulators, DC-to-DC converters for the PK-232's operating system and a battery system for backing up maintaining the data stored in RAM.

CHAPTER 3 – THEORY OF OPERATION

3.1 System Diagrams

Please refer to APPENDIX C for the various diagrams used as reference in this chapter.

3.1.1 Block Diagrams

Refer to the Functional Block Diagram, Figure 1 in APPENDIX C. Dashed lines separate the five sections of the PK-232. Each section will be described separately and is labeled as follows:

- Analog
- Digital
- Input/Output (I/O)
- Display
- Power Distribution

3.1.2 Logic Diagrams

Refer to the Logic Diagram, Figure 2 in APPENDIX C. Figure 2 shows the digital logic portion of the PK-232 which includes:

- the microprocessor
- Read Only Memory (ROM)
- Random Access Memory (RAM)
- clock and control logic
- part of the display logic
- the logic associated with the digital communications between the PK-232 and the terminal.

The major portion of the power distribution section is also shown on this drawing.

3.1.3 Schematic Diagrams

Refer to the Schematic Diagram, Figure 3 in APPENDIX C. Figure 3 shows the analog portion of the PK-232, which includes:

- mode-dependant bandpass filters
- MARK and SPACE resonators
- audio frequency discriminator
- mode-dependant low pass filter
- comparator circuitry
- AFSK generator
- watchdog time
- transmitter audio driver
- keying circuitry
- radio switching
- tuning indicator circuitry

3.2 Analog Subsystem

Each of the PK-232's operating modes requires different bandwidths, as well as varying AFSK tone and speed characteristics. The analog circuitry is adapted to these differences by using program-controlled switches to place precision resistors in parallel with the fixed components of the circuits. In this manner, under software command, the analog circuits are optimized for the mode selected.

3.2.1 Receive Function

Audio signals from the selected RADIO 1 or RADIO 2 input pass through a rolloff filter to compensate for VHF receiver characteristics and are then applied to a buffer amplifier to limit audio signal amplitude.

The signal then pass through a bandpass filter whose center frequency tuning and bandwidth are controlled by the OPMODE selection for optimum characteristics.

The bandwidth-limited signals pass through an amplitude limiter to remove variations in audio levels. Mark and Space tones are separated in the Mark and Space resonators. The separated Mark and Space output signals are rectified in a diode discriminator circuit.

The detected data is filtered in an active lowpass filter and compared with a reference voltage in a slicer circuit to digitize the data. The resulting data is then shaped and inverted for passing to the internal modem circuitry.

When a full word of data (eight bits) has been assembled in the modem circuitry, an interrupt is generated to the microprocessor which then processes the data.

3.2.1.1 Receive Circuits

The received audio signals are routed through one of two radio connection receptacles, J4 (RADIO 1) or J6 (RADIO 2), or through the respective paralleled radio input cable jacks, J3 or J5. The radio is selected by SW2, the RADIO 1/RADIO 2 push-button switch on the front panel. Radio input and output connections are bypassed for RF by C64 and C63.

The audio input consists of resistor-capacitor combination R34 and C54 to compensate for the audio characteristics of the average VHF FM radios by emphasizing higher-frequency tones and attenuating lower-frequency tones.

Input buffer amplifier U28-D limits received audio signal levels to prevent overloading multiple-resonator feedback-type 0.5-dB-ripple Chebyshev active bandpass filter formed by operational amplifiers U23 and U26.

Analog gates (FET switches) U22, U24, U25, U26 and U27 select filter center frequency and bandwidth. Gate selection is a function of the operating mode.

Bandpass-filtered output signals are amplified and limited by U28-A to remove amplitude variations and passed to MARK and SPACE resonators U30 and U32. Gate U29 determines the tone-pair resonator tuning.

For CW signals, U31 transfers the bandpass filter output directly to the MARK channel discriminator. The SPACE channel is decoupled from the circuit and ignored.

The tones processed by the resonators are rectified in a full-wave diode network formed by D19 and D20 for the MARK channel, and D22 and D23 for the SPACE channel. The MARK signal is positive with respect to the voltage reference (VR); the SPACE signal is negative.

Output signals from both channels are envelope-detected simultaneously by a short and long time-constant network and summed at the input of U32-B.

After passing through a lowpass filter to remove ripple, the signal is amplified by U32-C, U34-C and U34-D and then squared. Its logic level is shifted by a TTL inverter U15-C for use by either an external modem or U6, a Zilog 8536 Counter/Timer and Parallel I/O Device.

3.2.2 Transmit Function

Digital input signals from the terminal are received by the serial controller and passed to the microprocessor for action. If ECHO has been commended ON, the input signals are routed by the serial controller back to the terminal for display.

3.2.2.1 Transmit Circuits

Data to be transmitted is received over the RS-232 serial port and read by Z8530 Serial Communications Controller U7 under the control of the Z80 microprocessor. The data is periodically sent over the data bus to the Z8536, U6. The Z8536 translates the data into a binary string at the correct data rate and routes the binary string to AFSK generator U40.

The AFSK generator supplies one of two tones to selected RADIO connector J4 or J6, depending on the position of RADIO 1/Radio 2 switch SW2.

The transmit tone pair produced by the tone generator is selected by FET switch U35. This switch selects appropriate frequency-determining resistors R164, R165, R157 and R168, in accordance with the selected operating mode.

In addition, a logic level of five volts DC (5 VDC) or ground is presented at Scope/FSK receptacle J7, either inverted or normal, for FSK transmitter keying.

At the same time that the generator is keyed, a Push-to-Talk (PTT) signal is supplied to transistor keyers (Q4 and Q5). The polarity of the keying signal is selected by jumpers JP2 and JP3 for the two radio receptacles.

This PTT signal is present as long as pulses from the timer in U7 refresh the timeout circuitry of Q10 and Q3. If for any reason the program fails or becomes disabled, the timeout circuitry prevents the PTT line from being activated.

3.3 Digital Subsystem

In the following discussion of the Digital Subsystem, the dollar sign ($) indicates hexadecimal numbers.

3.3.1 Z80A Central Processing Unit

The processor, U1, is a Z80A operating at 4.0 MHz. The peripheral chips U6 and U7 cause system interrupts using the CPU's INT* input (pin 16). The NMI* and BUSRQ* inputs are unused, as are the RFSH, HALT and BUSAK outputs.

3.3.2 Memory

The PK-232 memory consists of 48 kB of EPROM at U2 and U3, and 16 kB of static RAM at U4 and U5.

U2 is a 27256 EPROM which occupies the 32 kB address space between $0000 and $7FFF.

EPROM U3 is a 27128 occupying 16 kB from $8000 to $BFFF.

The read-write memory consists of two 6264 8 kB volatile static RAMs, U4 occupying $C000-DFFF and U5 at $E000-FFFF. Both U4 and U5 have their power backed up by batteries when the system power is off.

The 74LS139 address decoder at U10 receives address bits A13, A14 and A15 from the CPU, and provides chip enables for U3, U4 and U5. Address bit A15 is used directly as the chip enable for the U2 EPROM at address $0000.

U17 pin 8 provides a Memory Read (OE*) line for U2, U3, U4 and U5. U17 pin 11 is the Memory Write (·WE*) line for the RAMs U4 and U5.

3.3.3 8536 CIO Counter/Timer and Parallel I/O Unit

U6 is an 8536 CIO chip which provides 20 parallel I/O pins and three timers. This chip receives address bits A0, A1 and A6 from the CPU. The PK-232's firmware addresses the 8536 by forcing all other address bits high, yielding these addresses:

$BC - Parallel port A data
$BD - Parallel port B data
$BE - Parallel port C data
$BF - Control port

The 8536 uses the CPU's 4.0 MHz clock on the PCLK input (pin 16). The 8536 interrupt has priority over the interrupt from the 8530 (U7).

Parallel port A controls several different functions in the PK-232:

Bit	U6 Pin	Function	Sense	Direction
PA7	26	Squelch		In
PA6	27	RTTY transmit data		Out
PA5	28	Wide shift enable	Pos	Out
PA4	29	Tone enable	Pos	Out
PA3	30	Calibration		In
PA2	31	FSK enable	Pos	Out
PA1	32	CW key closed	Neg	Out
PA0	33	Narrow shift enable	Neg	Out

PA3 counts the transition on the digital output of U40, the 2206 tone generator.

PA2 when high enables FSK for Packet and RTTY; when low PA2 enables CW. PA5, PA2 and PA0 work together to determine the shift the PK-232 uses:

PA5	PA2	PA0	
0	1	0	Narrow shift FSK
1	1	1	Wide shift FSK
0	0	1	CW

Parallel port A can be controlled by the user directly. Set ADDRESS to $BF0D and then read or write port A using the IO command.

Parallel port B pins PB4, PB5, PB6 and PB7 drive the Mode LEDs through the 7445 decoder at U11:

PB7	PB6	PB5	PB4	LED
0	0	0	0	PKT
0	0	0	1	MORSE
0	0	1	0	ASCII
0	0	1	1	BAUDOT
0	1	0	0	CHECK
0	1	0	1	FEC
0	1	1	0	MODE L
0	1	1	1	ARQ
1	0	0	0	STBY
1	0	0	1	(FAX)
1	0	1	0	(SIGNAL or SIAM)

PB3 drives the SEND LED through one of the 7406 drivers at U18. Parallel port B pins PB0, PB1 and PB2 drive the Status LED through the 7445 decoder at U12:

PB2	PB1	PB0	LED	
0	0	0	STBY	
0	0	1	PHASE	
0	1	0	OVER	
0	1	1	IDLE	
1	0	0	TFC	TRANS
1	0	1	ERROR	CONV
1	1	0	RQ	CMD

Parallel Port C drive their functions through parts of the 7406 driver chip at U18:

PC0	STA LED
PC1	CON LED
PC2	MULT LED
PC3	Bar graph enable

When parallel ports B and C are used with the parallel printer, the pins are brought out from the 8536 unbuffered to the J2 connector:

Bit	U6 Pin	J2 Pin	Function	Sense	Direction
PB7	15	21	Data 8	Pos	Out
PB6	14	19	Data 7	Pos	Out
PB5	13	18	Data 6	Pos	Out
PB4	12	17	Data 5	Pos	Out
PB3	11	16	Data 4	Pos	Out
PB2	10	15	Data 3	Pos	Out
PB1	9	14	Data 2	Pos	Out
PB0	8	13	Data 1	Pos	Out
PC2	21	24	Busy	Pos	In
PC1	20	23	Ack	Neg	In
PC0	19	22	Strobe	Neg	Out

The Busy and ACK signals should each contain a 200-Ohm series resistor in the cable to the parallel printer.

3.3.4 8530 SCC Serial Communications Controller

U7 is an 8530 SCC chip which provides two serial communications ports.

Port A is the HDLC port for Packet-Radio.
Port B is the RS-232 interface to the terminal.

This chip receives address bits A0, A1 and A7 from the CPU. The PK-232 firmware addresses the 8530 by forcing all other address bits high, yielding these addresses:

$7C - Terminal (Port B) control
$7D - Terminal (Port B) data
$7E - HDLC (Port A) control
$7F - HDLC (Port A) data

The 8530 chip has a 2.4576 MHz clock that is totally independent of the rest of the system.

The serial interface signals TXD (J2 pin 2), RTS (4) and DTR (20) flow from the RS-232 connector through the drivers on the 1489 chip at U19 to the 8530's RXDB, CTSB and DCDB pins respectively.

The 8530 output pins TXDB, RTSB and DTRB pins are buffered by the drivers on the 1488 chip at U20 to the J2 pins RXD (3), CTS (5) and DCD (8) respectively.

Both the RTS (J2 pin 4) and DTR (pin 20) signals must be high for the 8530 to send data on the RS-232 link.

J2 pin 8 (DCD) may be used as an active-high Packet Connect indicator for mailbox software. J2 pin 6 (DSR) is permanently pulled high. J2 pin 1 is frame ground and J2 pin 7 is signal ground. All other J2 pins are reserved for use with a parallel printer.

The SYNCB input (pin 29) is pulled high by R179. This input may be connected to a hardware EAS (Echo as Sent) switch to monitor the output in AMTOR, Morse, Baudot and ASCII modes.

Forcing this pin low has the same effect as typing the command EAS ON.

Port A interfaces with the modem. The RXDA and CTSA inputs are tied together so that the incoming data can be read either by the 8530's internal logic (for Packet mode) or by the firmware directly on a bit-by-bit basis (for AMTOR, Baudot, ASCII and Morse modes).

Similarly, the TXDA output is tied to the 8536 PA6 output so that the outgoing data can be controlled either by the 8530 or the firmware.

In HDLC operation for Packet mode reception, an internal phase-locked loop is driven by a clock that is 32 times the Packet baud rate.

The transmitter is driven by a clock right as the Packet baud rate. The divide-by-32 function is performed by the 74LS393 at U8.

The U7 TRXCA output (pin 14) is the 32× clock, which may be used with an external modem. This signal is connected to an input on U8, goes through 5 stages of divide-by-2 and is brought out out to the U7 RTXCA input (pin 12) as the 1× clock.

The RTSA output (pin 17) is the PTT signal, active low. For the PTT to remain active for any length of time, the DTRA output (pin 16) must furnish a constantly oscillating signal. This signal goes to the watchdog timer circuit.

Should the oscillating signal stop, the PTT will go inactive after a period determined by the time constant of R146 (10 k) and C55 (47 µ).

For this scheme to work properly, the DTRA output is toggled every pass trough the main loop of the PK-232 firmware.

In order to make the 8530 and 8536 chips operate with the Z80A CPU, a state machine consisting of U9 (74LS164), U14 (74LS11) and U1 (74LS04) is used.

This circuit provides a Wait signal to the CPU and I/O Request and Interrupt Acknowledge signals to the 8530 and 8536 at proper times, whenever the CPU begins an Interrupt Acknowledge cycle.

For more details, refer to the Z8036/Z8530 SCC Serial Communications Controller Technical Manual and the Z8036 Z-CIO/Z8536 CIO Counter/Timer and Parallel I/O Unit technical Manual, both available from Zilog, Inc.

CHAPTER 4 – HOST MODE AND SPECIAL APPLICATIONS

4.1 Introduction to Host Mode

In conventional AX.25 Packet operation, the PK-232 presents a reasonably 'user-friendly' verbose human interface that uses plain-language command words up to eight-letters that actually spell out a word.

The CONNECT command, for example, describes exactly what the command does; the PK-232 transmits a Connect Request (SABM) frame. In addition, when you want to know the connect path and other connection characteristics, you type CONNECT and the PK-232 sends your computer the following:

```
Link state is: CONNECTED to N6IA via W6AMT; v2; 1 unACKed
```

all of which tells you that:

- o you are connected to N6IA through the W6AMT digipeater;
- o you have AX25L2V2 set to ON;
- o you have sent a frame to N6IA which he has not acknowledged yet.

4.1.1 Why Do We Need a Host Mode ?

You may wish to write a program that permits your computer to control your PK-232. You could write a program that provides a split screen with status windows. You might want to include a personal mailbox, a multi-user bulletin board system, an automatic calling routine, with tutorials or help screens. you may also wish to experiment with some high-level protocols or techniques other than AX.25 Level 2. Your program could be written to take much of the burden of operating a data controller and free you from the need to memorize the PK-232 commands and possibly permit fully-automatic system operation.

When working with conventional, human-oriented data controllers and Packet TNCs, your hypothetical program must interpret and translate all the verbose, human-interface information sent in either direction over the serial interface between your computer and your PK-232. All PK-232's command and response features that permit convenient human operating unfortunately produce substantive difficulties for a computer. For example, your computer and program must:

- o decide if the PK-232 is in command mode or converse mode;
- o sort through status messages in English;
- o accept any data or status from the PK-232 at any time.

4.1.2 How Does Host Mode Help Us ?

Host mode does not use human-type dialog. By communication directly with the "host" or computer, Host mode provides the computer with much greater direct control over the PK-232. Host mode permits programmers to eliminate, reduce or greatly simplify the transfer and subsequent encoding and decoding of critical information, eliminating wasteful and redundant information. In Host mode, the PK-232 is 'unfriendly'; humans would find it difficult to operate the PK-232 in Host mode.

Host mode uses the normal RS-232 serial link between the PK-232 and the computer, but provides a more efficient link with fewer characters necessary.

In Host mode, the PK-232 sends data to the computer only when the computer requests data. This feature is selectable by the user (see HPOLL below).

4.1.3 Entering Host Mode

Enter Host mode by typing the following commands:

Type: AWLEN 8<CR>
PARITY 0<CR>
8BITCONV ON<CR>
RESTART<CR>

Verify that your computer's serial interface driver is also set to eight bits, no parity, and that you have set your system for hardware control by disabling XON/XOFF and enabling RTS/CTS/DTR.

Type: HOST ON<CR>

If you wish to apply power to your computer and Host mode immediately, be aware that power transients may cause random data to be sent to the PK-232.

- To ensure correct entry into Host mode, send XON, CANLINE and COMMAND characters first, as shown below with the hexadecimal representation of each character listed above each its acronym or literal. Note: 'sp' is the 'space bar' character.

$11	$18	$03	$48	$4F	$53	$54	$20	$59	$0D
XON	CAN	COM	H	O	S	T	sp	Y	CR

Follow these steps:

1. Send $01 [^A] (^ = CTRL)
2. Send the sequence: $01 $4F $47 $47 $17
3. If you receive the response: $01 $4F $47 $47 $00 $17

you have successfully entered the Host mode. If you receive some other response, got to Step 2.

4.1.4 Leaving Host Mode

To leave the Host mode and return to conventional or verbose mode, type:

	$01	$4F	$48	$4F	$4E	$17	
or	^A	O	H	O	N	^W	(^ = CTRL)

or transmit a BREAK signal (reversal to SPACE or START polarity of at least 300 milliseconds) on the serial link.

4.1.5 The Host Mode Dialog

The host computer and PK-232 communicate in blocks of characters:

- Each block starts with the SOH (Start of Header) character, which is $01 or ^A.
- The next character is a special data-type identification byte, called the CTL (control) character that indicates whether the following characters are a command, status or data.

- o Next come the literal command or data characters.
- o Each block end with $17 or ^W, the ETB (End of Transmission Block) character.

Therefore, in Host mode, each block appears as follows:

[SOH] [CTL] [data] [ETB]

In addition, DLE (Data Link Escape), $10 may be used as a pass character.

- o If one of the data field characters is SOH ($01), DLE ($10) or ETB ($17), the pass character DLE ($10) is inserted before the data character.

For example, data consisting of the bytes $04, $01, $05, $10, $12 is sent to channel 0 in this block:

$01	$20	$04	$10	$01	$05	$10	$10	$12	$17
SOH	CTL	data	DLE	data	data	DLE	data	data	ETB

This occurs in either direction, from the computer to the PK-232 or from the PK-232 to the computer.

4.1.6 Host Mode Recovery

If it becomes necessary to quit Host mode because of a problem on the RS-232 link, send any command, with TWO SOH characters to start the block. This makes sure the PK-232 goes to a known state where it is processing commands. A suggestion:

$01	$01	$4F	$47	$47	$17
SOH	SOH	O	G	G	ETB

4.2 Host Computer Commands

The host computer send blocks to the PK-232 using the CTL bytes:

$2x: Data to channel x, where x = 0-9
$4x: Command to channel x
$4F: Command, no change to input channel

The computer sends commands to the PK-232 in the following format:

SOH $4F *a* *b* (any data) ETB

where *a* and *b* are the two character Host mode mnemonic or abbreviation for that command.

- o When setting a parameter, <u>do not send</u> the SPACE character between the command and the first argument.
- o Arguments are entered just as in the verbose mode, with space characters or commas between arguments.
- o The command ends with ETB, without a carriage return before it.

4.2.1 Unsupported Commands

The PK-232 does not accept these commands in Hoste Mode:

CALIBRATE – CONVERSE – CSTATUS – DISPLAY – HELP – TRANS

4.2.2 Host Mode Mnemonic Indicators

Each command in the Host mode can be sent by issuing a unique mnemonic or abbreviation in the form of a two-letter character group. These mnemonics are shown in the following list before each command.

8B	**8BITCONV**	**AU**	**AAB**	**AB**	**ABAUD**	**AG**	**ACHG**	**AA**	**ACRDISP**
AK	**ACRPACK**	**AT**	**ACRRTTY**	**AE**	**ADDRESS**	**AD**	**ADELAY**	**AI**	**ALFDISP**
AP	**ALFPACK**	**AR**	**ALFRTTY**	**AL**	**ALIST**	**AM**	**AMOTR**	**AC**	**ARQ**
AO	**ARQTMO**	**AS**	**ASCII**	**AY**	**ASPECT**	**AW**	**AWLEN**	**AV**	**AX25L2V2**
AX	**AXDELAY**	**AH**	**AXHANG**	**BA**	**BAUDOT**	**BE**	**BEACON**	**BI**	**BITINV**
BK	**BKONDEL**	**BT**	**BTEXT**	**CL**	**CANLINE**	**CP**	**CANPAC**	**CX**	**CASEDISP**
CU	**CBELL**	**CC**	**CCITT**	**CF**	**CFROM**	**CB**	**CHCALL**	**CD**	**CHDOUBLE**
CH	**CHSWITCH**	**CK**	**CHECK**	**CQ**	**CMDTIME**	**CM**	**CMSG**	**CI**	**CODE**
CN	**COMMAND**	**CE**	**CONMODE**	**CO**	**CONNECT**	**CY**	**CONPERM**	**CG**	**CONSTAMP**
CI	**CPACTIME**	**CR**	**CRADD**	**CT**	**CTEXT**	**CW**	**CWID**	**DS**	**DAYSTAMP**
DA	**DAYTIME**	**DC**	**DCDCONN**	**DL**	**DELETE**	**DF**	**DFROM**	**DI**	**DISCONNE**
DW	**DWAIT**	**EA**	**EAS**	**EC**	**ECHO**	**ES**	**ESCAPE**	**FA**	**FAX**
FN	**FAXNEG**	**FE**	**FEC**	**FL**	**FLOW**	**FR**	**FRACK**	**FS**	**FSPEED**
FU	**FULLDUP**	**GR**	**GRAPHICS**	**HB**	**HBAUD**	**HD**	**HEADERLN**	**HI**	**HID**
HO	**HOST**	**HP**	**HPOLL**	**ID**	**ID**	**IL**	**ILFPAK**	**IO**	**IO**
JU	**JUSTIFY**	**KI**	**KISS**	**LR**	**LEFTRITE**	**LO**	**LOCK**	**MX**	**MAXFRAME**
MB	**MBX**	**MC**	**MCON**	**MD**	**MDIGI**	**MM**	**MEMORY**	**MI**	**MFILTER**
MF	**MFROM**	**MH**	**MHEARD**	**MN**	**MONITOR**	**MO**	**MORSE**	**MP**	**MSPEED**
MR	**MRPT**	**MS**	**MSTAMP**	**MT**	**MTO**	**MA**	**MYALIAS**	**ML**	**MYCALL**
MG	**MYSELCAL**	**MK**	**MYALTCAL**	**NE**	**NEWMODE**	**NO**	**NOMODE**	**NR**	**NUCR**
NF	**NULF**	**NU**	**NULLS**	**OK**	**OK**	**OP**	**OPMODE**	**PA**	**PACKET**
PL	**PACLEN**	**PT**	**PACTIME**	**PR**	**PARITY**	**PS**	**PASS**	**PX**	**PASSALL**
PE	**PERSIST**	**PP**	**PPERSIST**	**PC**	**PRCON**	**PF**	**PRFAX**	**PO**	**PROUT**
PY	**PRTYPE**	**RW**	**RAWHDLC**	**RB**	**RBAUD**	**RC**	**REVE**	**RE**	**RECEIVE**
RX	**RXREV**	**RD**	**REDISPLA**	**RL**	**RELINK**	**RS**	**RESET**	**RP**	**RESPTIME**
RT	**RESTART**	**RY**	**RETRY**	**RF**	**RFEC**	**SE**	**SELFEC**	**SP**	**SENDPAC**
SI	**SIGNAL**	**SL**	**SLOTIME**	**SQ**	**SQELCH**	**SR**	**SRXALL**	**ST**	**START**
SO	**STOP**	**TB**	**TBAUD**	**TC**	**TCLEAR**	**TM**	**TIME**	**TR**	**TRACE**
TW	**TRFLOW**	**TI**	**TRIES**	**TD**	**TXDELAY**	**TF**	**TXFLOW**	**TX**	**TXREV**
UN	**UNPROTO**	**UR**	**USERS**	**US**	**USOS**	**VH**	**VHF**	**WI**	**WIDESHFT**
WO	**WORDOUT**	**WR**	**WRU**	**XW**	**XFLOW**	**XM**	**XMIT**	**XO**	**XMITOK**
XF	**XOFF**	**XN**	**XON**						

For example, whereas in verbose mode, the human operator types:

MFILTER 7, 19

in Hoste mode the computer sends the same command as:

$01	$4F	$4D	$49	$37	$2C	$31	$39	$17
SOH	CTL	M	I	7	,	1	9	ETB

4.2.3 CONNECT and DISCONNECT

Send the CONNECT (CO) and DISCONNECT (DI) commands with:

CTL byte = $4*x*, where *x* is the channel 0-9.

4.2.4 ON/OFF Booleans or Switches

To set ON/OFF switch, the only argument is an ASCII Y or N for YES or NO respectively.

o Y or N is returned in response to a query command.

For example, to set MRPT ON, send:

SOH	$4F	M	R	Y	ETB

o To set a numerical value, the value must be in ASCII, not binary.

4.2.5 TXDELAY

To set TXDELAY to 40, send:

$01	$4F	$54	$44	$34	$30	$17
SOH	CTL	T	D	4	0	ETB

4.2.6 SENDPAC

To set SENDPAC to $0D, send:

$01	$4F	$53	$50	$24	$30	$44	$17
SOH	CTL	S	P	$	0	D	ETB

4.2.7 Interrogation or Querry Commands

Querry commands are similar to the human or verbose mode in that the two-letter command to the PK-232 is <u>not</u> followed by arguments, that is, the command is followed by ETB.

4.3 PK-232 Responses

The PK-232 always issues a response to each command.

o The computer should always wait for the response before issuing another command.

The PK-232 sends blocks to the computer using these CTL bytes:

$2F: Echoed data in Morse, Baudot ASCII and AMTOR,
$3x: Data from channel x.
$3F: Monitored frames.
$4x: Link status from channel x (response to CONNECT command).
$4F: Response to command.
$5x: Link messages from channel x.
$5F: Status errors.

NOTE: Channel 'x' (0-9) refers to the Packet multi-connect channel that would be available in verbose (human) mode by using the CSWITCH character. Only channel 0 is used in all communications mode other than Packet mode.

The PK-232's response structure is:

SOH	$4F	*a*	*b*	*c*	ETB

where *a* and *b* are the two-letter host command received from the computer, and *c* is:

$00	acknowledge, no error
$01	bad
$02	too many
$03	not enough
$04	too long
$05	range
$06	callsign
$07	unknown command
$08	VIA
$09	not while connected
$0A	need MYCALL
$0B	need MYSELCAL
$0C	already connected
$0D	not while disconnected
$0E	different connects
$0F	too many packets outstanding
$10	clock not set
$11	need ALL/NONE/YES/NO
$15	not in this mode

Other errors are:	SOH	$5F	X	X	W	ETB	bad block
	SOH	$5F	X	X	Y	ETB	bad CTL in block

4.3.1 Responses to Interrogation or Querry Commands

Responses to query commands consist of the following block:

SOH	$4F	a	b	(value)	ETB

where the value is:

for a switch:	Y or N
for a BEACON or PACTIME:	E or A, space, and a number
for CONMODE:	C or T
for all other commands:	the same as in the verbose mode, except for CONNECT, which is discussed in this chapter.

For example, to read the state of MRPT, send:

SOH	$4F	M	R	ETB

The reply should be:

SOH	$4F	M	R	Y	ETB

4.3.2 OPMODE Response

The computer can use the OPMODE command to interrogate the PK-232 for information on its current operating mode. The PK-232 sends one of the following responses:

Packet:	**SOH**	**$4F**	**O**	**P**	**P**	**A**	**ETB**			
Morse:	**SOH**	**$4F**	**O**	**P**	**M**	**O**	***x***	***y***	***z***	**ETB**
Baudot:	**SOH**	**$4F**	**O**	**P**	**B**	**A**	***x***	**ETB**		
ASCII:	**SOH**	**$4F**	**O**	**P**	**A**	**S**	***x***	**ETB**		
AMTOR Standby	**SOH**	**$4F**	**O**	**P**	**A**	**M**	**$30**	***x***	**ETB**	
ARQ:	**SOH**	**$4F**	**O**	**P**	**A**	**C**	***v***	***x***	**ETB**	
ARQ Listen:	**SOH**	**$4F**	**O**	**P**	**A**	**L**	***v***	**R**	**ETB**	
FEC:	**SOH**	**$4F**	**O**	**P**	**F**	**E**	***v***	***x***	**ETB**	
SELFEC:	**SOH**	**$4F**	**O**	**P**	**S**	**E**	***v***	***x***	**ETB**	
FAX:	**SOH**	**$4F**	**O**	**P**	**F**	**A**	***v***	***x***	**ETB**	

where *v* = $30 if Standby
$31 if Phasing
$32 if Change-over
$33 if Idle
$34 if Traffic
$35 if Error
$36 if RQ
$37 if Sync

x = S if transmit
R if receive

yz = present receive Morse code speed in words per minute

4.3.3 Link Status Request Response

Use the CONNECT command to query the link state of channel x:

SOH $4x C O ETB

The PK-232 response is:

SOH $4x C O *a b c d e* path ETB

where *a, b, c, d* and *e* are values $0-$F, ORed with $30:

a = the link state -1, e.g., S05 is sent as $34
b = $31 if the link is AX.25 L2 version 2, $30 if pre-v2
c = the number of outstanding (unacknowledged) packets
d = the number of retries done at this time
e = $31 if CONPERMed, $30 otherwise
path = the callsign of the connects, followed by the digipeater callsigns

4.3.4 Callsign Formats

Examples of the callsign format are:

WA2DFI
W6CUS-1 via K6LLK, WD6CMU-1

4.4 Sending Data to the PK-232

Send data to channel 'x' (all modes except Packet assume channel 0) as follows:

SOH	$2x	(data)	ETB

This type of block does not produce an immediate acknowledgement as does a command block. The data acknowledgement from the PK-232 is sent as a result of a data poll (see below) and produces the following response when the PK-232 can process the data:

SOH	$5F	X	X	$00	ETB

If the PK-232 is busy, the response is delayed until the data can be processed. The host computer should wait for each data ack before sending more data.

4.4.1 Data Polling

The computer can poll the PK-232 for information such as channel data or status, link messages, data acknowledgements or monitored frames. To poll the PK-232, send the poll command as follows:

SOH	$4F	G	G	ETB

If the PK-232 has nothing to send to the computer, the response is:

SOH	$4F	G	G	$00	ETB

If the PK-232 has data to send and information is waiting to be sent, the response is shown below. Only one block is sent for each poll.

SOH	$2F	.....		ETB	echoed data
SOH	$3x	.....		ETB	data
SOH	$3F	.....		ETB	monitored data
SOH	$4x	.....		ETB	link status
SOH	$5x	.....		ETB	link messages
SOH	$5x	X X	.....	ETB	status errors
SOH	$5F	X X	$00	ETB	data acknowledgement

Here is an example of the Host mode dialog between the computer and the PK-232:

Computer:	SOH	$30	(data)	ETB		data sent
Computer:	SOH	$4F	G G	ETB		poll
PK-232:	SOH	$4F	G G	$00	ETB	nothing yet
Computer:	SOH	$4F	G G	ETB		poll
PK-232:	SOH	$5F	X X	$00		data acknowledged

4.4.2 The HPOLL Command

The HPOLL command determines whether the host computer must constantly poll the PK-232.

- o If HPOLL is ON, the host must poll for everything, as described above.
- o If HPOLL is OFF, the PK-232 sends all blocks to the host computer when they are formed and the data poll ($4F G G) is not needed.

In communications mode other than Packet, received data is formed into blocks one character at a time, for total of four bytes per block (SOH, CTL, data, ETB).

4.4.3 Special Case in AMTOR

AMTOR operation produces some special responses.

- o When the PK-232 is in Mode A (ARQ), data received from another station is block type $30.
- o When the PK-232 is in Mode B (FEC), S (SELFEC) or L (ALIST) block type $3F is used.

4.4.4 Link Messages

Link messages have the following format:

SOH $5x (message) ETB

Some of these link messages are:

CONNECTED to <callsign>
<callsign> busy
Connect request: <callsign>
FRMR sent: xx yy zz
FRMR rcvd: xx yy zz
Retry count exceeded
DISCONNECTED: <callsign>
LINK OUT OF ORDER, possible data loss
(3 bell characters, output from CBELL)
Transmit data remaining

4.5 Host Mode and Special Packet Applications

Certain differences exist when operating in Packet mode using the Host mode.

- o Data is packetized only after the ETB character.
- o PACTIME and CPACTIME are ignored.

The Host mode permits the use of special machine-oriented interfaces that are inappropriate for direct interpretation by human users.

4.6 Raw HDLC

Raw HDLC is available <u>only</u> in Host mode. When RAWHDLC is set to ON, data sent from the computer to the PK-232 is converted to pure Packet frames without adding headers or protocol bytes.

In RAW HDLC, the host computer <u>must</u> provide the AX.25 header with each outgoing frame. Raw HDLC moves the AX.25 protocol from the PK-232 to the host computer, permitting the system operator to create his own version of AX.25, with possible inclusion of layer 3 and higher-level protocols, as well as protocols other than AX.25.

Data from the PK-232 to the computer included every byte in the received frame, minus flags and checksum.

In Raw HDLC, data is sent from the computer to the PK-232 using a CTL byte of $20 (data to channel 0); received data from the PK-232 uses $3F (monitored data).

Very little error checking is done in the RAWHDLC mode. Do not send commands such as CONNECT and DISCONNECT that could possible create problems. Assume that all frames are being sent in the UNPROTO state.

Enter Raw HDLC from the human or verbose mode by typing the following commands and parameter values:

AWLEN 8
PARITY 0
RESTART
TRACE OFF
CONMODE TRANS
HID OFF
BEACON EVERY 0
PACKET
RAWHDLC ON
KISS OFF
HOST ON

From the Host mode, send the Host mode equivalents.

For details on implementing AX.25, see the "AX.25 Link Layer Protocol" document, available from the ARRL Publications Department foe $10.

4.7 "KISS" TNC Asynchronous Packet Protocol

The PK-232 provides a simple, asynchronous, computer-to-TNC protocol for a Raw HDLC or "KISS" TNC developed by Phil Karn, KA9Q. The computer must again provide all AX.25 headers and timing functions. KISS protocol is similar to the Raw HDLC protocol described earlier, with a few exception:

- The frame delimiter characters.
- The commands available.
- The method of determining the proper time to transmit.
- The KISS protocol does not use the Host mode [SOH CTL ... ETB] described earlier. Host mode commands and responses described before this KISS section do not apply when KISS is active.

4.7.1 Starting "KISS" TNC Operation

Before entering the KISS mode, set the PK-232 for hardware flow control on the host side of the RS-232 link.

Type the following parameter values from the human or verbose mode:

AWLEN	8
PARITY	0
RESTART	
CONMODE	TRANS
TRACE	OFF
HID	OFF
BEACON	EVERY 0
PACKET	
RAWHDLC	ON
HPOLL	OFF
PPERSIST	ON
KISS	ON
HOST	ON

4.7.2 "KISS" TNC Special Characters

The special "KISS" TNC characters are:

FEND	(frame End)	$C0
FESC	(Frame Escape)	$DB
TFEND	(Transposed Frame End)	$DC
TFESC	(Transposed Frame Escape)	$DD

4.7.3 "KISS" TNC Frame Structure

Each KISS frame or block of characters begins with a FEND (Frame End) character.

- o The next character is like the original PK-232 CTL character in that it defines whether the following characters are data or commands.
- o The literal data or parameter value follows the CTL character.
- o Each block ends with another FEND character.

The FESC (Frame Escape) character is the pass character.

- o If any data character is a FEND, that character is translated into the two-byte sequence FESC TFEND (Frame Escape, Transposed Frame End).
- o If any data character is a FESC, that character is replaced by the two-byte sequence FESC TFESC (Frame Escape, Transposed Frame Escape).
- o If any data character is a TFEND or TFESC, that character is not changed.

4.7.4 "KISS" TNC Commands

There are only six commands in the "KISS" TNC protocol.

4.7.4.1 TXDELAY: CTL = $01

TXDELAY in the "KISS" TNC protocol has the same meaning as traditional TNCs: it determines how long the TNC waits between activating the PTT line and the start of HDLC data.

The KISS command is:	$C0	$01	*n*	$C0
	FEND	CTL	TXD	FEND

where *n* is a binary number expressing TXDELAY in units of 10 milliseconds.

4.7.4.2 **Persistence: CTL = $02**

PERSISTENCE (P) and SLOTTIME work together to establish the time at which the PK-232 will transmit.

OERSICTENCE is best dexcribed by quoting Phil Karn, KA9Q, directly from his "Raw TNC Functional Spec".

"Whenever the host has queued data for transmission, the TNC begins monitoring the carrier detect signal from the modem. It waits indefinitely for this signal to go inactive. Once the channel is clear, the TNC generates a random number between 0 and 255. If this number is less than or equal to P, the TNC asserts the transmitter PTT line, waits .01 # TXDELAY seconds, and transmits all frames in its queue. The TNC then releases PTT and goes back to the idle state. If the random number is greater than P, the TNC delays .01 # SLOTTIME seconds and repeats the procedure. If the carrier detect signal has gone active in the meantime, the TNC again waits for it to clear before continuing. Note that P=255 means always transmit as soon as possible, regardless of the random number.

The result is that the TNC waits for an exponentially-distributed random interval afr sensing that the channel has gone clear before attempting to transmit. The idea here is that with proper tuning of the parameters P and SLOTTIME, several stations with traffic to send are much less likely to collide with each other when they simultaneously see the channel go clear."

The form of the command is:

$C0	$02	*n*	$C0
FEND	CTL	p	FEND

where *n* can be 0-255; the persistence parameter p is the fraction: (n + 1)/256

If *n* = 0,	p = 1/256	(lowest value)
If *n* = 255,	p = (155 + 1)/256 = 1.0	(highest value)

4.7.4.3 **SLOTTIME: CTL = $03**

The SLOTTIME command controls the slot interval as described above in the PERSISTENCE command.

The form of the command is:

$C0	$03	*n*	$C0
FEND	CTL	SLO	FEND

where *n* is the slot interval in units of 10 milliseconds.

4.7.4.4 **TXTAIL: CTL = $04**

In both the PK-232's human or verbose mode and the original Host mode, the length of time between the end of data and the release of the PTT line is determined automatically by the baud rate (HBAUD).

In the KISS protocol, the computer controls this interval.

The form of the command is:

$C0	$04	*n*	$C0
FEND	CTL	TXT	FEND

where *n* is the binary value of "TX Tail" time in units of 10 milliseconds.

4.7.4.5 FULLDUP: CTL = $05

The FULLDUD command has the same meaning as in conventional TNCs, that of full duplex operation.

When FULLDUP is ON, the PK-232 transmits without checking to see if the channel is clear.

$C0	$05	*n*	$C0
FEND	CTL	FUL	FEND

where n is $00 for FULLDUP OFF, and $01 for FULLDUP ON.

4.7.4.6 HOST OFF: CTL = $FF

The HOST OFF command returns the PK-232 to the human or verbose mode. HOST OFF has no arguments.

$C0	$FF	$C0
FEND	CTL	FEND

4.7.4.7 DATA: CTL = $00

The DATA block is used to send data from the computer to the PK-232, or from the PK-232 to the computer. There is no data acknowledgement.

$C0	$00	(data)	$C0
FEND	CTL		FEND

NOTE: Data characters that are FEND or FESC characters must be translated to two-byte sequences.

The data must include all AX.25 headers and control bytes. Except for HDLC flags at the beginning and end of the frame and the SDLC checksum, the PK-232 adds nothing to the data transmitted.

4.8 Maximum Block Size

For blocks sent by the host computer to the PK-232, maximum block size is 330 characters, not including SOH, CTL, DLE and ETB.

For blocks sent by the PK-232 to the host, maximum block size depends on the communications mode.

- In Morse, Baudot ASCII and AMTOR, the received and echoed data is divided into blocks containing a maximum of 256 bytes, not including SOH, CTL, DLE and ETB.

- In Packet, the worst-case is a monitored frame containing the addresses of eight digipeaters and 256 bytes of data, with the PK-232's MSTAMP, DAYSTAMP and MRPT parameters all set to ON, and MONTITOR set to 6.

  This worst-case configration produces a maximum of 136 + 256, or 392 bytes, not including SOH, CTL, DLE and ETB. By stuffing couls add as many as 256 additional DLE characters, for a total of 648 bytes.

  If CONMODE is CONVERSE, linefeed characters may be appended to carriage returns because ALFDISP is active. Theoretically, a packet contains 256 carriage returns would result in a block of 136 + 256 + 256, or 648 bytes, plus SOH, CTL and ETB.

4.9 MEMORY, I/O and ADDRESS Commands

The Memory and I/O commands work with the ADDRESS command to permit host access to the PK-232's mmory an I/O locations.

4.9.1 MEMORY Command

To use the Memory command:

- Set the memory address into the ADDRESS command.
- Use the MEMORY command <u>without</u> arguments to read memory locations one after another.
- Use MEMORY with one argument 0-$FF to write to memory locations. After each MEMORY command, the PK-232 adds 1 to the value of the ADDRESS.

PK-232 RAM locations are $C0000-$FFFF. ROM begins at $0000.

4.9.2 ADDRESS Command

To access I/O locations, set ADDRESS as follows:

ADDRESS $*aabb* or SOH $4F A E $ *a a b b* ETB (host mode)

where *aa* is the device address: 7C = 8530 terminal CTRL port
7E = 8530 HDLC CTRL port
BF = 8536 timer/parallel CTRL port

and *bb* is the register address on the device.

4.9.3 I/O Command

Use the I/O command without arguments to read an I/O location and with one argument 0-FF to write to an I/O location. The value in ADDRESS is not incremented after using the I/O command.

Load ADDRESS with these values for easy access:

$BF0D Parallel port A
$BF0E Parallel port B
$BF0F Parallel port C
$7C08 Terminal data
$7E08 HDLC data
$7E00 HDLC RR0

HDLC RR0 is the status register of the radio interface. The following information can be read from RR00:

bit 7 Break/Abort
bit 6 TX Underrun/End of message
bit 5 CTS (Read data)
bit 4 Sync/Hunt
bit 3 DCD
bit 2 Transmit buffer empty
bit 1 Zero count
bit 0 Receive character available

4.10 Converse and Transparent Modes

In Host mode, data is sent using the setting of CONMODE.

- o Do not send the CONVERSE or TRANS commands to the PK-232.
- o If CONMODE is CONV, the parameters 8BITCONV, ALFPACK, ALFDISP, LCOK and ESCAPE are active.
- o If CONMODE is TRANS, all characters are passed without modification.

These are the only differences between CONV and TRANS while in Host mode.

4.11 MHEARD Command in Host mode

The MH command has been altered in Host mode because the verbose mode MHEARD response is potentially too long for the limited Host mode response buffer.

The MHEARD response is divided into lines numbered 0-17. The MHEARD list must be polled on a line-by-line basis.

- o Use the host command MH0 (SOH $4F M H $30 ETB), then MH1, MH2, etc. until you get an empty response (SOH $4F M H $00 ETB), or until you reach the last (MH17) line.

CAUTION: If the PK-232 receives a Packet frame while the computer is polling the middle of the MHEARD list, the list entries may be garbled. One possible solution would be to temporarily set HBAUD to 110 to disable Packet, poll the 18 MHEARD lines, then set HBAUD back to 1200.

4.12 Software Release Data Code

The PK-232's software version (date code) can be read from memory locations $0006-$0008.

For the 29. JUL. 86 release, the contents are:

$86 $07 $29

4.13 Product Type Code

The product type is available at EPROM location $0009. The byte return is:

bits 7 6 5	device	bits 4 3 2 1 0	product
0 0 1	PK-80	0 0 0 0 1	PK-87
0 1 0	PK-87	0 0 0 1 0	PK-232
0 1 1	PK-232	0 0 0 1 1	PK-81T
1 0 0	UDC-232	0 0 1 0 0	PK-90
		0 0 1 0 1	PK-80E
		0 0 1 1 0	PK-87J
		0 0 1 1 1	SPR-10A
		0 1 0 0 0	PK-80I
		0 1 0 0 1	HK-232

Previous software versions arre identifiable by a value of $C3.

CHAPTER 5 – MECHANICAL ASSEMBLY AND DISASSEMBLY

CAUTION

DISCONNECT ALL POWER FROM THE PK-232 BEFORE OPENING THE PK-232 CASE. FAILURE TO DISCONNECT THE POWER SOURCE MAY DAMAGE THE PK-232!!

5.1. Cover Removal

- Disconnect all connectors from the rear of the PK-232.
- Remove six black anodized #6 Phillips head screws from the sides and rear of the cover.
- Slide the cover towards the rear of the unit until the connectors clear the cutouts on the rear of the cover.
- Lift the cover and carefully tip it reward. You will see a black wire and a red wire connected from the circuit board to the battery holder fastened to the cover. Use caution not to break these wires. The leads are long enough to allow the cover to be set on edge next to the chassis without disconnecting the wires from the board.

5.2. Circuit Board Removal

Refer to the Printed Circuit Board Drawing, Figure 4 in APPENDIX C, for the locations of the jumpers and other components.

- Remove the jumper from JP-1 to disconnect the battery backup voltage from the RAM circuit. All preset parameters will be lost and will have to be reentered when the unit is powered up.
- With a small blade screwdriver, loosen the screw holding the control knob to the shaft of the THRESHOLD control and remove the knob.
- Remove the control nut and control washer.
- Unsolder the red and black leads where they enter the circuit board. This can be done from the top of the circuit board.
- Slide the circuit board towards the rear of the chassis until the control shaft clears the front panel hole and lift the circuit board clear of the chassis.

5.3. Circuit Board Assembly

Before starting the assembly make sure that the front panel indicator lights are properly aligned and not to bent.

- Slide the circuit board towards the front panel. Tip the rear edge of the board slightly upward to aid in aligning the LED indicators with the holes in the front panel.
- Fasten the circuit board to the chassis with the six #6 black anodized Phillips head screws.
- Fasten the control shaft to the front panel using the control washer and control nut.
- Turn the shaft of the control knob fully counterclockwise and place the knob on the shaft with the white index mark pointing to the "R" in the Word PAKRATT on the panel.
- Tighten the knob securely on the shaft.
- Place the cover edge next to the chassis and solder the black lead from the battery holder to the land closest to the switch assembly.
- Solder the red lead to the land closest to JP-1.
- Replace the Jumper on JP-1.

5.4. Cover Assembly

- Lower the cover onto the chassis and slide it forward so that the rear connectors extend through the cutouts in the rear of the cover.
- Replace six #6 Phillips head screws on the sides and rear of the cover.

This completes the assembly.

CHAPTER 6 – ADJUSTMENTS AND TESTS

6.1. Preliminary Setup

Connect the P to a well-regulated source of 13 Volts DC. The power supply must be rated at one ampere minimum.

Connect the terminal to the PK-232's RS-232 connector.

Set the terminal for:

BAUD RATE	1200
DATA BITS	7
PARITY	NONE
STOP BIT	1

Apply power to the PK-232 and observe the sign-on message on the terminal's screen:

`Please type a star (*) for auto-baud routine`

Type a few stars (*****) until the screen shows the AEA sign-on message.

6.2. Calibration Procedure

Refer to Figures 2, 3 and 4 in APPENDIX C for circuit diagrams and printed circuit board component locations, and also to APPENDIX D for sample signal waveforms.

The AFSK generator frequencies will be adjusted using the PK-232's internal calibration routine and built-in frequency meter. An external frequency counter may also be connected at the junction of R160 and C59.

Allow the PK-232 at least 20 minutes to reach normal operating temperatures before starting the calibration routines.

Preset variable resistors R164, 165, 167 and 168 fully CCW (counterclockwise). Preset R155 to the center of its range.

NOTE: Commands are shown enclosed in quotation marks. Do not type the quotation marks when entering commands. Command may also be written in red letters.

6.2.1. AFSK Generator (Transmit) Adjustments

While in calibration routine use:

- o [SPACE BAR] to toggle between mark and space;
- o [H] to toggle between VHF (1200/2200 Hz) and HF (2110/2310 Hz);
- o [K] to toggle Push-to-talk (PTT) and key the transmitter;
- o [Q] to exit the calibration program.

1. Connect a wire jumper or loopback shorting plug between pins 1 and 2 on the PK-232's **RADIO 1** (J4) receptacle.
2. Type **MY AAA** followed by a [RETURN] (or [ENTER] key). The monitor should display:

```
MYCALL   was PK232
MYCALL   now AAA
```

3. Type **CAL** followed by the [RETURN] key.
4. Adjust R167 for a reading of 1200 ± 10 Hz as display on the screen or frequency counter. Use the [SPACE BAR] to toggle between mark and space. If adjusting R167 has no effect, press the [SPACE BAR].
5. Press the [SPACE BAR].
6. Adjust R165 for reading 2200 ± 10 Hz. as displayed on the screen or frequency counter.
7. Press the [H] key to change to narrow-shift operation.
8. Adjust R164 for a reading of 2310 ± 5 Hz. If adjusting R164 has no effect, press the [SPACE BAR] once and try again.
9. Press the [SPACE BAR].
10. Adjust R168 for a reading of 2110 ± 5 Hz.
11. Connect an oscilloscope to the junction of R160 and C59.
12. Adjust R157 for minimum signal. This is the AFSK null adjustment.

6.2.2. Demodulator (Receive) Adjustment

1. Connect the oscilloscope to U30 pin 14.
2. Adjust R81 for maximum signal amplitude.
3. Press the [SPACE BAR]. The on-screen frequency counter should read 2310 Hz.
4. Connect the oscilloscope to U30 pin 1.
5. Adjust R96 for maximum signal amplitude.
6. Type **Q** to exit the calibration routine.

6.3. Functional Test

1. Type **MY AEA** [RETURN].
   The monitor should display **MYCALL was AAA**.
2. Type **C AEA** [RETURN].
   The monitor should display ***** CONNECTED TO AEA**.
3. Type a few characters and press [RETURN]. These same characters should be echoed back to the terminal.
4. Type **^C** <CTRL-C>.
   The monitor should display **CMD:**.
5. Type **K** [RETURN] to enter CONVERSE mode.
6. Adjust the PK-232's **THRESHOLD** control clockwise until the **DCD LED** on the PK-232's front panel is lit. Type several characters and a [RETURN]. Observe the **SEND LED** on the front panel. The **SEND LED** should NOT be lit.
7. Adjust the **THRESHOLD** control counterclockwise until the **DCD LED** is extinguished. Observe that the **SEND LED** is lit and that the characters that were typed are echoed on the screen.
8. Adjust R155 (AFSK output level) to 200 millivolts peak-to-peak as measured at the junction of C59 and R160.
9. Type **^C**.
10. Type **VHF OFF** [RETURN].
    The monitor should echo **VHF was ON**.
11. Type **HB 300** [RETURN].
    The monitor should echo **HBAUD was 1200**.
12. Type **K** [RETURN].
13. Type several characters followed by a [RETURN].

14. Type **^C**.
    The monitor should echo back **CMD:**.

15. Adjust R155 (AFSK output level) to 10 millivolts peak-to-peak as measured at the junction of C59 and R160.

16. Type **C AEA** [RETURN].
    The PK-232 responds ***** CONNECTED TO AEA**.

17. Type a few characters and press [RETURN].

18. Type **^C**.
    The monitor should echo back **CMD:**.

19. Type **D** [RETURN].
    The monitor should respond ***** DISCONNECTED**.

This completes the functional tests.

CHAPTER 7 – TROUBLESHOOTING

7.1. Introduction

WARNING!
Never remove or insert an integrated circuit with power on!

The AEA PK-232 is a complex piece of electronic equipment. Servicing must be performed in a logical manner. Prepare for troubleshooting by studying the circuit description in Chapter 3 and the circuit diagrams in APPENDIX C.

Although it is not possible to present all possible problems, symptoms and probable solutions, this section offers general troubleshooting directions based on our experience.

7.2. General Tests

In most cases, careful visual inspection combined with simple measurments usually reveals the problem.

The single most-useful tool for troubleshooting is a digital voltmeter (DVM) for reading AC and DC voltages, one that permits non-destructive resistance measurements while the integrated circuits are still in their sockets.

Although certain tests can be done without the aid of an oscilloscope, it will be required to verify signals at various points on the board if the problem cannot be located by visual means or with a meter.

Avoid short-circuiting on integrated circuits when connecting meter or oscilloscope probes to the board. It is good parctice to attach a secure ground wire to the meter or oscilloscope, some point that cannot accidentally short-circuit components on the board.

A good point to pick up this ground is on the threads of the screws that mount RS-232 I/O connector at the rear of the printed-circuit board.

7.2.1. Power Supply

Verfiy the poer supply fpr correct operation. Verify power supply levels at the outputs of voltage regulators U21 and U41. as well as the output of U28.

- Are the close to their nominal values?
- Do all the integrated circuits in the suspected area have the proper voltage on their power pins?
- Is there excessive ripple in any of the DC voltage lines?
- If so, verify the regulaor and associated components, workin bachwards toward the input power source.
- If the voltage is low in conjunction with a hot regulator, suspect a short circuit on the board.

7.2.2. Obvious Problems

Look for any unusual physical symtoms.

- o Are any components discolored?
- o Does something smell burned?
- o Do any of the parts seems excessively warm?

7.2.3. Assembly Problems

Carefully inspect the PC board and component installation.

- o Are all integrated circuits firmly seated in their sockets?
- o Are any integrated circuits leads tucked under the chip or bent so that they aren't making proper contact with the integrated circuit socket?

7.2.4. Cabling Problems

Inspect the interconnection cabling.

- o Do the cables perform correctly with another controller?
- o Has the radio and/or terminal been successfully used with this or another controller?
- o Are all connections tight?
- o Does the cable appear frayed or broken?

7.3. Specific Symptoms

Although the above steps may seem obvious, careful visual inspection often points to a problem or provides significant indications as to the controller's most suspect area.

Proceed to more specific analysis after completing the physical inspection and delaing with the apparent problem.

7.3.1. Symptom: Controller appears dead

If the controller responds to initial power application with the **MULT**, **STA**, **SEND** and **CON LED**s lit but fails to respond to any commands:

- o Verify that the integrated circuits at locations U2 and U3 are correctly installed with the indicator notch position matching the marking on the printed-circuit board marking.
- o Verify that the controller's power source can provide at least 700 milliamperes continuos current at 13 volts.

If the controller responds to initial power application with only the **BAUDOT LED** lit but fails to respond to any commands:

- o Suspect the terminal port at this point. The processor and the software in EPROM are probably operating correctly.
- o Verify all cables and connections between the controller and the terminal.
- o Verify logic levels according to the terminal interface troubleshooting section in this chapter.

Oscillator and Reset Circuits

If no LEDs are illuminated during the startup or reset cycle:

- o Verify that system clock crystal oscillator Y1 is operating and that a square wave signal (4 MHz, 0 to +5 volts) exists at U1 pin 6. The clock signal should be a moderately-distorted square wave.
- o Verfiy that baud-rate generator crystal oscillator Y2 is operating and that a square wave signal (2.4576 MHz, 0 to +5 volts) exists at U7 pin 29.
- o Verify line receiver U19 and diodes D1 and D2 in the reset timer circuits.

Digital Logic Lines

All logic circuits operate at standard TTL levels. "Low" is less than +0.8 V; "High" is greater than +2.4 volts. All digital inputs and outputs alternate between these two levels.

- o If logic signals are alternating between 0 and perhaps 1 volt, there is a problem, usually a short circuit.
- o Do not mistake switching transients on digital logic lines for improper operation – such as transients appear as ringing anf other distortions.
- o Verify with the oscilloscope and the waveform diagrams shown in APPENDIX D that there is activity on the following lines:

ADDRESS LINE A0	U1 Pin 30
DATA LINE D0	U1 Pin 14
ROM CHIP ENABLE CE*	U20 Pin 20
OUTPUT ENABLE OE*	U2 Pin 22
CE*/OE* TIMING	U2 Pin 20, U2 Pin22
READ LINE RD*	U1 Pin 21
WRITE LINE WR*	U1 Pin 22
PCLK	U7 Pin 20
CLOCK TRXCA	U7 Pin 14
CLOCK RTXCA	U7 Pin 12

Each of these lines should show activity. If any line is inactive, this is a sign of trouble.

Logic lines that do not show activity can often be traced to a short circuit on the printed circuit board.

short circuits an the address and data lines can also appear as lack of activity on the control bus lines, especially device select lines.

- o Verify each of the 16 address and 8 data lines for activity. Any lines showing a lack of activity are not operating properly.
- o Remove all memory chips if you suspect problems with address or data lines. Each address and data line will now show a distinct pattern. The address lines should be (possibly distorted) square waves whose periods increase by a fyctor of two on successive lines as you move line by line from A0 to A15.
- o Use a low-voltage, low-current test instrument if you decide to use an ohmmeter to verify short-circuited lines. Most modern DVMs are adequate for such tests. Remove any integrated circuits connected to the lines being measured if in doubt.
- o Verify the high-density areas of the printed circuit board for the problem if you suspect a short circuit. In most cases the short will be found there.

7.3.2. Symptom: Modem cannot be calibrated

If the calibration signalis present, but you cannot seccessfully calibrate the frequency, the value of a frequency-determining component may have changed.

- o Attempt the calibration procedures presented in Chapter 6.
- o Verify that the proper signals are present at U40 pin 2.
- o Verify the signal frequency with a frequency counter.
- o verfiy the values of the appropriate passive components.

7.3.3. Symptom: Transmitter cannot be keyed

If the transmitter cannot be keyed and the **DCD Threshold** control functions normally and causes the **DCD LED** to be lighted or extinguished:

- o The problem may be in the watchdog timer circuit at Q10 and Q3, or the PTT driver transistors Q4 and Q5. Verify timing capacitor C55.

If **DCD Threshold** control has no effect on the **DCD LED** and the LED is permanently lighted:

- o The problem may be in the DCD Threshold circuits at U34. Verify variable resistor R136 for an open lead.

7.3.4. Symptom: Transmitted signals not copyable by other stations

If other stations are unable to decode your transmissions when using FM:

- o Verify the associated transmitter's modulation index. Verify that peak deviation at any tone does not exceed 4 kHz.
- o Adjust the AFSK output level with R155 to produce the correct transmitter deviation.

If other stations are unable to decode your transmission when using an SSB transmitter:

- o Verfiy that the transmitter's audio input stage and ALC systems are not being overdriven.
- o Adjust the transmitter's microphone gain in accordance with the radio manufacturer's duty-cycle, plate current and power dissipation specifications.
- o If reducing the radio's gain control does not solve the problem adjust the AFSK output level with R155 to produce the correct transmitter operating conditions.

7.3.5. Symptom: Received signals not copyable

If unable to correctly decode signals from other stations:

- o Perform the calibration procedures presented in Chapter 6.
- o Verify the correct settings of variable resistors R81 and R96.
- o Verify that the DCD Threshold control is operating correctly.

If the calibration procedure is satisfactory:

- o Inject a 1200-Hz test tone into J3 or J5.
- o Verify audio signal flow through switch SW2, C54, R34, U28-14 and U28-7.

7.4. Terminal Interface Troubleshooting

If the controller does not appear to start, respond to commands or accept data from the terminal or computer, the problem may be in the RS-232 interface. The following troubleshooting suggestions can aid in resolving problems related to the RS-232C port.

7.4.1. Symptom: Controller does not communicate with the terminal

Use a 'breakout box' or oscilloscope to verify that correct control voltages are present on pin 4, 5, 6, 8 and 20 of J2, the RS-232 I/O connector.

- o Verify that the DTR line on Pin 2 of J2, the RS-232 I/O connector is not being held low.

If the controller software flow control is disabled by setting START, STOP, XON and XOFF to $00 and XFLOW to OFF, the controller will not send data to the terminal unless its DTR is asserted.

If th computer or terminal does not provide the DTR/CTS protocol or "handshake", the DTR/CTS lines (pins 20 and 5 on J2) should not be connected.

- o Verify that the voltages on the controller are correct.

If the tests are valid, verify the signal on U7 pin 27 with an oscilloscope.

- o Recycle the power switch on the controller. Transitions on this pin shortly after reset indicate that the controller is sending data.
- o Verify that transitions are also present on U10 pin 1.

7.4.2. Symptom: Controller signs on with mutilated data

If the terminal displays strange characters, 'garbage', graphics, etc., one or more of the following parameter values is incorrectly set and does not agree between the terminal and the controller:

data rate (TBAUD)
data word length (AWLEN)
parity (PARITY)
number of start and stop bits

- o Set the terminal 1200 bauds if possible, seven data bits, even parity, and one stop bit. These are the default settings stored in EPROM.
- o Restart the controller by cycling the power switch OFF then ON (out then in). The sign on message should appear.

If the controller still prints gibberish:

- o Verify that the terminal is set to 1200 bauds and recycle the power switches on both the controller and terminal.

If the sign-on message still fails to appear:

- o Verify signals with an oscilloscope connected to TXD pin 25 of U7, the Z8530 chip, and then at the X32 baud rate clock (38.4 kHz at 1200 bauds) on pin 14 of U7.

7.4.3. Symptom: Controller does not respond or accept commands

Type a command such as MYCALL or any other command. If the default settings are in effect, the controller should echo typed characters back to the screen.

- o Verify that U7 pin 21 shows a positive voltage level. If not, the fault could be in U19.

If the above tests are valid:

- o Type any keys on the terminal and verify that data is present on U7 pin 27 and U19 pin 1.

If data is not seen, the data is not reaching the controller from the terminal.

- o Verify J2, the cable and U19 again.

APPENDIX A – AX.25 LEVEL 2 PROTOCOL

3/13/83

Terry Fox, WB4JFI
Vice-President, AMRAD
1819 Anderson Road
Falls Church, VA 22043

Abstract

This paper contains the latest draft of the AX.25 protocol specification. This is the first public release of this draft. Earlier drafts have been given to specific individuls for comment and as a reference for software development. Changes should be expected. Please check the AMRD Newsletter for announcements of later versions.

History

Over the years there have been several protcols suggested for use at layer 2 of the ISO Open System Interface Reference Model (OSI-RM) over Amateur Radio. The open system that has been in use is based on the IBM SDLC protocol, and it has been working as far as it went. One of the immediate problems that came up with SDLC was that the address field of SDLC is very limited (being one byte long), causing problems if there are many amateurs on at a time.

Trying to come up with a protocol that everyone would agree to seemed like an almost impossible task a year ago. What we at AMRAD decided to do was to go over the various protocols in use or available to the amateur, figure out the best and worst parts of each prtocol and see id the protocol could be "enhanced" to work properly over the amateur radio enviroment. After reviewing the various protocols araund and talking with people in the computer networking industry, we decided to push the AX.25 standard, modified to allow a larger address field. A about this time, a group of amateurs in New Jersy were coming to the same conclusion, so about mid-June of 1982 the two groups got together and after two weekends came to an "understanding" on a level 2 protocol. The most delicate part of the negotiations between the two groups concerned the name to be given to the protocol AX.25, which stands for Amateur X.25.

The next step in the evolvement of AX.25 was that in October of 1982, AMSAT hosted a gathering of some leaders in amateur packet radio. AMRAD was at the meeting, along with representatives from TAPR, SLAPR, AMSAT and PPRS.

Three days of intense discussion followed, and an agreement was finally reached on a nationwide compaible protocol. AX.25 was then modiefied to be compatible with this new protocol (basically the only major changes were an additional extension of the address field and the addition of a Protocol IDentifier, or PID field).

The rest of this paper will describe the basics of the AX.25 level 2 protocol.

AX.25 Layer 2 Protocol Specification

A.1 Scope and Field of Operation

In order to provide a mechanism for the reliable transport of data between two signaling terminals, it is necessary to define a protocol that can accept and deliver data over a variety of types of communications links. The AX.25 Link- Layer Protocol is designed to provide this service, independent of any other level that may or may not exist.

This protocol conforms to ISO Recommendations 3309, 4335 (including DAD 1&2) and 6256 high-level data link control (HDLC) and uses some terminology found in these documents. It also conforms with ANSI X3.66, which describes ADCCP, balanced mode.

This protocol follows, in principle, the CCITT X.25 Recommendation, with the exception of an extended address field and the addition of the Unnumbered Information (UI) frame. It also follows the principles of CCITT Recommendation Q.921 (LAPD) in the use of multiple links, distinguished by the address field, on a single shared channel.

As defined, this protocol will work equally well in either half- or full-duplex Amateur Radio environments. This protocol has been designed to work equally well for direct connections between two individual amateur packet-radio stations or an individual station and a multiport controller.

This protocol allows for the establishment of more than one link-layer connection per device, if the device is so capable. This protocol does not prohibit self-connections. A self-connection is considered to be when a device establishes a link to itself using its own address for both the source and destination of the frame.

Most link-layer protocols assume that one primary (or master) device (generally called a DCE, or data circuit- terminating equipment) is connected to one or more secondary (or slave) device(s) (usually called a DTE, or data terminating equipment). This type of unbalanced operation is not practical in a shared-RF Amateur Radio environment. Instead, AX.25 assumes that both ends of the link are of the same class, thereby eliminating the two different classes of devices. The term DXE is used in this protocol specification to describe the balanced type of device found in amateur packet radio.

A.2 Frame Structure

Link layer packet radio transmissions are sent in small blocks of data, called frames. Each frame is made up of several smaller groups, called fields. Fig.1 shows the three basic types of frames. Note that the first bit to be transmitted is on the left side.

Fig. 1A – U and S frame construction

First Bit Sent				
Flag	**Address**	**Control**	**FCS**	**Flag**
01111110	112/560 Bits	8 Bits	16 Bits	01111110

Fig. 1B – Information frame construction

First Bit Sent						
Flag	**Address**	**Control**	**PID**	**Info.**	**FCS**	**Flag**
01111110	112/560 Bits	8 Bits	8 Bits	N*8 Bits	16 Bits	01111110

Each field is made up of an integral number of octets (or bytes), and serves a specific function as outlined below.

A.2.1 Flag Field

The flag field is one octet long. Since the flag is used to delimit frames, it occurs at both the beginning and end of each frame. Two frames may share one flag, which would denote the end of the first frame, and the start of the next frame. A flag consists of a zero followed by six ones followed by another zero, or 01111110 (7E hex). As a result of bit stuffing (see A.2.6, below), this sequence is not allowed to occur anywhere else inside a complete frame.

A.2.2 Address Field

The address field is used to identify both the source of the frame and its destination. In addition, the address field contains the command/response information and facilities for level 2 repeater operation. The encoding of the address field is described in A.2.13.

A.2.3 Control Field

The control field is used to identify the type of frame being passed and control several attributes of the level 2 connection. It is one octet in length, and its encoding is discussed in A.3.2.1, below.

A.2.4 PID Field

The Protocol Identifier (PID) field shall appear in information frames (I and UI) only. It identifies what kind of layer 3 protocol, if any, is in use.

The PID itself is not included as part of the octet count of the information field. The encoding of the PID is as follows:

HEX	MSB LSB	Translation
0x01	**00000001**	ISO 8208/CCITT X.25 PLP
0x06	**00000110**	Compressed TCP/IP packet. Van Jacobson (RFC 1144)
0x07	**00000111**	Uncompressed TCP/IP packet. Van Jacobson (RFC 1144)
0x08	**00001000**	Segmentation fragment
**	**yy01yyyy**	AX.25 layer 3 implemented.
**	**yy10yyyy**	AX.25 layer 3 implemented.
0xC3	**11000011**	TEXNET datagram protocol
0xC4	**11000100**	Link Quality Protocol
0xCA	**11001010**	Appletalk
0xCB	**11001011**	Appletalk ARP
0xCC	**11001100**	ARPA Internet Protocol
0xCD	**11001101**	ARPA Address resolution
0xCE	**11001110**	FlexNet
0xCF	**11001111**	NET/ROM
0xF0	**11110000**	No layer 3 protocol implemented.
0xFF	**11111111**	Escape character. Next octet contains more Level 3 protocol information.

Where:
A **y** indicates all combinations used.

Note:
All forms of **`yy11yyyy`** and **`yy00yyyy`** other than those listed above are reserved at this time for future level 3 protocols. The assignment of these formats is up to amateur agreement. It is recommended that the creators of level 3 protocols contact the ARRL Ad Hoc Committee on Digital Communications for suggested encodings.

A.2.5 Information Field

The information field is used to convey user data from one end of the link to the other. I fields are allowed in only three types of frames: the I frame, the UI frame, and the FRMR frame. The I field can be up to 256 octets long, and shall contain an integral number of octets. These constraints apply prior to the insertion of zero bits as specified in A.2.6, below. Any information in the I field shall be passed along the link transparently, except for the zero-bit insertion (see A.2.6) necessary to prevent flags from accidentally appearing in the I field.

A.2.6 Bit Stuffing

In order to assure that the flag bit sequence mentioned above doesn't appear accidentally anywhere else in a frame, the sending station shall monitor the bit sequence for a group of five or more contiguous one bits. Any time five contiguous one bits are sent the sending station shall insert a zero bit after the fifth one bit. During frame reception, any time five contiguous one bits are received, a zero bit immediately following five one bits shall be discarded.

A.2.7 Frame-Check Sequence

The frame-check sequence (FCS) is a sixteen-bit number calculated by both the sender and receiver of a frame. It is used to insure that the frame was not corrupted by the medium used to get the frame from the sender to the receiver. It shall be calculated in accordance with ISO 3309 (HDLC) Recommendations.

A.2.8 Order of Bit Transmission

With the exception of the FCS field, all fields of an AX.25 frame shall be sent with each octet's least-significant bit first. The FCS shall be sent most-significant bit first.

A.2.9 Invalid Frames

Any frame consisting of less than 136 bits (including the opening and closing flags), not bounded by opening and closing flags, or not octet aligned (an integral number of octets), shall be considered an invalid frame by the link layer. See also A.4.4.4, below.

A.2.10 Frame Abort

If a frame must be prematurely aborted, at least fifteen contiguous ones shall be sent with no bit stuffing added.

A.2.11 Interframe Time Fill

Whenever it is necessary for a DXE to keep its transmitter on while not actually sending frames, the time between frames should be filled with contiguous flags.

A.2.12 Link Channel States

Not applicable.

A.2.13 Address-Field Encoding

The address field of all frames shall be encoded with both the destination and source amateur call signs for the frame. Except for the Secondary Station Identifier (SSID), the address field should be made up of upper-case alpha and numeric ASCII characters only. If level 2 amateur "repeaters" are to be used, their call signs shall also be in the address field.

The HDLC address field is extended beyond one octet by assigning the least-significant bit of each octet to be an "extension bit". The extension bit of each octet is set to zero, to indicate the next octet contains more address information, or one, to indicate this is the last octet of the HDLC address field. To make room for this extension bit, the amateur Radio call sign information is shifted one bit left.

A.2.13.1 Nonrepeater Address-Field Encoding

If level 2 repeaters are not being used, the address field is encoded as shown in Fig. 2. The destination address is the call sign and SSID of the amateur radio station to which the frame is addressed, while the source address contains the amateur call sign and SSID of the station that sent the frame. These call signs are the call signs of the two ends of a level 2 AX.25 link only.

Fig. 2 – Nonrepeater Address-Field Encoding

First Octet Sent													
Address Field of Frame													
Destination Address							**Source Address**						
A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14

A1 through A14 are the fourteen octets that make up the two address subfields of the address field. The destination subaddress is seven octets long (A1 thru A7), and is sent first. This address sequence provides the receivers of frames time to check the destination address subfield to see if the frame is addressed to them while the rest of the frame is being received. The source address subfield is then sent in octets A8 through A14. Both of these subfields are encoded in the same manner, except that the last octet of the address field has the HDLC address extension bit set.

There is an octet at the end of each address subfield that contains the Secondary Station Identifier (SSID). The SSID subfield allows an Amateur Radio operator to have more than one packet-radio station operating under the same call sign. This is useful when an amateur wants to put up a repeater in addition to a regular station, for example. The C bits (see A.4.1.2, below) and H bit (see A.2.13.2, below) are also contained in this octet, along with two bits which are reserved for future use.

Fig. 3A shows a typical AX.25 frame in the nonrepeater mode of operation.

Fig. 3A – Nonrepeater AX.25 frame

Octet	ASCII	Bin. Data	Hex Data
Flag		01111110	7E
A1	K	10010110	96
A2	8	01110000	70
A3	M	10011010	9A
A4	M	10011010	9A
A5	O	10011110	9E
A6	space	01000000	40
A7	SSID	11100000	E0
A8	W	10101110	AE
A9	B	10000100	84
A10	4	01100100	68
A11	J	10010100	94
A12	F	10001100	8C
A13	I	10010010	92
A14	SSID	01100001	61
Control	I	00111110	3E
PID	none	11110000	F0
FCS	part 1	xxxxxxxx	HH
FCS	part 2	xxxxxxxx	HH
Flag		01111110	7E
Bit position		76543210	

The frame shown is an I frame, not going through a level 2 repeater, from WB4JFI (SSID=0) to K8MMO (SSID=0), with no level 3 protocol. The P/F bit is set; the receive sequence number (N(R)) is 1; the send sequence number (N(S)) is 7.

A.2.13.1.1 Destination Subfield Encoding

Fig. 3 shows how an amateur call sign is placed in the destination address subfield, occupying octets A1 thru A7.

Fig. 3 – Destination Field Encoding

Octet	ASCII	Bin. Data	Hex Data
A1	W	10101110	AE
A2	B	10000100	84
A3	4	01101000	68
A4	J	10010100	94
A5	F	10001100	8C
A6	I	10010010	92
A7	SSID	CRRSSID0	
Bit Position		76543210	

Where:

1. The top octet (A1) is the first octet sent, with bit 0 of each octet being the first bit sent, and bit 7 being the last bit sent.
2. The first (low-order or bit 0) bit of each octet is the HDLC address extension bit, which is set to zero on all but the last octet in the address field, where it is set to one.
3. The bits marked "r" are reserved bits. They may be used in an agreed-upon manner in individual networks. When not implemented, they should be set to one.

4. The bit marked "C" is used as the command/response bit of an AX.25 frame, as outlined in A.4.1.2 below.
5. The characters of the call sign should be standard seven-bit ASCII (upper case only) placed in the leftmost seven bits of the octet to make room for the address extension bit. If the call sign contains fewer than six characters, it should be padded with ASCII spaces between the last call sign character and the SSID octet.
6. The 0000 SSID is reserved for the first personal AX.25 station. This establishes one standard SSID for "normal" stations to use for the first station.

A.2.13.2 Level 2 Repeater-Address Encoding

If a frame is to go through level 2 amateur packet repeater(s), there is an additional address subfield appended to the end of the address field. This additional subfield contains the call sign(s) of the repeater(s) to be used. This allows more than one repeater to share the same RF channel. If this subfield exists, the last octet of the source subfield has its address extension bit set to zero, indicating that more address-field data follows. The repeater-address subfield is encoded in the same manner as the destination and source address subfields, except for the most-significant bit in the last octet, called the "H" bit. The H bit is used to indicate whether a frame has been repeated or not.

In order to provide some method of indicating when a frame has been repeated, the H bit is set to zero on frames going to a repeater. The repeater will set the H bit to one when the frame is retransmitted. Stations should monitor the H bit, and discard any frames going to the repeater (uplink frames), while operating through a repeater. Fig. 4 shows how the repeater- address subfield is encoded. Fig. 4A is an example of a complete frame after being repeated.

Fig. 4 – Repeater-address encoding

Octet	ASCII	Bin. Data	Hex Data
A15	W	**10101110**	AE
A16	B	**10000100**	84
A17	4	**01101000**	68
A18	J	**10010100**	94
A19	F	**10001100**	8C
A20	I	**10010010**	92
A21	SSID	**HRRSSID1**	
Bit Order -->		**76543210**	

Where:
1. The top octet is the first octet sent, with bit 0 being sent first and bit 7 sent last of each octet.
2. As with the source and destination address subfields discussed above, bit 0 of each octet is the HDLC address extension bit, which is set to zero on all but the last address octet, where it is set to one.
3. The "R" bits are reserved in the same manner as in the source and destination subfields.
4. The "H" bit is the has-been-repeated bit. It is set to zero whenever a frame has not been repeated, and set to one by the repeater when the frame has been repeated.

Fig. 4A – AX.25 frame in repeater mode

Octet	ASCII	Bin. Data	Hex Data
Flag		01111110	7E
A1	K	10010110	96
A2	8	01110000	70
A3	M	10011010	9A
A4	M	10011010	9A
A5	O	10011110	9E
A6	space	01000000	40
A7	SSID	11100000	E0
A8	W	10101110	AE
A9	B	10000100	84
A10	4	01100100	68
A11	J	10010100	94
A12	F	10001100	8C
A13	I	10010010	92
A14	SSID	01100000	60
A15	W	10101110	AE
A16	B	10000100	84
A17	4	01101000	68
A18	J	10010100	94
A19	F	10001100	8C
A20	I	10010010	92
A21	SSID	11100011	E3
Control	I	00111110	3E
PID	none	11110000	F0
FCS	part 1	xxxxxxxx	HH
FCS	part 2	xxxxxxxx	HH
Flag		01111110	7E
Bit position		76543210	

The above frame is the same as Fig. 3A, except for the addition of a repeater-address subfield (WB4JFI, SSID=1). The H bit is set, indicating this is from the output of the repeater.

A.2.13.3 Multiple Repeater Operation

The link-layer AX.25 protocol allows operation through more than one repeater, creating a primitive frame routing mechanism. Up to eight repeaters may be used by extending the repeater-address subfield. When there is more than one repeater address, the repeater address immediately following the source address subfield will be considered the address of the first repeater of a multiple-repeater chain. As a frame progresses through a chain of repeaters, each successive repeater will set the H bit (has-been-repeated bit) in its SSID octet, indicating that the frame has been successfully repeated through it. No other changes to the frame are made (except for the necessary recalculation of the FCS). The destination station can determine the route the frame took to each it by examining the address field.

The number of repeater addresses is variable. All but the last repeater address will have the address extension bits of all octets set to zero, as will all but the last octet (SSID octet) of the last repeater address. The last octet of the last repeater address will have the address extension bit set to one, indicating the end of the address field.

It should be noted that various timers (see A.4.7, below) may have to be adjusted to accommodate the additional delays encountered when a frame must pass through a multiple-repeater chain, and the return acknowledgement must travel through the same path before reaching the source device.

It is anticipated that multiple-repeater operation is a temporary method of interconnecting stations over large distances until such time that a layer 3 protocol is in use. Once this layer 3 protocol becomes operational, repeater chaining should be phased out.

A.3 Elements of Procedure

A.3.1

The elements of procedure are defined in terms of actions that occur on receipt of frames.

A.3.2 Control-Field Formats and State Variables

A.3.2.1 Control-Field Formats

The control field is responsible for identifying the type of frame being sent, and is also used to convey commands and responses from one end of the link to the other in order to maintain proper link control.

The control fields used in AX.25 use the CCITT X.25 control fields for balanced operation (LAPB), with an additional control field taken from ADCCP to allow connectionless and round- table operation.

There are three general types of AX.25 frames. They are the Information frame (I frame), the Supervisory frame (S frame), and the Unnumbered frame (U frame). Fig. 5 shows the basic format of the control field associated with these types of frames.

Fig. 5 – Control-field formats

Control-Field Type	**Control-Field Bits**							
	7	6	5	4	3	2	1	0
I Frame	N(R)			P	N(S)			0
S Frame	N(R)			P/F	S	S	0	1
U Frame	M	M	M	P/F	M	M	1	1

Where:
1. Bit 0 is the first bit sent and bit 7 is the last bit sent of the control field.
2. N(S) is the send sequence number (bit 1 is the LSB).
3. N(R) is the receive sequence number (bit 5 is the LSB).
4. The "S" bits are the supervisory function bits, and their encoding is discussed in A.3.4.2.
5. The "M" bits are the unnumbered frame modifier bits and their encoding is discussed in A.3.4.3.
6. The P/F bit is the Poll/Final bit. Its function is described in A.3.3. The distinction between command and response, and therefore the distinction between P bit and F bit, is made by addressing rules discussed in A.4.1.2.

A.3.2.1.1 Information-Transfer Format

All I frames have bit 0 of the control field set to zero. N(S) is the sender's send sequence number (the send sequence number of this frame). N(R) is the sender's receive sequence number (the sequence number of the next expected received frame). These numbers are described in A.3.2.4. In addition, the P/F bit is to be used as described in A.4.2.

A.3.2.1.2 Supervisory Format

Supervisory frames are denoted by having bit 0 of the control field set to one, and bit 1 of the control field set to zero. S frames provide supervisory link control such as acknowledging or requesting retransmission of I frames, and link- level window control. Since S frames do not have an information field, the sender's send variable and the receiver's receive variable are not incremented for S frames. In addition, the P/F bit is used as described in A.4.2.

A.3.2.1.3 Unnumbered Format

Unnumbered frames are distinguished by having both bits 0 and 1 of the control field set to one. U frames are responsible for maintaining additional control over the link beyond what is accomplished with S frames. They are also responsible for establishing and terminating link connections. U frames also allow for the transmission and reception of information outside of the normal flow control. Some U frames may contain information and PID fields. The P/F bit is used as described in A.4.2.

A.3.2.2 Control-Field Parameters

A.3.2.3 Sequence Numbers

Every AX.25 I frame shall be assigned, modulo 8, a sequential number from 0 to 7. This will allow up to seven outstanding I frames per level 2 connection at a time.

A.3.2.4 Frame Variables and Sequence Numbers

A.3.2.4.1 Send State Variable V(S)

The send state variable is a variable that is internal to the DXE and is never sent. It contains the next sequential number to be assigned to the next transmitted I frame. This variable is updated upon the transmission of each I frame.

A.3.2.4.2 Send Sequence Number N(S)

The send sequence number is found in the control field of all I frames. It contains the sequence number of the I frame being sent. Just prior to the transmission of the I frame, N(S) is updated to equal the send state variable.

A.3.2.4.3 Receive State Variable V(R)

The receive state variable is a variable that is internal to the DXE. It contains the sequence number of the next expected received I frame. This variable is updated upon the reception of an error-free I frame whose send sequence number equals the present received state variable value.

A.3.2.4.4 Received Sequence Number N(R)

The received sequence number is in both I and S frames. Prior to sending an I or S frame, this variable is updated to equal that of the received state variable, thus implicitly acknowledging the proper reception of all frames up to and including N(R)-1.

A.3.3 Functions of Poll/Final (P/F) Bit

The P/F bit is used in all types of frames. It is used in a command (poll) mode to request an immediate reply to a frame. The reply to this poll is indicated by setting the response (final) bit in the appropriate frame. Only one outstanding poll condition per direction is allowed at a time. The procedure for P/F bit operation is described in A.4.2.

A.3.4 Control Field Coding for Commands and Responses

The following commands and responses, indicated by their control field encoding, are to be use by the DXE:

A.3.4.1 Information Command Frame Control Field

The function of the information (I) command is to transfer across a data link sequentially numbered frames containing an information field.

The information-frame control field is encoded as shown in Fig. 6. These frames are sequentially numbered by the N(S) subfield to maintain control of their passage over the link-layer connection.

Fig. 6 – I frame control field

Control Field Bits							
7	**6**	**5**	**4**	**3**	**2**		**0**
N(R)			P	N(S)			0

A.3.4.2 Supervisory Frame Control Field

The supervisory frame control fields are encoded as shown in Fig. 7.

Fig. 7 – S frame control fields

Control Field Type		Control Field Bits							
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Receive Ready	RR	N(R)			P/F	0	0	0	1
Receive Not Ready	RNR	N(R)			P/F	0	1	0	1
Reject	REJ	N(R)			P/F	1	0	0	1

The Frame identifiers:

C or SABM	Layer 2 Connect Request
D or DISC	Layer 2 Disconnect Request
I	Information Frame
RR	Receive Ready. System Ready To Receive
RNR or NR	Receive Not Ready. TNC Buffer Full
RJ or REJ	Reject Frame. Out of Sequence or Duplicate
FRMR	Frame Reject. Fatal Error
UI	Unnumbered Information Frame. "Unproto"
DM	Disconnect Mode. System Busy or Disconnected.

A.3.4.2.1 Receive Ready (RR) Command and Response

Receive Ready is used to do the following:

1. to indicate that the sender of the RR is now able to receive more I frames.
2. to acknowledge properly received I frames up to, and including N(R)-1, and
3. to clear a previously set busy condition created by an RNR command having been sent.

The status of the DXE at the other end of the link can be requested by sending a RR command frame with the P-bit set to one.

A.3.4.2.2 Receive Not Ready (RNR) Command and Response

Receive Not Ready is used to indicate to the sender of I frames that the receiving DXE is temporarily busy and cannot accept any more I frames. Frames up to N(R)-1 are acknowledged. Any I frames numbered N(R) and higher that might have been caught between states and not acknowledged when the RNR command was sent are not acknowledged.

The RNR condition can be cleared by the sending of a UA, RR, REJ, or SABM frame.

The status of the DXE at the other end of the link can be requested by sending a RNR command frame with the P bit set to one.

A.3.4.2.3 Reject (REJ) Command and Response

The reject frame is used to request retransmission of I frames starting with N(R). Any frames that were sent with a sequence number of N(R)-1 or less are acknowledged. Additional I frames may be appended to the retransmission of the N(R) frame if there are any.

Only one reject frame condition is allowed in each direction at a time. The reject condition is cleared by the proper reception of I frames up to the I frame that caused the reject condition to be initiated.

The status of the DXE at the other end of the link can be requested by sending a REJ command frame with the P bit set to one.

A.3.4.3 Unnumbered Frame Control Fields

Unnumbered frame control fields are either commands or responses. Fig. 8 shows the layout of U frames implemented within this protocol.

Fig. 8 – U frame control fields

Control Field Type			**Control Field Bits**							
			7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Set Asynchronous Balanced Mode	SABM	Res	0	0	1	P	1	1	1	1
Disconnect	DISC	Cm	0	1	0	P	0	0	1	1
Disconnected Mode	DM	Res	0	0	0	F	1	1	1	1
Unnumbered Acknowledge	UA	Res	0	1	1	F	0	0	1	1
Frame Reject	FRMR	Res	1	0	0	F	0	1	1	1
Unnumbered Information	UI	Either	0	0	0	P/F	0	0	1	1

A.3.4.3.1 Set Asynchronous Balanced Mode (SABM) Command

The SABM command is used to place 2 DXEs in the asynchronous balanced mode. This is a balanced mode of operation known as LAPB where both devices are treated as equals.

Information fields are not allowed in SABM commands. Any outstanding I frames left when the SABM command is issued will remain unacknowledged.

The DXE confirms reception and acceptance of a SABM command by sending a UA response frame at the earliest opportunity. If the DXE is not capable of accepting a SABM command, it should respond with a DM frame if possible.

A.3.4.3.2 Disconnect (DISC) Command

The DISC command is used to terminate a link session between two stations. No information field is permitted in a DISC command frame.

Prior to acting on the DISC frame, the receiving DXE confirms acceptance of the DISC by issuing a UA response frame at its earliest opportunity. The DXE sending the DISC enters the disconnected state when it receives the UA response.

Any unacknowledged I frames left when this command is acted upon will remain unacknowledged.

A.3.4.3.3 Frame Reject (FRMR) Response

A.3.4.3.3.1

The FRMR response frame is sent to report that the receiver of a frame cannot successfully process that frame and that the error condition is not correctable by sending the offending frame again. Typically this condition will appear when a frame without an FCS error has been received with one of the following conditions:

1. The reception of an invalid or not implemented command or response frame.
2. The reception of an I frame whose information field exceeds the agreed-upon length. (See A.4.7.3, below.)
3. The reception of an improper N(R). This usually happens when the N(R) frame has already been sent and acknowledged, or when N(R) is out of sequence with what was expected.
4. The reception of a frame with an information field where one is not allowed, or the reception of a U or S frame whose length is incorrect. Bits W and Y described in A.3.4.3.3.2 should both be set to one to indicate this condition.
5. The reception of a supervisory frame with the F bit set to one, except during a timer recovery condition (see A.4.4.9), or except as a reply to a command frame sent with the P bit set to one. Bit W (described in A.3.4.3.3.2) should be set to one.
6. The reception of an unexpected UA or DM response frame. Bit W should be set to one.
7. The reception of a frame with an invalid N(S). Bit W should be set to one.

An invalid N(R) is defined as one which points to an I frame that previously has been transmitted and acknowledged, or an I frame which has not been transmitted and is not the next sequential I frame pending transmission.

An invalid N(S) is defined as an N(S) that is equal to the last transmitted N(R)+k and is equal to the received state variable V(R), where k is the maximum number of outstanding information frames as defined in A.4.7.4 below.

An invalid or not implemented command or response is defined as a frame with a control field that is unknown to the receiver of this frame.

A.3.4.3.3.2

When a FRMR frame is sent, an information field is added to the frame that contains additional information indicating where the problem occurred. This information field is three octets long and is shown in Fig. 9.

Fig. 9 – FRMR frame information field

Information Field Bits																							
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	Z	Y	X	W	V(R)			CR	V(S)			0	Rejected Frame Control Field							

Where:
1. The rejected frame control field carries the control field of the frame that caused the reject condition. It is in bits 0-7 of the information field.
2. V(S) is the current send state variable of the device reporting the rejection (bit 9 is the low bit).
3. The CR bit is set to zero to indicate the rejected frame was a command, or one if it was a response.
4. V(R) is the current receive state variable of the device reporting rejection (bit 13 is the low bit).
5. If W is set to 1, the control field received was invalid or not implemented.
6. If X is set to 1, the frame that caused the reject condition was considered invalid because it was a U or S frame that had an information field that is not allowed. Bit W must be set to 1 in addition to the X bit.
7. If Y is set to 1, the control field received and returned in bits exceeded the maximum allowed under this recommendation in A.4.7.3, below.
8. If A is set to 1, the control field received and returned in bits 1 to 8 contained an invalid N(R).
9. Bits 8, and 20 to 23 are set to 0.

A.3.4.3.4 Unnumbered Acknowledge (UA) Response

The UA response frame is sent to acknowledge the reception and acceptance of a SABM or DISC command frame. A received command is not actually processed until the UA response frame is sent. Information fields are not permitted in a UA frame.

A.3.4.3.5 Disconnected Mode (DM) Response

The disconnected mode response is sent whenever a DXE receives a frame other than a SABM or UI frame while in a disconnected mode. It is also sent to request a set mode command, or to indicate it cannot accept a connection at the moment. The DM response does not have an information field.

Whenever a SABM frame is a received, and it is determined that a connection is not possible, a DM frame shall be sent. This will indicate that the called station cannot accept a connection at that time.

While a DXE is in the disconnected mode, it will respond to any command other than a SABM or UI frame with a DM response with the P/F bit set to 1.

A.3.4.3.6 Unnumbered Information (UI) Frame

The Unnumbered Information frame contains PID and information fields and is used to pass information along the link outside the normal information controls. This allows information fields to go back and forth on the link bypassing flow control. Since these frames are not acknowledgeable, if one gets obliterated, there is no way to recover it. A received UI frame with the P bit set shall cause a response to be transmitted. This response shall be a DM frame when in the disconnected state or a RR (or RNR, if appropriate) frame in the information transfer state.

A.3.5 Link Error Reporting and Recovery

There are several link-layer errors that are recoverable without terminating the connection. These error situations may occur as a result of malfunctions within the DXE, or if transmission errors occur.

A.3.5.1 DXE Busy Condition

When a DXE becomes temporarily unable to receive I frames, such as when receive buffers are full, it will send a Receive Not Ready (RNR) frame. This informs the other DXE that this DXE cannot handle any more I frames at the moment. This condition is usually cleared by the sending of a UA, RR, REJ, or SABM command frame.

A.3.5.2 Send Sequence Number Error

If the send sequence number, N(S), of an otherwise error- free received frame does not match the receive state variable, V(R), a send sequence error has occurred, and the information field will be discarded. The receiver will not acknowledge this frame, or any other I frames, until N(S) matches V(R).

The control field of the erroneous I frame(s) will be accepted so that link supervisory functions such as checking the P/F bit can still be performed. Because of this updating, the retransmitted I frame may have an updated P bit and N(R).

A.3.5.3 Reject (REJ) Recovery

REJ is used to request a retransmission of I frames following the detection of a N(S) sequence error. Only one outstanding "sent REJ" condition is allowed at a time. This condition is cleared when the requested I frame has been received.

A DXE receiving the REJ command will clear the condition by resending all outstanding I frames (up to the window size), starting with the one indicated in N(R) of the REJ frame.

A.3.5.4 Time-out Error Recovery

A.3.5.4.1 T1 Timer Recovery

If a DXE, due to a transmission error, does not receive (or receives and discards) a single I frame or the last I frame in a sequence of I frames, it will not detect a send-sequence- number error, and therefore will not transmit a REJ. The DXE which transmitted the unacknowledged I frame(s) shall, following the completion of time-out period T1, take appropriate recovery action to determine when I frame retransmission should begin as described in A.4.4.9, below. This condition is cleared by the reception of an acknowledgement for the sent frame(s), or by the link being reset. See A.4.6.

A.3.5.4.2 Timer T3 Recovery

Timer T3 is used to assure the link is still functional during periods of low information transfer. Whenever T1 is not running (no outstanding I frames), T3 is used to periodically poll the other DXE of a link. When T3 times out, a RR or RNR frame is transmitted as a command and with the P bit set. The waiting acknowledgement procedure (A.4.4.9, below) is then executed.

A.3.5.5 Invalid Frame or FCS Error

If an invalid frame is received, or a frame is received with an FCS error, that frame will be discarded with no action taken.

A.3.5.6 Frame Rejection Condition

A frame rejection condition occurs when an otherwise error-free frame has been received with one of the conditions listed in A.3.4.3.3 above.

Once a rejection error occurs, no more I frames are accepted (except for the examination of the P/F bit) until the error is resolved. The error condition is reported to the other DXE by sending a FRMR response frame. See A.4.5.

A.4 Description of AX.25 Procedures

The following describes the procedures used to setup, use, and disconnect a balanced link between two DXE stations.

A.4.1 Address Field Operation

A.4.1.1 Address Information

All transmitted frames shall have address fields conforming to A.2.13, above. All frames shall have both the destination device and the source device addresses in the address field, with the destination address coming first. This allows many links to share the same RF channel. The destination address is always the address of the station(s) to receive the frame, while the source address contains the address of the device that sent the frame.

The destination address can be a group name or club call sign if the point-to-multipoint operation is allowed. Operation with destination addresses other than actual amateur call signs is a subject for further study.

A.4.1.2 Command/Response Procedure

AX.25 Version 2.0 has implemented the command/response information in the address field. In order to maintain compatibility with previous versions of AX.25, the command/response information is conveyed using two bits.

An upward-compatible AX.25 DXE can determine whether it is communicating with a DXE using an older version of this protocol by testing the command/response bit information located in bit 7 of the SSID octets of both the destination and source address subfields. If both C bits are set to zero, the device is using the older protocol. The newer version of the protocol always has one of these two bits set to one and the other set to zero, depending on whether the frame is a command or a response.

The command/response information is encoded into the address field as shown in Fig. 10.

Fig. 10 – Command/Response encoding

Frame Type	Dest. SSID C-Bit	Source SSID C-Bit
Previous versions	0	0
Command (V.2.0)	1	0
Response (V.2.0)	0	1
Previous versions	1	1

Since all frames are considered either commands or responses, a device shall always have one of the bits set to one, and the other bit set to zero.

The use of the command/response information in AX.25 allows S frames to be either commands or responses. This aids maintenance of proper control over the link during the information transfer state.

A.4.2 P/F Bit Procedures

The next response frame returned by the DXE to a SABM or DISC command with the P bit set to 1 will be a UA or DM response with the F bit set to 1.

The next response frame returned to an I frame with the P bit set to 1, received during the information transfer state, will be a RR, RNR, or REJ response with the F bit set to 1.

The next response frame returned to a supervisory command frame with the P bit set to 1, received during the information transfer state, will be a RR, RNR, or REJ response frame with the F bit set to 1.

The next response frame returned to a S or I command frame with the P bit set to 1, received in the disconnected state, will be a DM response frame with the F bit set to 1.

The P bit is used in conjunction with the time-out recovery condition discussed in A.3.5.4, above. When not used, the P/F bit is set to zero.

A.4.3 Procedures For Link Set-Up and Disconnection

A.4.3.1 LAPB Link Connection Establishment

When one DXE wishes to connect to another DXE, it will send a SABM command frame to that device and start timer (T1). If the other DXE is there and able to connect, it will respond with a UA response frame, and reset both of its internal state variables (V(S) and V(R)). The reception of the UA response frame at the other end will cause the DXE requesting the connection to cancel the T1 timer and set its internal state variables to 0.

If the other DXE doesn't respond before T1 times out, the device requesting the connection will resend the SABM frame, and start T1 running again. The DXE should continue to try to establish a connection until it has tried unsuccessfully N2 times. N2 is defined in A.4.7.2, below.

If, upon reception of a SABM command, the DXE decides that it cannot enter the indicated state, it should send a DM frame.

When receiving a DM response, the DXE sending the SABM should cancel its T1 timer, and not enter the information- transfer state.

The DXE sending a SABM command will ignore and discard any frames except SABM, DISC, UA, and DM frames from the other DXE.

Frames other than UA and DM in response to a received SABM will be sent only after the link is set up and if no outstanding SABM exists.

A.4.3.2 Information-Transfer Phase

After establishing a link connection, the DXE will enter the information-transfer state. In this state, the DXE will accept and transmit I and S frames according to the procedure outlined in A.4.4, below.

When receiving a SABM command while in the information- transfer state, the DXE will follow the resetting procedure outlined in A.4.6 below.

A.4.3.3 Link Disconnection

A.4.3.3.1

While in the information-transfer state, either DXE may indicate a request to disconnect the link by transmitting a DISC command frame and starting timer T1 (see A.4.7).

A.4.3.3.2

A DXE, upon receiving a valid DISC command, shall send a UA response frame and enter the disconnected state. A DXE, upon receiving a UA or DM response to a sent DISC command, shall cancel timer T1, and enter the disconnected state.

A.4.3.3.3

If a UA or DM response is not correctly received before T1 times out, the DISC frame should be sent again and T1 restarted. If this happens N2 times, the DXE should enter the disconnected state.

A.4.3.4 Disconnected State

A.4.3.4.1
A DXE in the disconnected state shall monitor received commands and react upon the reception of a SABM as described in A.4.3.1 above and will transmit a DM frame in response to a DISC command.

A.4.3.4.2

In the disconnected state, a DXE may initiate a link set-up as outlined in connection establishment above (A.4.3.1). It may also respond to the reception of a SABM and establish a connection, or it may ignore the SABM and send a DM instead.

A.4.3.4.3

Any DXE receiving a command frame other than a SABM or UI frame with the P bit set to one should respond with a DM frame with the F bit set to one. The offending frame should be ignored.

A.4.3.4.4

When the DXE enters the disconnected state after an error condition or if an internal error has resulted in the DXE being in the disconnected state, the DXE should indicate this by sending a DM response rather than a DISC frame and follow the link disconnection procedure outlined in A.4.3.3.3, above. The DXE may then try to re-establish the link using the link set-up procedure outlined in A.4.3.1, above.

A.4.3.5 Collision Recovery

A.4.3.5.1 Collisions in a Half-Duplex Environment

Collisions of frames in a half-duplex environment are taken care of by the retry nature of the T1 timer and retransmission count variable. No other special action needs to be taken.

A.4.3.5.2 Collisions of Unnumbered Commands

If sent and received SABM or DISC command frames are the same, both DXEs should send a UA response at the earliest opportunity, and both devices should enter the indicated state.

If sent and received SABM or DISC commands are different, both DXEs should enter the disconnected state and transmit a DM frame at the earliest opportunity.

A.4.3.5.3 Collision of a DM with a SABM or DISC

When an unsolicited DM response frame is sent, a collision between it and a SABM or DISC may occur. In order to prevent this DM from being misinterpreted, all unsolicited DM frames should be transmitted with the F bit set to zero. All SABM and DISC frames should be sent with the P bit set to one. This will prevent any confusion when a DM frame is received.

A.4.3.6 Connectionless Operation

In Amateur Radio, there is an additional type of operation that is not feasible using level 2 connections. This operation is the round table, where several amateurs may be engaged in one conversation. This type of operation cannot be accommodated by AX.25 link-layer connections.

The way round-table activity is implemented is technically outside the AX.25 connection, but still using the AX.25 frame structure.

AX.25 uses a special frame for this operation, called the Unnumbered Information (UI) frame. When this type of operation is used, the destination address should have a code word installed in it to prevent the users of that particular round table from seeing all frames going through the shared RF medium. An example of this is if a group of amateurs are in a round-table discussion about packet radio, they could put PACKET in the destination address, so they would receive frames only from others in the same discussion. An added advantage of the use of AX.25 in this manner is that the source of each frame is in the source address subfield, so software could be written to automatically display who is making what comments.

Since this mode is connectionless, there will be no requests for retransmissions of bad frames. Collisions will also occur, with the potential of losing the frames that collided.

A.4.4 Procedures for Information Transfer

Once a connection has been established, as outlined above, both devices are able to accept I, S, and U frames.

A.4.4.1 Sending I Frames

Whenever a DXE has an I frame to transmit, it will send the I frame with N(S) of the control field equal to its current send state variable V(S). Once the I frame is sent, the send state variable is incremented by one. If timer T1 is not running, it should be started. If timer T1 is running, it should be restarted.

The DXE should not transmit any more I frames if its send state variable equals the last received N(R) from the other side of the link plus seven. If it were to send more I frames, the flow control window would be exceed, and errors could result.

If a DXE is in a busy condition, it may still send I frames as long as the other device is not also busy.

If a DXE is in the frame-rejection mode, it will stop sending I frames.

A.4.4.2 Receiving I Frames

A.4.4.2.1

If a DXE receives a valid I frame (one with a correct FCS and whose send sequence number equals the receiver's receive state variable) and is not in the busy condition, it will accept the received I frame, increment its receive state variable, and act in one of the following manners:

1. If it has an I frame to send, that I frame may be sent with the transmitted N(R) equal to its receive state variable V(R) (thus acknowledging the received frame). Alternately, the device may send a RR frame with N(R) equal to V(R), and then send the I frame.
2. If there are no outstanding I frames, the receiving device will send a RR frame with N(R) equal to V(R). The receiving DXE may wait a small period of time before sending the RR frame to be sure additional I frames are not being transmitted.

A.4.4.2.2

If the DXE is in a busy condition, it may ignore any received I frames without reporting this condition other than repeating the indication of the busy condition.

If a busy condition exists, the DXE receiving the busy condition indication should poll the sender of the busy indication periodically until the busy condition disappears.

A DXE may poll the busy DXE periodically with RR or RNR frames with the P bit set to one.

The reception of I frames that contain zero-length information fields shall be reported to the next level but no information field will be transferred.

A.4.4.3 Reception of Out of Sequence Frames

When an I frame is received with a correct FCS, but its send sequence number, N(S), does not match the current receiver's receive state variable, the frame should be discarded. A REJ frame shall be sent with a receive sequence number equal to one higher (modulo 8) than the last correctly received I frame if an uncleared N(S) sequence error condition has not been previously established. The received state variable and poll bit of the discarded frame should be checked and acted upon, if necessary, before discarding the frame.

A.4.4.4 Reception of Incorrect Frames

When a DXE receives a frame with an incorrect FCS, an invalid frame, or a frame with an improper address, that frame shall be discarded.

A.4.4.5 Receiving Acknowledgement

Whenever an I or S frame is correctly received, even in a busy condition, the N(R) of the received frame should be checked to see if it includes an acknowledgement of outstanding sent I frames. The T1 timer should be cancelled if the received frame actually acknowledges previously unacknowledged frames. If the T1 timer is cancelled and there are still some frames that have been sent that are not acknowledged, T1 should be started again. If the T1 timer runs out before an acknowledgement is received, the device should proceed to the retransmission procedure in A.4.4.9.

A.4.4.6 Receiving Reject

Upon receiving a REJ frame, the transmitting DXE will set its send state variable to the same value as the REJ frame's received sequence number in the control field. The DXE will then retransmit any I frame(s) outstanding at the next available opportunity conforming to the following:

1. If the DXE is not transmitting at the time, and the channel is open, the device may commence to retransmit the I frame(s) immediately.
2. If the DXE is operating on a full-duplex channel transmitting a UI or S frame when it receives a REJ frame, it may finish sending the UI or S frame and then retransmit the I frame(s).
3. If the DXE is operating in a full-duplex channel transmitting another I frame when it receives a REJ frame, it may abort the I frame it was sending and start retransmission of the requested I frames immediately.
4. The DXE may send just the one I frame outstanding, or it may send more than the one indicated if more I frames followed the first one not acknowledged, provided the total to be sent does not exceed the flow-control window (7 frames).

If the DXE receives a REJ frame with the poll bit set, it should respond with either a RR or RNR frame with the final bit set before retransmitting the outstanding I frame(s).

A.4.4.7 Receiving a RNR Frame

Whenever a DXE receives a RNR frame, it shall stop transmission of I frames until the busy condition has been cleared. If timer T1 runs out after the RNR was received, the waiting acknowledgement procedure listed in A.4.4.9, below, should be performed. The poll bit may be used in conjunction with S frames to test for a change in the condition of the busied-out DXE.

A.4.4.8 Sending a Busy Indication

Whenever a DXE enters a busy condition, it will indicate this by sending a RNR response at the next opportunity. While the DXE is in the busy condition, it may receive and process S frames, and if a received S frame has the P bit set to one, the DXE should send a RNR frame with the F bit set to one at the next possible opportunity. To clear the busy condition, the DXE should send either a RR or REJ frame with the received sequence number equal to the current receive state variable, depending on whether the last received I frame was properly received or not.

A.4.4.9 Waiting Acknowledgement

If timer T1 runs out waiting the acknowledgement from the other DXE for an I frame transmitted, the DXE will restart timer T1 and transmit an appropriate supervisory command frame (RR or RNR) with the P bit set. If the DXE receives correctly a supervisory response frame with the F bit set and with an N(R) within the range from the last N(R) received to the last N(S) sent plus one, the DXE will restart timer T1 and will set its send state variable V(S) to the received N(R). It may then resume with I frame transmission or retransmission, as appropriate. If, on the other hand, the DXE receives correctly a supervisory response frame with the F bit not set, or an I frame or supervisory command frame, and with an N(R) within the range from the last N(R) received to the last N(S) sent plus one, the DXE will not restart timer T1, but will use the received N(R) as an indication of acknowledgement of transmitted I frames up to and including I frame numbered N(R)-1.

If timer T1 runs out before a supervisory response frame with the F bit set is received, the DXE will retransmit an appropriate supervisory command frame (RR or RNR) with the P bit set. After N2 attempts to get a supervisory response frame with the F bit set from the other DXE, the DXE will initiate a link resetting procedure as described in A.4.6, below.

A.4.5 Frame Rejection Conditions

A DXE shall initiate the frame-reset procedure when a frame is received with the correct FCS and address field during the information-transfer state with one or more of the conditions in A.3.4.3.3, above.

Under these conditions, the DXE will ask the other DXE to reset the link by transmitting a FRMR response as outlined in A.4.6.3, below.

After sending the FRMR frame, the sending DXE will enter the frame reject condition. This condition is cleared when the DXE that sent the FRMR frame receives a SABM or DISC command, or a DM response frame. Any other command received while the DXE is in the frame reject state will cause another FRMR to be sent out with the same information field as originally sent.

In the frame rejection condition, additional I frames will not be transmitted, and received I frames and S frames will be discarded by the DXE.

The DXE that sent the FRMR frame shall start the T1 timer when the FRMR is sent. If no SABM or DISC frame is received before the timer runs out, the FRMR frame shall be retransmitted, and the T1 timer restarted as described in the waiting acknowledgement section (A.4.4.9) above. If the FRMR is sent N2 times without success, the link shall be reset.

A.4.6 Resetting Procedure

A.4.6.1

The resetting procedure is used to initialize both directions of data flow after a nonrecoverable error has occurred. This resetting procedure is used in the information- transfer state of an AX.25 link only.

A.4.6.2

A DXE shall initiate a reset procedure whenever it receives an unexpected UA response frame or an unsolicited response frame with the F bit set to one. A DXE may also initiate the reset procedure upon receipt of a FRMR frame. Alternatively, the DXE may respond to a FRMR by terminating the connection with a DISC frame.

A.4.6.3

A DXE shall reset the link by sending a SABM frame and starting timer T1. Upon receiving a SABM frame from the DXE previously connected to, the receiver of a SABM frame should send a UA frame back at the earliest opportunity, set its send and receive state variables, V(S) and V(R), to zero and stop T1 unless it has sent a SABM or DISC itself. If the UA is correctly received by the initial DXE, it resets its send and receive state variables, V(S) and V(R), and stops timer T1. Any busy condition that previously existed will also be cleared.

If a DM response is received, the DXE will enter the disconnected state and stop timer T1. If timer T1 runs out before a UA or DM response frame is received, the SABM will be retransmitted and timer T1 restarted. If timer T1 runs out N2 times, the DXE will enter the disconnected state, and any previously existing link conditions will be cleared.

Other commands or responses received by the DXE before completion of the reset procedure will be discarded.

A.4.6.4

One DXE may request that the other DXE reset the link by sending a DM response frame. After the DM frame is sent, the sending DXE will then enter the disconnected state.

A.4.7 List of System Defined Parameters

A.4.7.1 Timers

To maintain the integrity of the AX.25 level 2 connection, use of these timers is recommended.

A.4.7.1.1 Acknowledgement Timer T1

The first timer, T1, is used to make sure a DXE doesn't wait forever for a response to a frame it sends. This timer cannot be expressed in absolute time, since the time required to send frames varies greatly with the signaling rate used at level 1. T1 should take at least twice the amount of time it would take to send maximum length frame to the other DXE, and get the proper response frame back from the other DXE. This would allow time for the other DXE to do some processing before responding.

If level 2 repeaters are to be used, the value of T1 should be adjusted according to the number of repeaters the frame is being transferred through.

A.4.7.1.2 Response Delay Timer T2

The second timer, T2, may be implemented by the DXE to specify a maximum amount of delay to be introduced between the time an I frame is received, and the time the resulting response frame is sent. This delay may be introduced to allow a receiving DXE to wait a short period of time to determine if there is more than one frame being sent to it. If more frames are received, the DXE can acknowledge them at once (up to seven), rather than acknowledge each individual frame. The use of timer T2 is not mandatory, but is recommended to improve channel efficiency. Note that, on full-duplex channels, acknowledgements should not be delayed beyond k/2 frames to achieve maximum throughput. The k parameter is defined in A.4.7.4, below.

A.4.7.1.3 Inactive Link Timer T3

The third timer, T3, is used whenever T1 isn't running to maintain link integrity. It is recommended that whenever there are no outstanding unacknowledged I frames or P-bit frames (during the information-transfer state), a RR or RNR frame with the P bit set to one be sent every T3 time units to query the status of the other DXE. The period of T3 is locally defined, and depends greatly on level 1 operation. T3 should be greater than T1, and may be very large on channels of high integrity.

A.4.7.2 Maximum Number of Retries (N2)

The maximum number of retries is used in conjunction with the T1 timer.

A.4.7.3 Maximum Number of Octets in an I Field (N1)

The maximum number of octets allowed in the I field will be 256. There shall also be an integral number of octets.

A.4.7.4 Maximum Number of I Frames Outstanding (k)

The maximum number of outstanding I frames at a time is seven.

This document was originally downloaded from the TAPR archives and FTP site, as *ax25.doc*. HTML markup was done by Bill Buthod, N5RRS. Last updated: 27 Dec 1997.

APPENDIX B – KISS TNC SPECIFICATION

The KISS TNC: A simple Host-to-TNC communications protocol

Mike Chepponis, K3MC
Phil Karn, KA9Q

Presented at the ARRL 6th Computer Networking Conference, Redondo Beach CA, 1987.
Translated to HTML by KA9Q, January 1997.

ABSTRACT

The KISS ("Keep It Simple, Stupid") TNC provides direct computer to TNC communication using a simple protocol described here. Many TNCs now implement it, including the TAPR TNC-1 and TNC-2 (and their clones), the venerable VADCG TNC, the AEA PK-232/PK-87 and all TNCs in the Kantronics line. KISS has quickly become the protocol of choice for TCP/IP operation and multi-connect BBS software.

1. Introduction

Standard TNC software was written with human users in mind; unfortunately, commands and responses well suited for human use are ill-adapted for host computer use, and vice versa. This is especially true for multi-user servers such as bulletin boards which must multiplex data from several network connections across a single host/TNC link. In addition, experimentation with new link level protocols is greatly hampered because there may very well be no way at all to generate or receive frames in the desired format without reprogramming the TNC.

The KISS TNC solves these problems by eliminating as much as possible from the TNC software, giving the attached host complete control over and access to the contents of the HDLC frames transmitted and received over the air. This is central to the KISS philosophy: the host software should have control over all TNC functions at the lowest possible level.

The AX.25 protocol is removed entirely from the TNC, as are all command interpreters and the like. The TNC simply converts between synchronous HDLC, spoken on the full- or half-duplex radio channel, and a special asynchronous, full duplex frame format spoken on the host/TNC link. Every frame received on the HDLC link is passed intact to the host once it has been translated to the asynchronous format; likewise, asynchronous frames from the host are transmitted on the radio channel once they have been converted to HDLC format.

Of course, this means that the bulk of AX.25 (or another protocol) must now be implemented on the host system. This is acceptable, however, considering the greatly increased flexibility and reduced overall complexity that comes from allowing the protocol to reside on the same machine with the applications to which it is closely coupled.

It should be stressed that the KISS TNC was intended only as a stopgap. Ideally, host computers would have HDLC interfaces of their own, making separate TNCs unnecessary. Unfortunately, HDLC interfaces are rare, although they are starting to appear for the IBM PC. The KISS TNC therefore becomes the "next best thing" to a real HDLC interface, since the host computer only needs an ordinary asynchronous interface.

2. Asynchronous Frame Format

The "asynchronous packet protocol" spoken between the host and TNC is very simple, since its only function is to delimit frames. Each frame is both preceded and followed by a special FEND (Frame End) character, analogous to an HDLC flag. No CRC or checksum is provided. In addition, no RS-232C handshaking signals are employed.

The special characters are:

Abbreviation	Description	Hex value
FEND	Frame End	C0
FESC	Frame Escape	DB
TFEND	Transposed Frame End	DC
TFESC	Transposed Frame Escape	DD

The reason for both preceding and ending frames with FENDs is to improve performance when there is noise on the asynch line. The FEND at the beginning of a frame serves to "flush out" any accumulated garbage into a separate frame (which will be discarded by the upper layer protocol) instead of sticking it on the front of an otherwise good frame. As with back-to-back flags in HDLC, two FEND characters in a row should not be interpreted as delimiting an empty frame.

3. Transparency

Frames are sent in 8-bit binary; the asynchronous link is set to 8 data bits, 1 stop bit, and no parity. If a FEND ever appears in the data, it is translated into the two byte sequence FESC TFEND (Frame Escape, Transposed Frame End). Likewise, if the FESC character ever appears in the user data, it is replaced with the two character sequence FESC TFESC (Frame Escape, Transposed Frame Escape).

As characters arrive at the receiver, they are appended to a buffer containing the current frame. Receiving a FEND marks the end of the current frame. Receipt of a FESC puts the receiver into "escaped mode", causing the receiver to translate a following TFESC or TFEND back to FESC or FEND, respectively, before adding it to the receive buffer and leaving escaped mode. Receipt of any character other than TFESC or TFEND while in escaped mode is an error; no action is taken and frame assembly continues. A TFEND or TESC received while not in escaped mode is treated as an ordinary data character.

This procedure may seem somewhat complicated, but it is easy to implement and recovers quickly from errors. In particular, the FEND character is never sent over the channel except as an actual end-of-frame indication. This ensures that any intact frame (properly delimited by FEND characters) will always be received properly regardless of the starting state of the receiver or corruption of the preceding frame.

This asynchronous framing protocol is identical to "SLIP" (Serial Line IP), a popular method for sending ARPA IP datagrams across asynchronous links. It could also form the basis of an asynchronous amateur packet radio link protocol that avoids the complexity of HDLC on slow speed channels.

4. Control of the KISS TNC

Each asynchronous data frame sent to the TNC is converted back into "pure" form and queued for transmission as a separate HDLC frame. Although removing the human interface and the AX.25 protocol from the TNC makes most existing TNC commands unnecessary (i.e., they become host functions), the TNC is still responsible for keying the transmitter's PTT line and deferring to other activity on the radio channel. It is therefore necessary to allow the host to control a few TNC pa-

rameters, namely the transmitter keyup delay, the transmitter persistence variables and any special hardware that a particular TNC may have.

To distinguish between command and data frames on the host/TNC link, the first byte of each asynchronous frame between host and TNC is a "type" indicator. This type indicator byte is broken into two 4-bit nibbles so that the low-order nibble indicates the command number (given in the table below) and the high-order nibble indicates the port number for that particular command. In systems with only one HDLC port, it is by definition Port 0. In multi-port TNCs, the upper 4 bits of the type indicator byte can specify one of up to sixteen ports. The following commands are defined in frames to the TNC (the "Command" field is in hexadecimal):

Command	Function	Comments
0	Data frame	The rest of the frame is data to be sent on the HDLC channel.
1	TXDELAY	The next byte is the transmitter keyup delay in 10 ms units. The default start-up value is 50 (i.e., 500 ms).
2	P	The next byte is the persistence parameter, p, scaled to the range 0 - 255 with the following formula: $P = p \times 256 - 1$ The default value is P = 63 (i.e., p = 0.25).
3	SlotTime	The next byte is the slot interval in 10 ms units. The default is 10 (i.e., 100 ms).
4	TXtail	The next byte is the time to hold up the TX after the FCS has been sent, in 10 ms units. This command is obsolete, and is included here only for compatibility with some existing implementations.
5	FullDuplex	The next byte is 0 for half duplex, nonzero for full duplex. The default is 0 (i.e., half duplex).
6	SetHardware	Specific for each TNC. In the TNC-1, this command sets the modem speed. Other implementations may use this function for other hardware-specific functions.
FF	Return	Exit KISS and return control to a higher-level program. This is useful only when KISS is incorporated into the TNC along with other applications.

The following types are defined in frames to the host:

Command	Function	Comments
0	Data frame	Rest of frame is data from the HDLC channel.

No other types are defined; in particular, there is no provision for acknowledging data or command frames sent to the TNC. KISS implementations must ignore any unsupported command types. All KISS implementations must implement commands 0, 1, 2, 3 and 5; the others are optional.

5. Buffer and Packet Size Limits

One of the things that makes the KISS TNC simple is the deliberate lack of TNC/host flow control. The host computers run a higher level protocol (typically TCP, but AX.25 in the connected mode also qualifies) that handles flow control on an end-to-end basis. Ideally, the TNC would always have more buffer memory than the sum of all the flow control windows of all of the logical connections using it at that moment. This would allow for the worst case (i.e., all users sending simultaneously). In practice, however, many (if not most) user connections are idle for long periods of time,

so buffer memory may be safely "overbooked". When the occasional "bump" occurs, the TNC must drop the packet gracefully, i.e., ignore it without crashing or losing packets already queued. The higher level protocol is expected to recover by "backing off" and retransmitting the packet at a later time, just as it does whenever a packet is lost in the network for any other reason. As long as this occurs infrequently, the performance degradation is slight; therefore the TNC should provide as much packet buffering as possible, limited only by available RAM.

Individual packets at least 1024 bytes long should be allowed. As with buffer queues, it is recommended that no artificial limits be placed on packet size. For example, the K3MC code running on a TNC-2 with 32K of RAM can send and receive 30K byte packets, although this is admittedly rather extreme. Large packets reduce protocol overhead on good channels. They are essential for good performance when operating on high speed modems such as the new WA4DSY 56 kbps design.

6. Persistence

The P and SlotTime parameters are used to implement true p-persistent CSMA. This works as follows:

Whenever the host queues data for transmission, the TNC begins monitoring the carrier detect signal from the modem. It waits indefinitely for this signal to go inactive. When the channel clears, the TNC generates a random number between 0 and 1. If this number is less than or equal to the parameter p, the TNC keys the transmitter, waits .01 × TXDELAY seconds, and transmits all queued frames. The TNC then unkeys the transmitter and goes back to the idle state. If the random number is greater than p, the TNC delays .01 × SlotTime seconds and repeats the procedure beginning with the sampling of the carrier detect signal. (If the carrier detect signal has gone active in the meantime, the TNC again waits for it to clear before continuing). Note that p = 1 means "transmit as soon as the channel clears"; in this case the p-persistence algorithm degenerates into the 1-persistent CSMA generally used by conventional AX.25 TNCs.

p-persistence causes the TNC to wait for an exponentially-distributed random interval after sensing that the channel has gone clear before attempting to transmit. With proper tuning of the parameters p and SlotTime, several stations with traffic to send are much less likely to collide with each other when they all see the channel go clear. One transmits first and the others see it in time to prevent a collision, and the channel To conform to the literature, here p takes on values between 0 to 1. However, fractions are difficult to use in a fixed point microprocessor so the KISS TNC actually works with P values that are rescaled to the range 0 to 255. To avoid confusion, we will use lower-case p to mean the former (0-1) and upper-case P whenever we mean the latter (0-255). remains stable under heavy load.

We believe that optimum p and SlotTime values could be computed automatically. This could be done by noting the channel occupancy and the length of the frames on the channel. We are proceeding with a simulation of the p-persistence algorithm described here that we hope will allow us to construct an automatic algorithm for p and SlotTime selection.

We added p-persistence to the KISS TNC because it was a convenient opportunity to do so. However, it is not inherently associated with KISS nor with new protocols such as TCP/IP. Rather, persistence is a channel access protocol that can yield dramatic performance improvements regardless of the higher level protocol in use; we urge it be added to every TNC, whether or not it supports KISS.

7. Implementation History

The original idea for a simplified host/TNC protocol is due to Brian Lloyd, WB6RQN. Phil Karn, KA9Q, organized the specification and submitted an initial version on 6 August 1986. As of this writing, the following KISS TNC implementations exist:

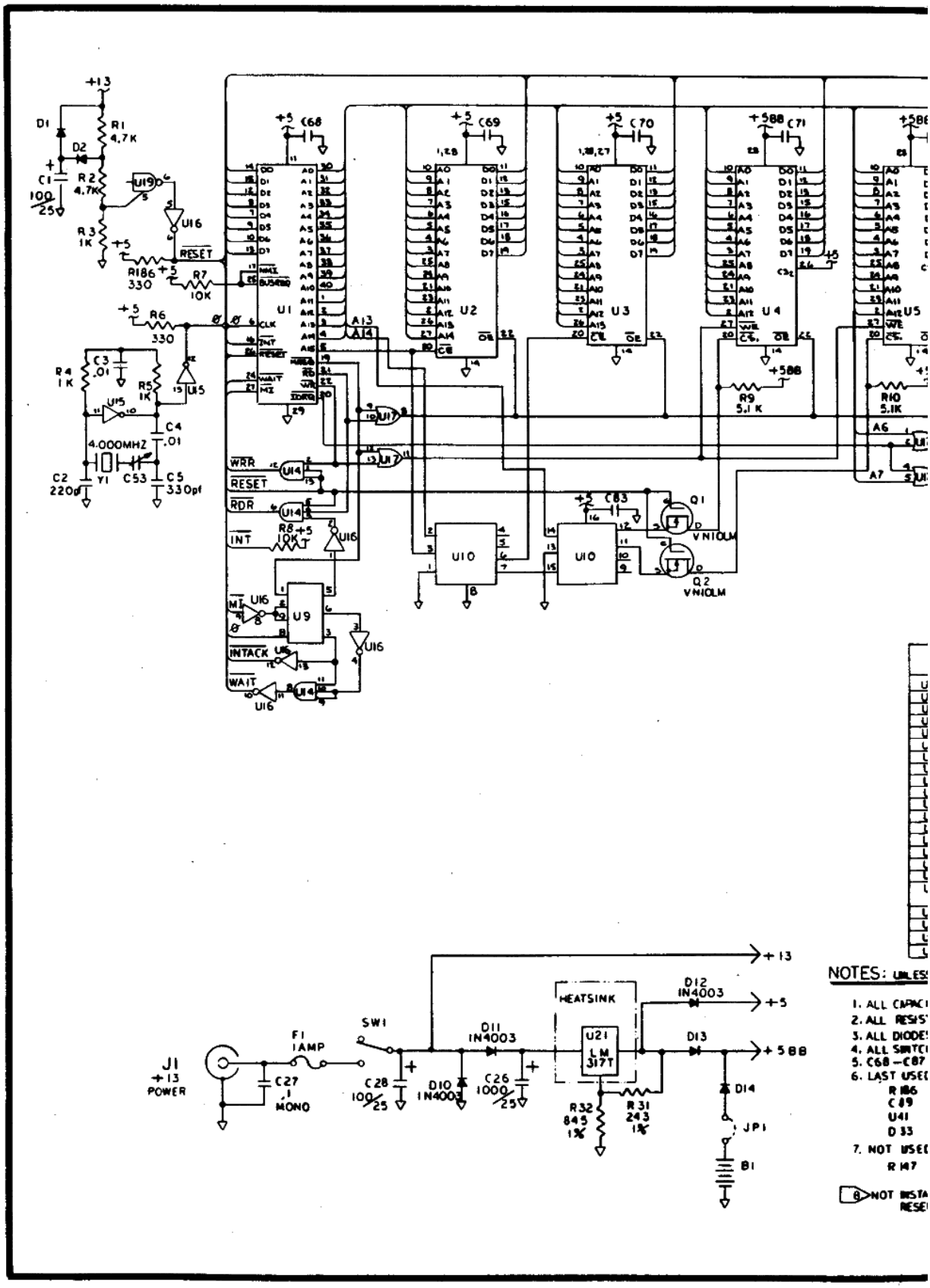
TNC type	Author	Comments
TAPR TNC-2 & clones	Mike Chepponis, K3MC	First implementation, most widely used. Exists in both downloadable and dedicated ROM.
TAPR TNC-1 & clones	Marc Kaufman, WB6ECE	Both download and dedicated ROM.
VADCG TNC	Mike Bruski, AJ9X	Dedicated ROM.
AEA PK-232 & PK-87	Steve Stuart, N6IA	Integrated into standard AEA firmware as of 21 Jan 1987.

The special commands "KISS ON" and "KISS OFF" (!) control entry into KISS mode.

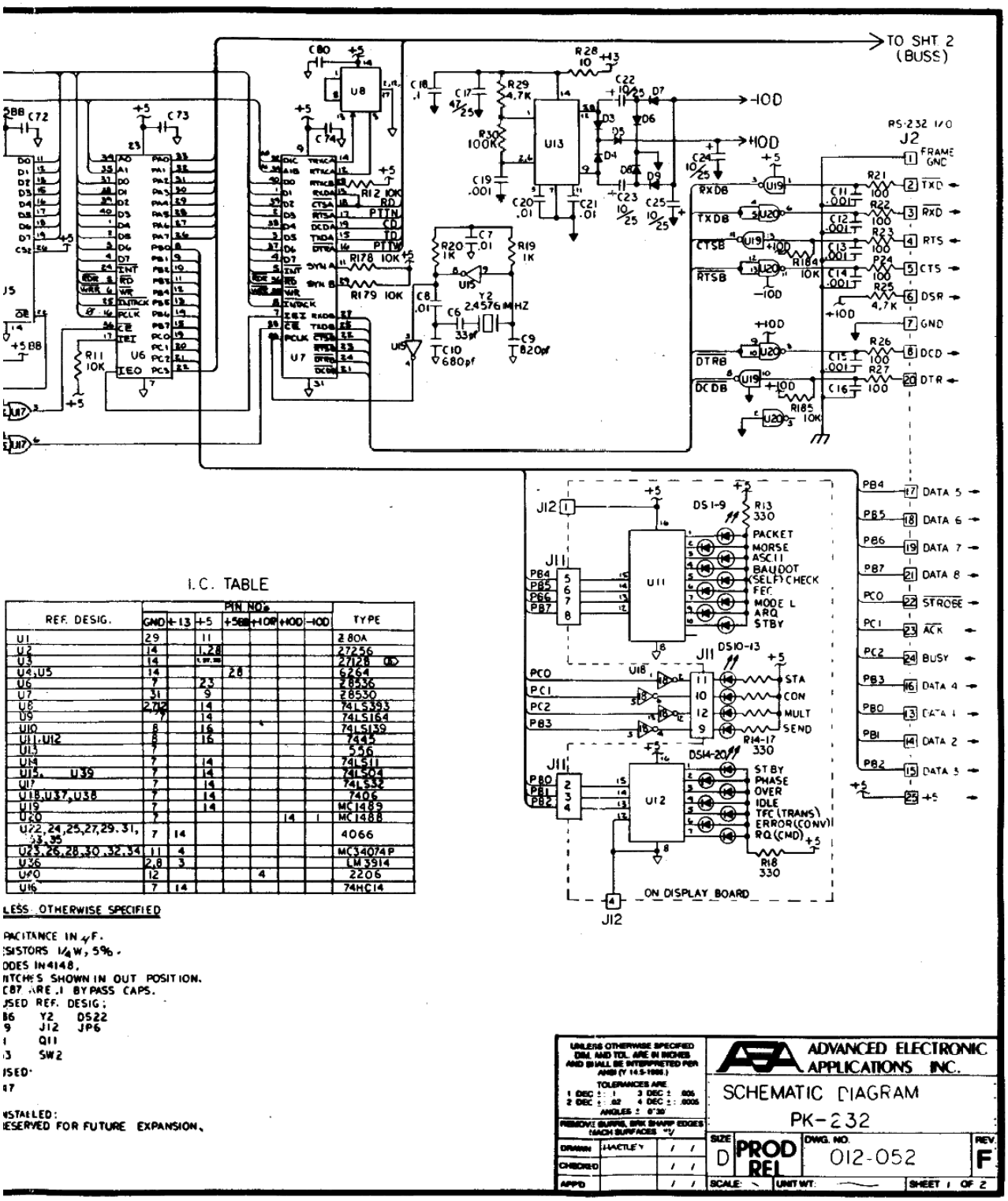
TNC type	Author	Comments
Kantronics	Mike Huslig	Integrated into standard Kantronics firmware as of July 1987.

The AEA and Kantronics implementations are noteworthy in that the KISS functions were written by those vendors and integrated into their standard TNC firmware. Their TNCs can operate in either KISS or regular AX.25 mode without ROM changes. Since the TNC-1 and TNC-2 KISS versions were written by different authors than the original AX.25 firmware, and because the original source code for those TNCs was not made available, running KISS on these TNCs requires the installation of nonstandard ROMs. Two ROMs are available for the TNC-2. One contains "dedicated" KISS TNC code; the TNC operates only in the KISS mode. The "download" version contains standard N2WX firmware with a bootstrap loader overlay. When the TNC is turned on or reset, it executes the loader. The loader will accept a memory image in Intel Hex format, or it can be told to execute the standard N2WX firmware through the "H" command. The download version is handy for occasional KISS operation, while the dedicated version is much more convenient for full-time or demo KISS operation.

The code for the TNC-1 is also available in both download and dedicated versions. However, at present the download ROM contains only a bootstrap; the original ROMs must be put back in to run the original TNC software.


+13
D1
R1 4.7K
D2
C1 100/25
R2 4.7K
U19
U16
R3 1K
+5
RESET
R186 330
R7 10K
R6 330
R4 1K
C3 .01
R5 1K
U15
C4 .01
4.000MHZ
Y1
C53
C2 220pf
C5 330pf
+5 C68
+5 C69
+5 C70
+5 C71
U1
U2
U3
U4
U5
A13
A14
WRR
RESET
RDR
INT
R8 10K
U14
U17
U9
INTACK
WAIT
U10
C83
Q1
VN10LM
Q2
VN10LM
R9 5.1K
R10 5.1K
A6
A7
J1
+13
POWER
C27 .1 MONO
F1 1AMP
SW1
C28 100/25
D10 1N4003
D11 1N4003
C26 1000/25
HEATSINK
U21 LM 317T
R32 845 1%
R31 243 1%
D12 1N4003
D13
D14
JP1
B1
+13
+5
+5 BB
NOTES:
1. ALL CAPAC
2. ALL RESIS
3. ALL DIODE
4. ALL SWITC
5. C68 – C87
6. LAST USED
R186
C89
U41
D33
7. NOT USED
R147
8 NOT INSTA
RESE

H-1



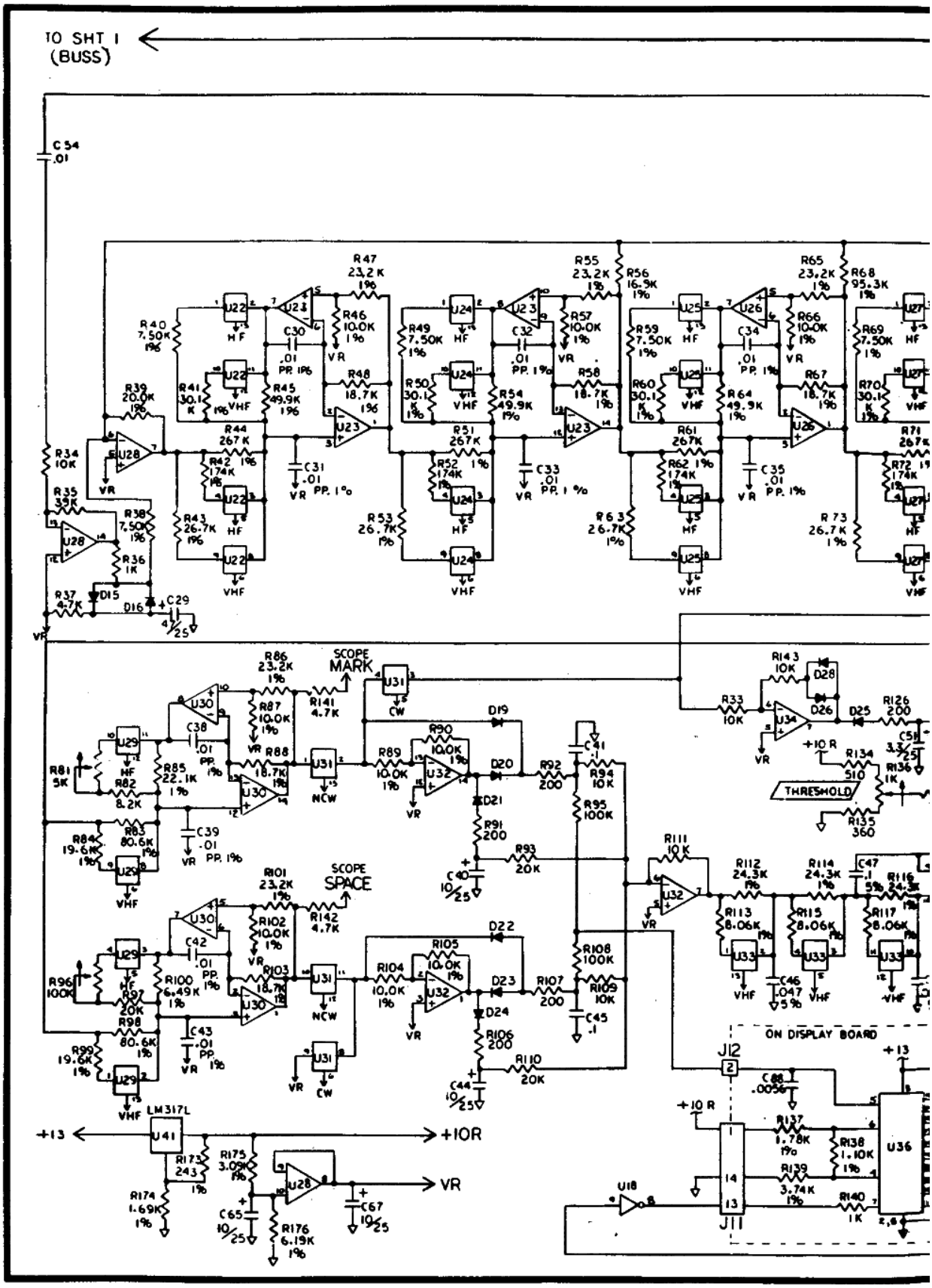
I.C. TABLE

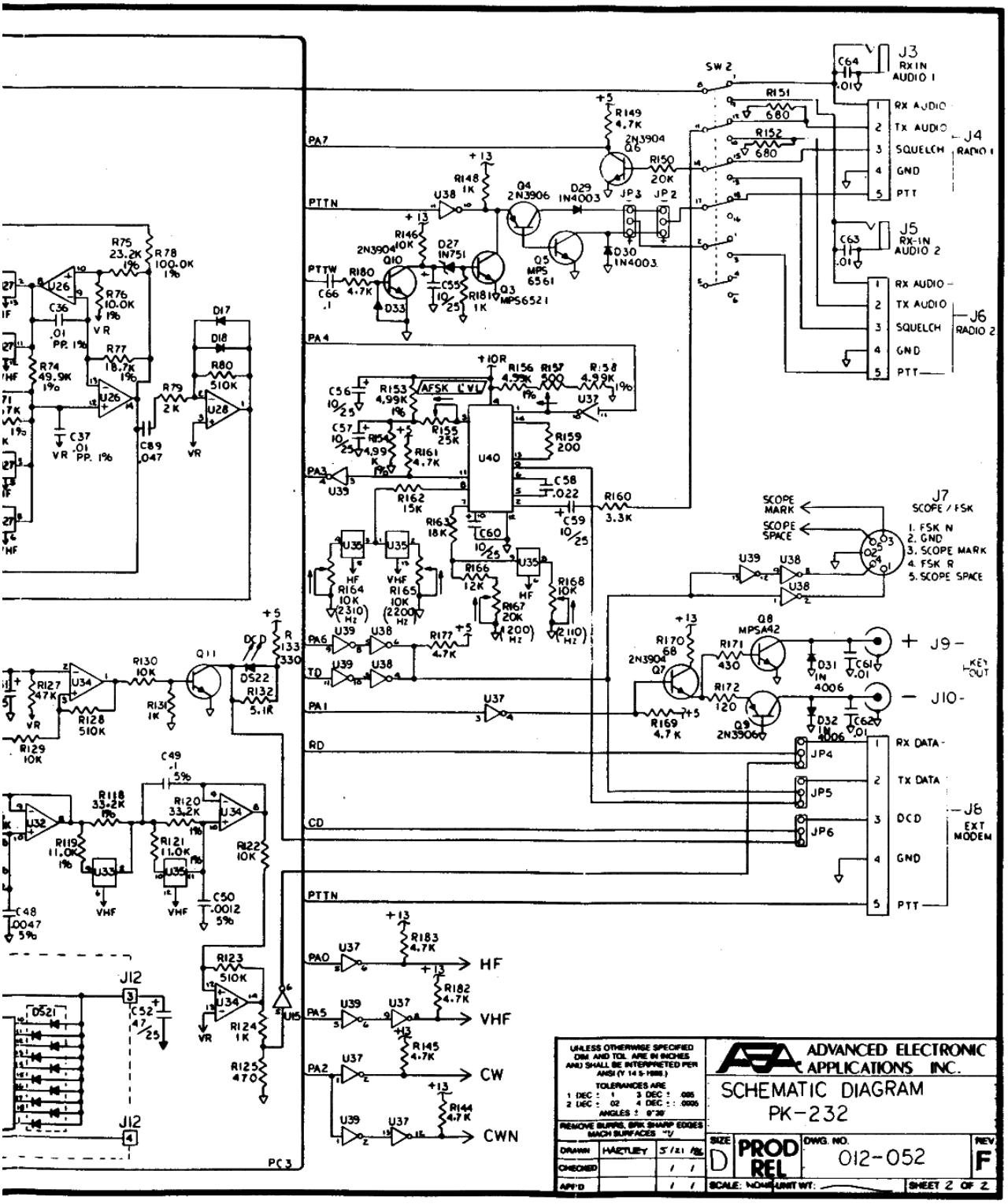
REF. DESIG.	PIN NO. GND	+13	+5	+5GB	+10P	+10D	-10D	TYPE
U1	29		11					280A
U2	14		1,28					27256
U3	14		1,28,30					27128
U4,U5	14			28				6264
U6	7		23					28536
U7	31		9					28530
U8	2,7,12		14					74LS393
U9	7		14					74LS164
U10	8		16					74LS139
U11,U12	8		16					7445
U13	8							556
U14	7		14					74LS11
U15, U39	7		14					74LS04
U17	7		14					74LS32
U18,U37,U38	7		14					7406
U19	7		14					MC1489
U20	7					14	1	MC1488
U22,24,25,27,29,31,33,35	7	14						4066
U23,26,28,30,32,34	11	4						MC34074P
U36	2,8	3						LM3914
U40	12				4			2206
U16	7	14						74HC14

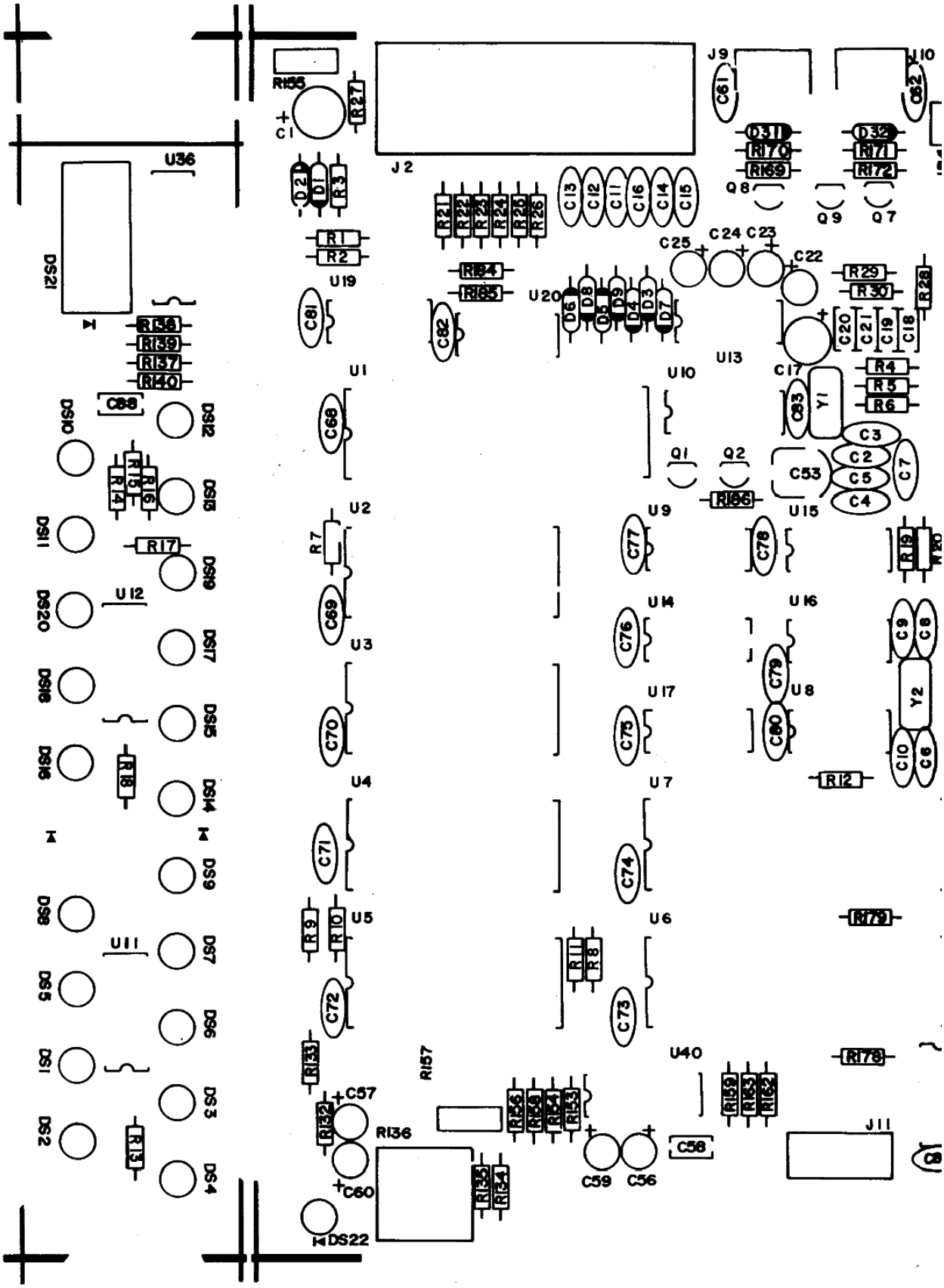
LESS OTHERWISE SPECIFIED

PACITANCE IN µF.
ISISTORS 1/4W, 5%.
ODES IN4148.
NTCHES SHOWN IN OUT POSITION.
C87 ARE .1 BYPASS CAPS.
USED REF. DESIG:
B6 Y2 DS22
9 J12 JP6
1 Q11
3 SW2
ISED:
7

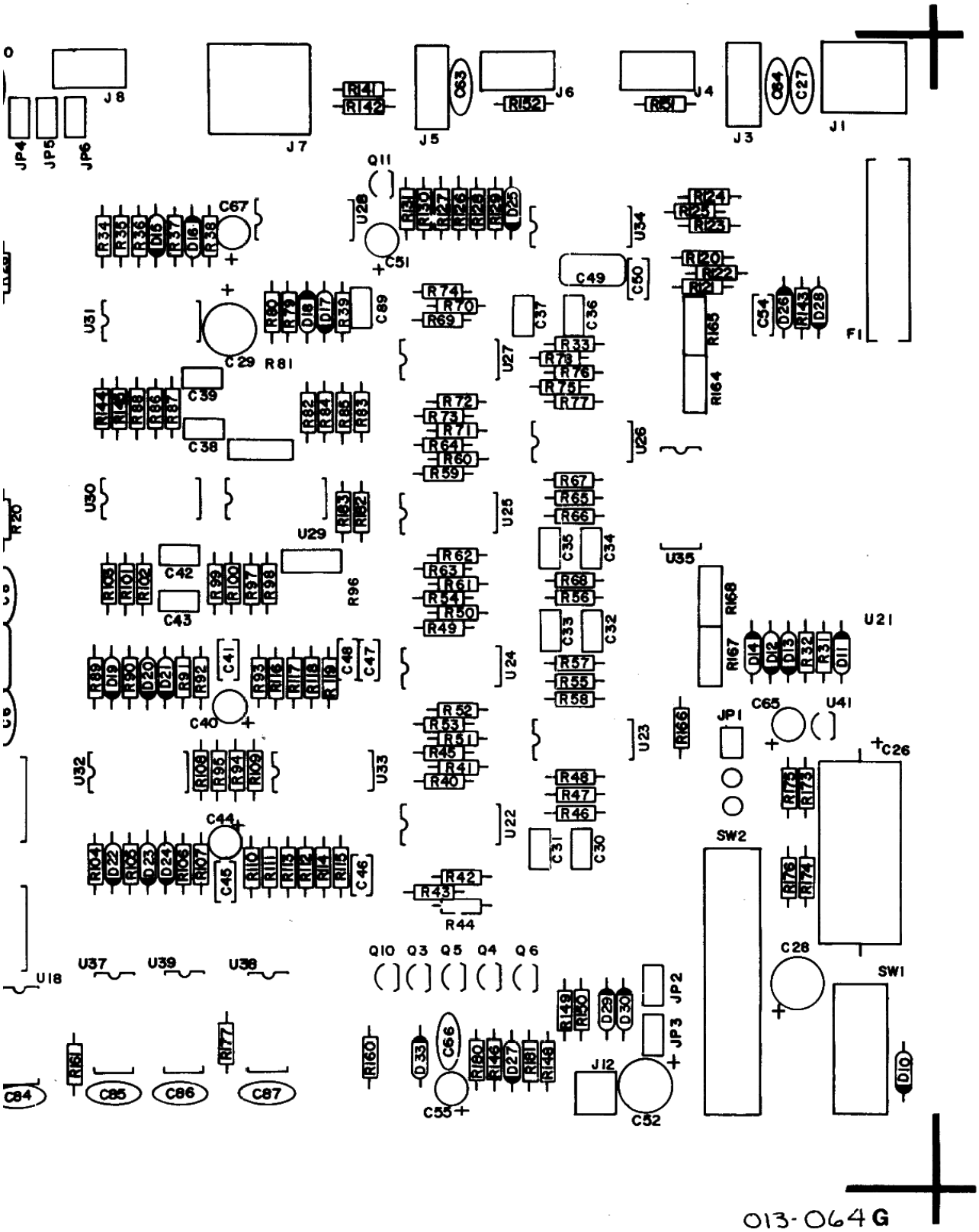
NSTALLED:
ESERVED FOR FUTURE EXPANSION.

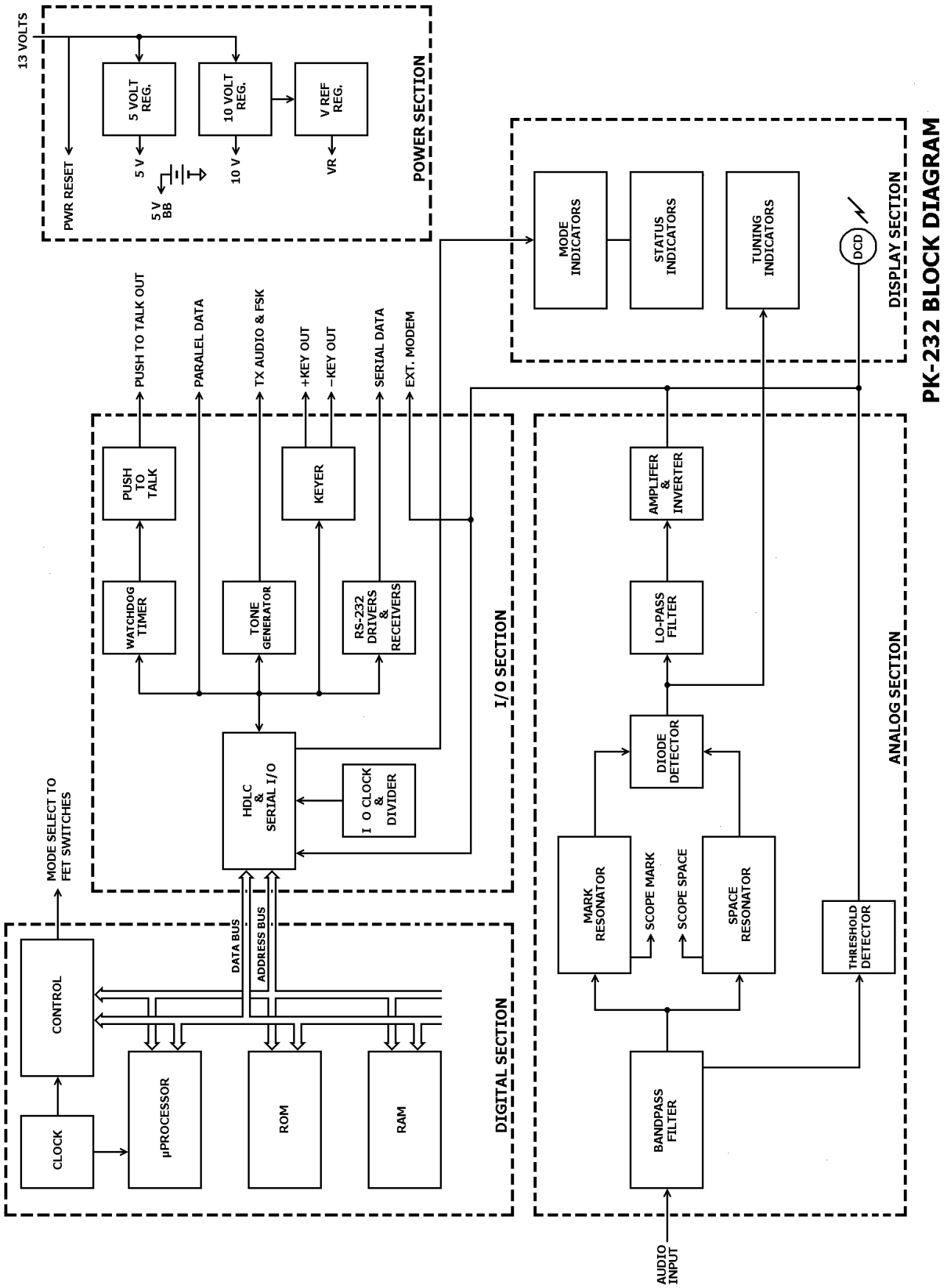

TO SHT 1
(BUSS)
SCOPE
MARK
SCOPE
SPACE
THRESHOLD
ON DISPLAY BOARD
LM317L
+13
+10R
VR
H-2

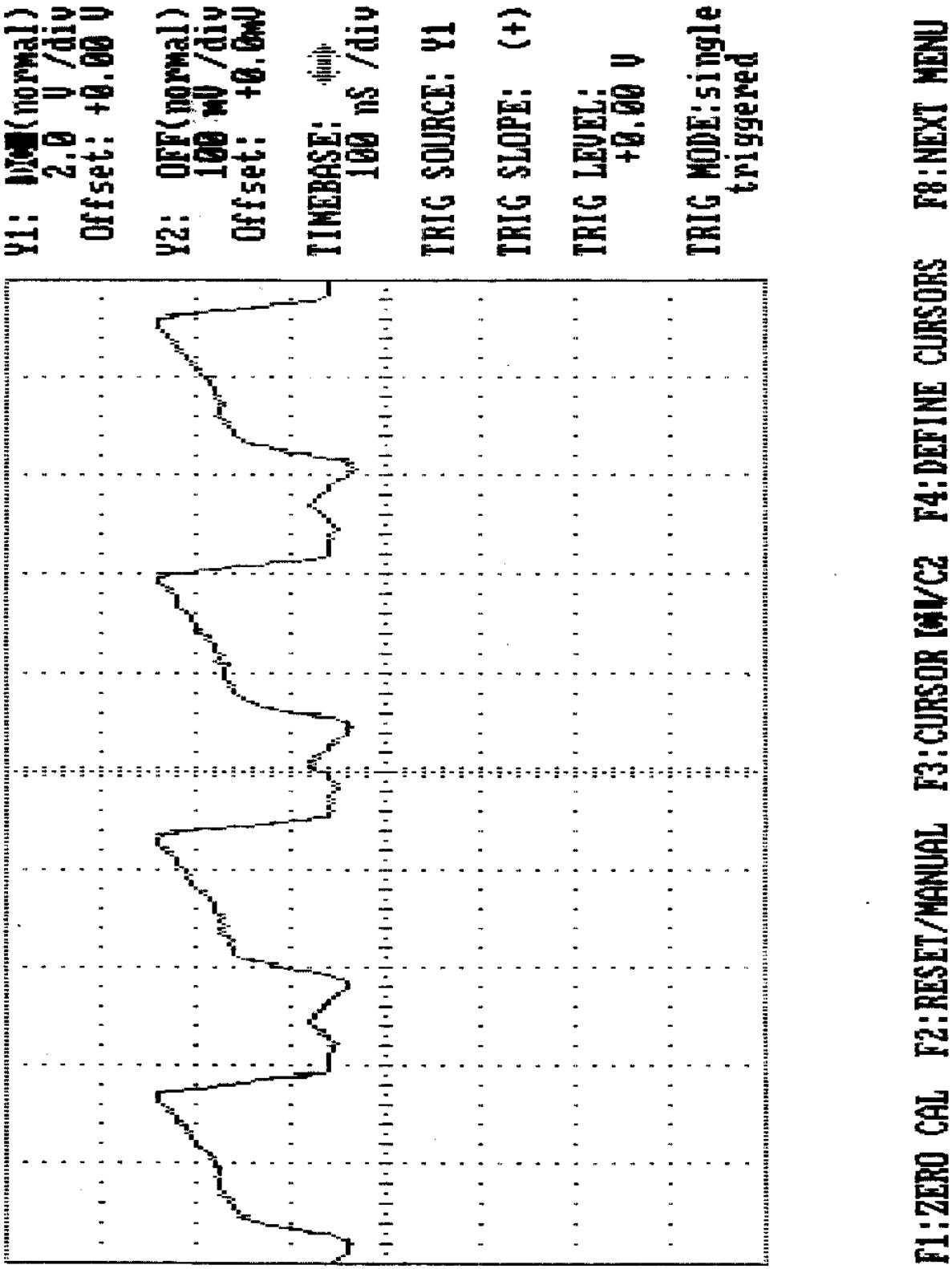

J3
RX IN
AUDIO 1
J4
RADIO 1
J5
RX-IN
AUDIO 2
J6
RADIO 2
RX AUDIO
TX AUDIO
SQUELCH
GND
PTT
SW 2
PA7
PTTN
PTTW
PA4
PA3
PA6
TD
PA1
RD
CD
PA0
PA5
PA2
HF
VHF
CW
CWN
U40
AFSK LVL
J7
SCOPE / FSK
SCOPE MARK
SCOPE SPACE
1. FSK N
2. GND
3. SCOPE MARK
4. FSK R
5. SCOPE SPACE
J9 –
J10 –
KEY OUT
JP4
JP5
JP6
RX DATA
TX DATA
DCD
J8
EXT MODEM
J12
PC3
ADVANCED ELECTRONIC APPLICATIONS INC.
SCHEMATIC DIAGRAM
PK-232
PROD REL
DWG. NO.
012-052
REV.
F
SHEET 2 OF 2

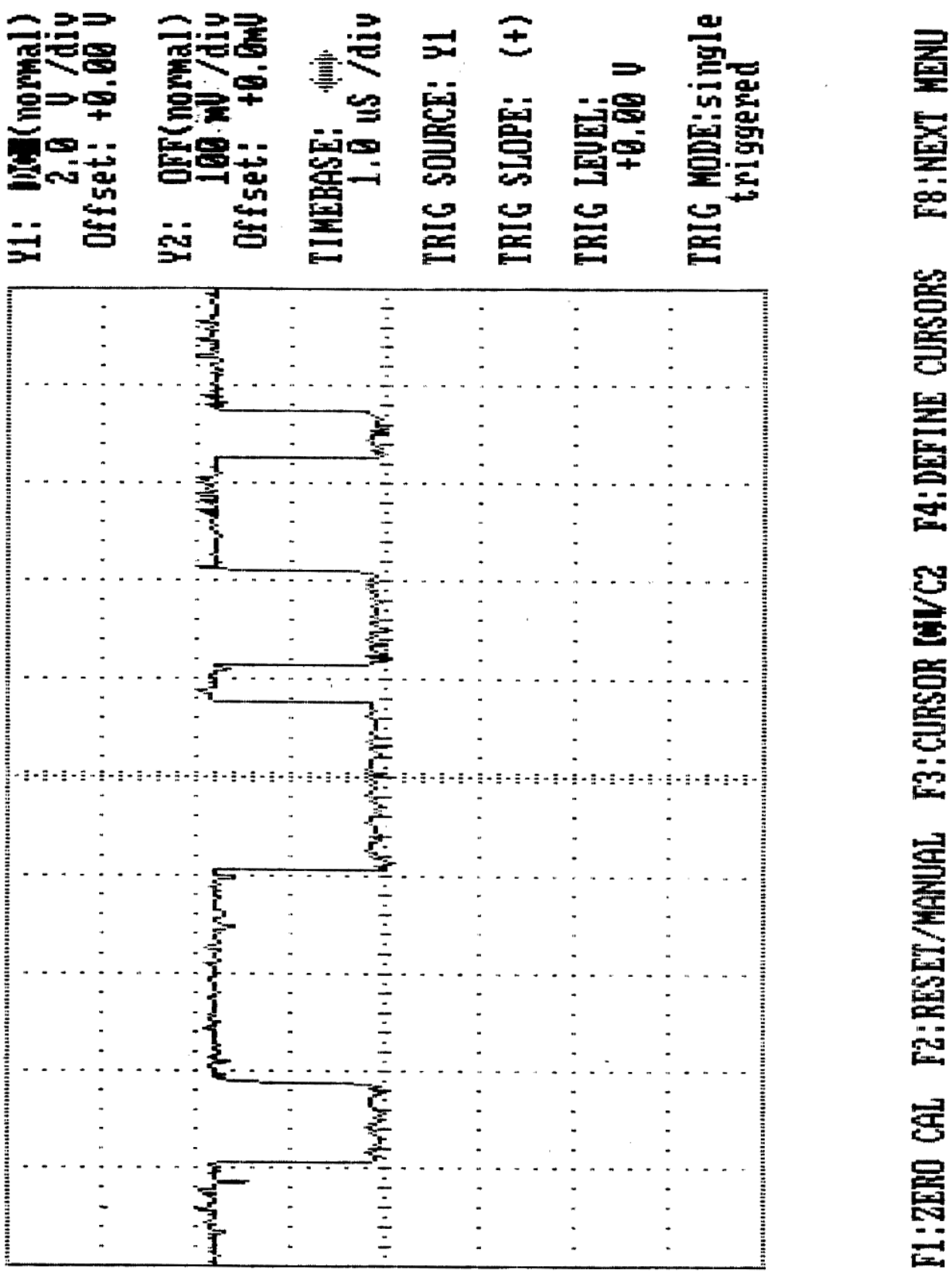

J2
J9
J10
J11
U36
DS21
U19
U20
U13
U1
U10
U2
U9
U15
U12
U3
U14
U16
U17
U8
U4
U7
U5
U6
U11
U40
R155
R27
C1
D2
D1
R3
R1
R2
R21
R22
R23
R24
R25
R26
C13
C12
C11
C16
C14
C15
C61
C62
D31
R170
R169
D32
R171
R172
Q8
Q9
Q7
C25
C24
C23
C22
R29
R30
R28
R184
R185
D6
D8
D5
D9
D4
D3
D7
C81
C82
C20
C21
C19
C18
R138
R139
R137
R140
C88
C68
C17
R4
R5
R6
C83
Y1
C3
C2
C5
C4
C7
C53
Q1
Q2
R186
DS10
DS12
DS13
DS11
DS19
DS20
DS17
DS18
DS15
DS16
DS14
DS9
DS8
DS7
DS5
DS6
DS1
DS3
DS2
DS4
R14
R15
R16
R17
R18
R13
R7
C69
C70
C71
C72
R9
R10
R11
R8
C73
C74
C75
C76
C77
C78
C79
C80
C9
C8
Y2
C10
C6
R12
R179
R178
R133
R157
C57
R132
R136
C60
DS22
R135
R134
R156
R158
R154
R153
C59
C56
C58
R159
R163
R162

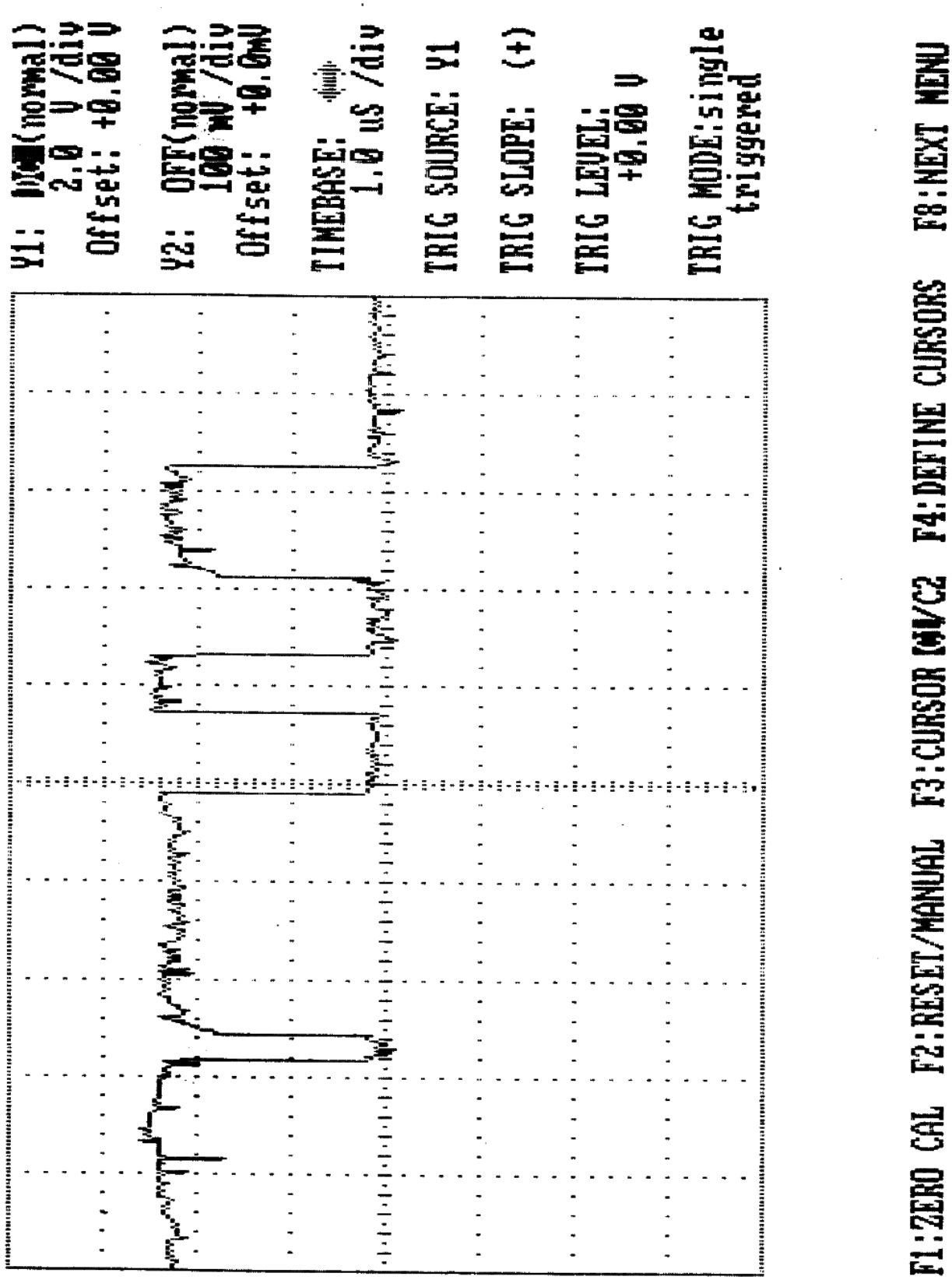
CCMP I.D.

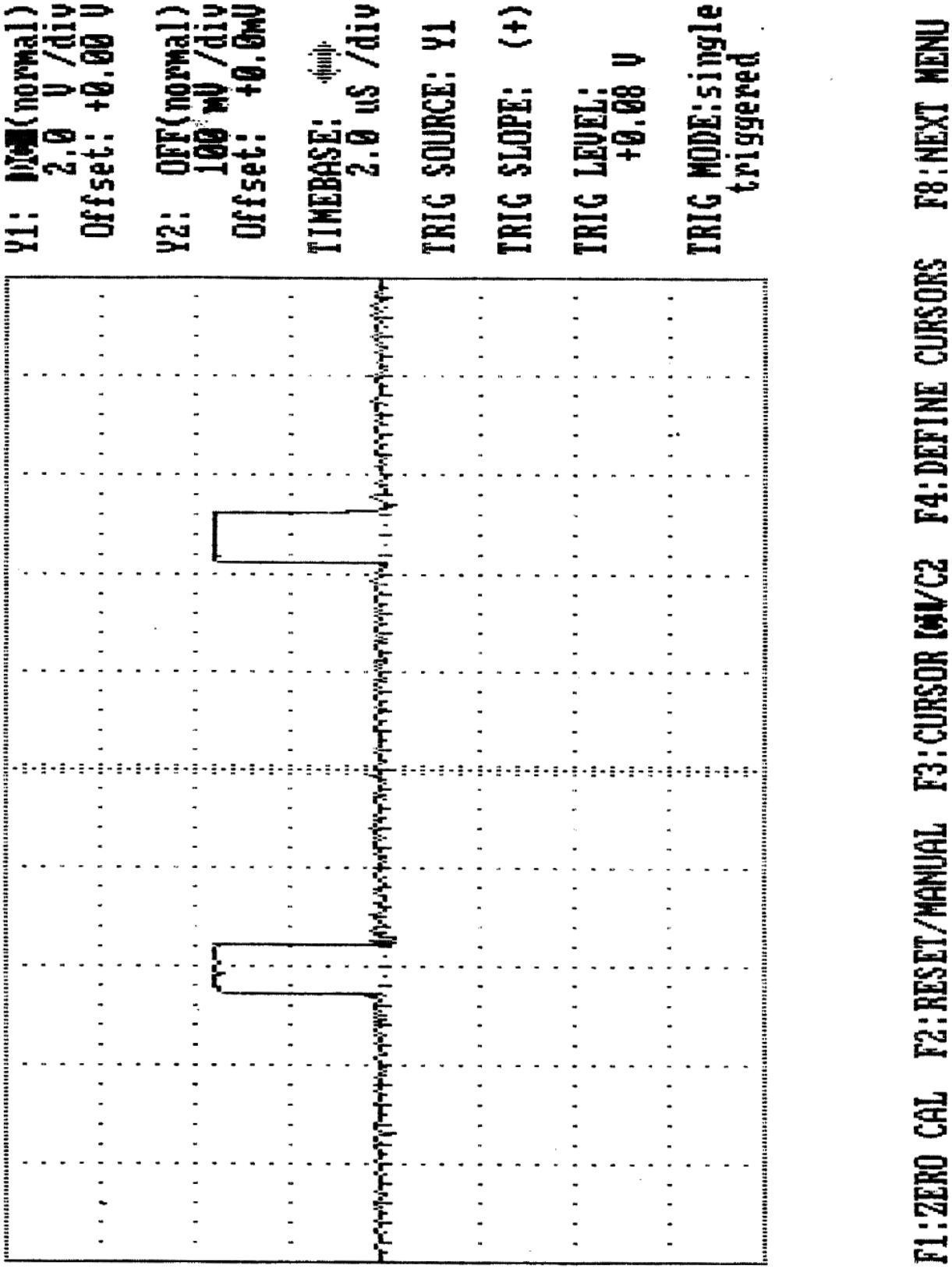

J8
J7
J5
J6
J4
J3
J1
JP4
JP5
JP6
Q11
U28
C67
C51
U34
C49
C50
U31
C29
R81
C89
U27
C37
C36
F1
C39
C38
U26
U30
U29
U25
U35
C42
C43
R96
U21
C41
C40
U24
C33
C32
C65
U41
JP1
C26
U32
U33
U23
C44
U22
C31
C30
SW2
C45
C46
R44
Q10
Q3
Q5
Q4
Q6
U18
U37
U39
U38
C28
SW1
JP2
JP3
J12
C52
C55
C66
C84
C85
C86
C87
013-064G

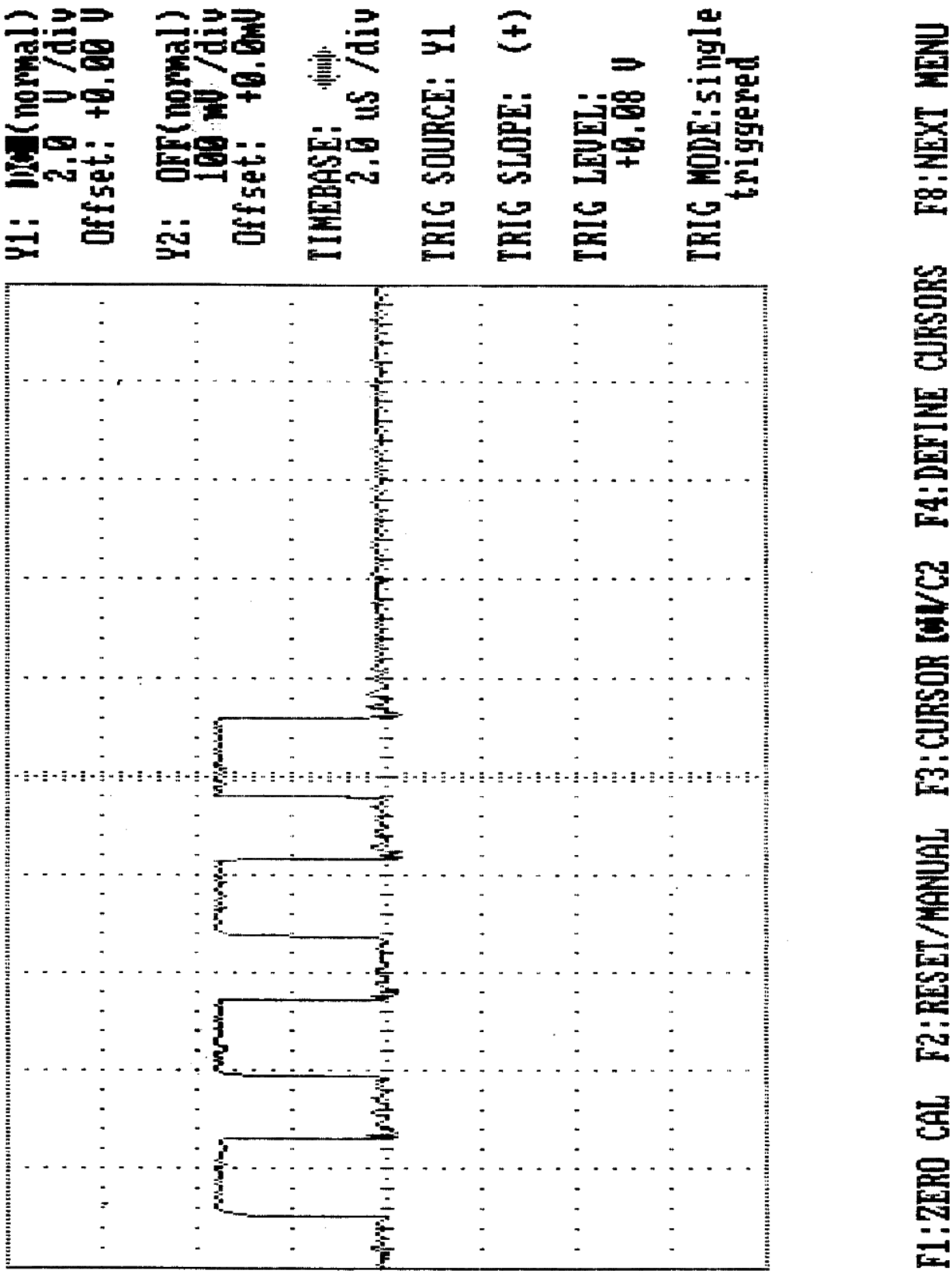

PK-232 BLOCK DIAGRAM
POWER SECTION
13 VOLTS
PWR RESET
5 VOLT REG.
10 VOLT REG.
V REF REG.
5 V
5 V BB
10 V
VR
DISPLAY SECTION
MODE INDICATORS
STATUS INDICATORS
TUNING INDICATORS
DCD
I/O SECTION
PUSH TO TALK OUT
PARALEL DATA
TX AUDIO & FSK
+KEY OUT
–KEY OUT
SERIAL DATA
EXT. MODEM
PUSH TO TALK
KEYER
WATCHDOG TIMER
TONE GENERATOR
RS-232 DRIVERS & RECEIVERS
HDLC & SERIAL I/O
I O CLOCK & DIVIDER
ANALOG SECTION
AMPLIFER & INVERTER
LO-PASS FILTER
DIODE DETECTOR
MARK RESONATOR
SCOPE MARK
SCOPE SPACE
SPACE RESONATOR
THRESHOLD DETECTOR
BANDPASS FILTER
AUDIO INPUT
DIGITAL SECTION
MODE SELECT TO FET SWITCHES
CONTROL
DATA BUS
ADDRESS BUS
CLOCK
µPROCESSOR
ROM
RAM

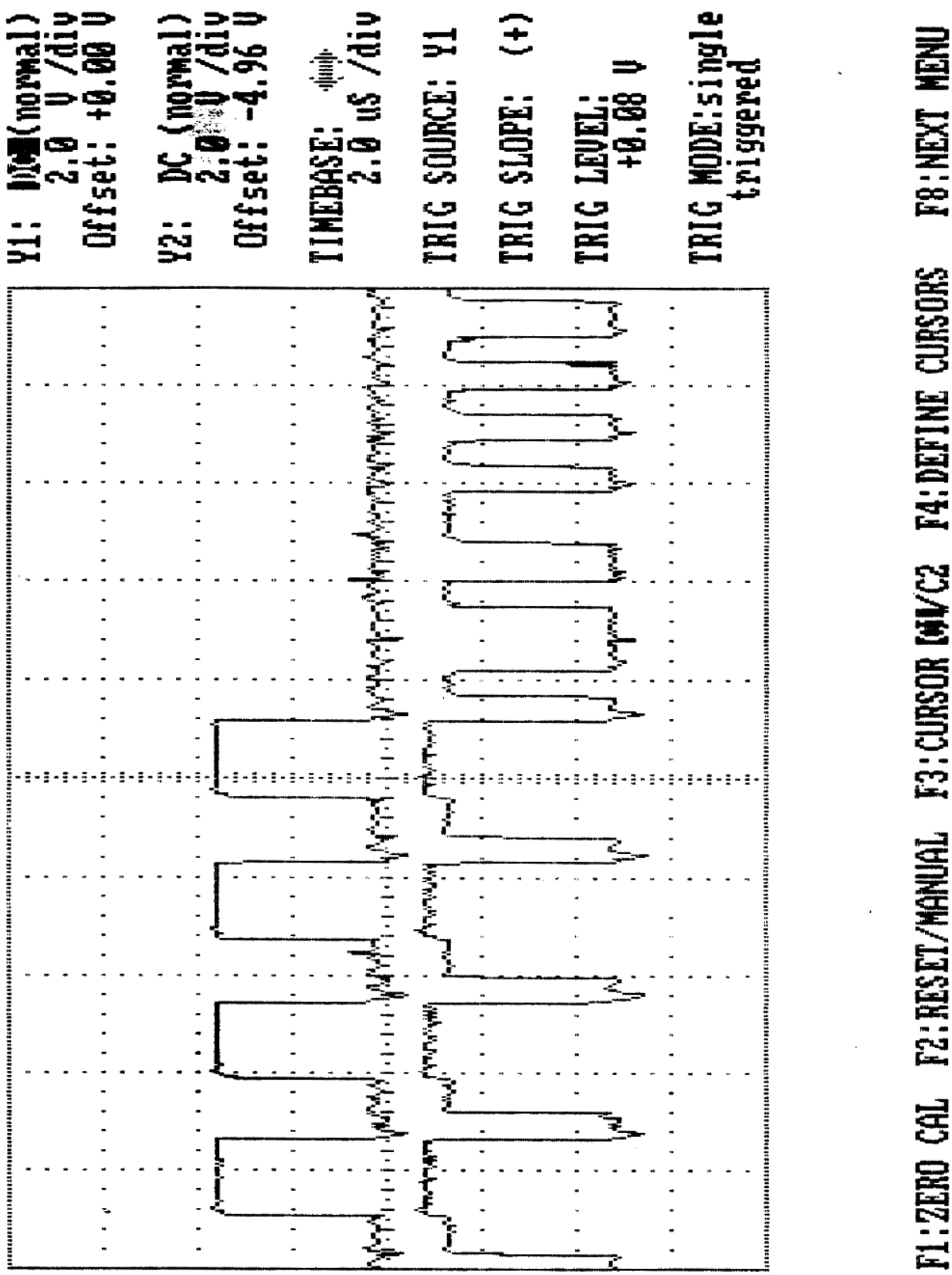

Y1: DCM(normal)
2.0 V /div
Offset: +0.00 V
Y2: OFF(normal)
100 mV /div
Offset: +0.0mV
TIMEBASE:
100 nS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.00 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR C1/C2 F4:DEFINE CURSORS F8:NEXT MENU

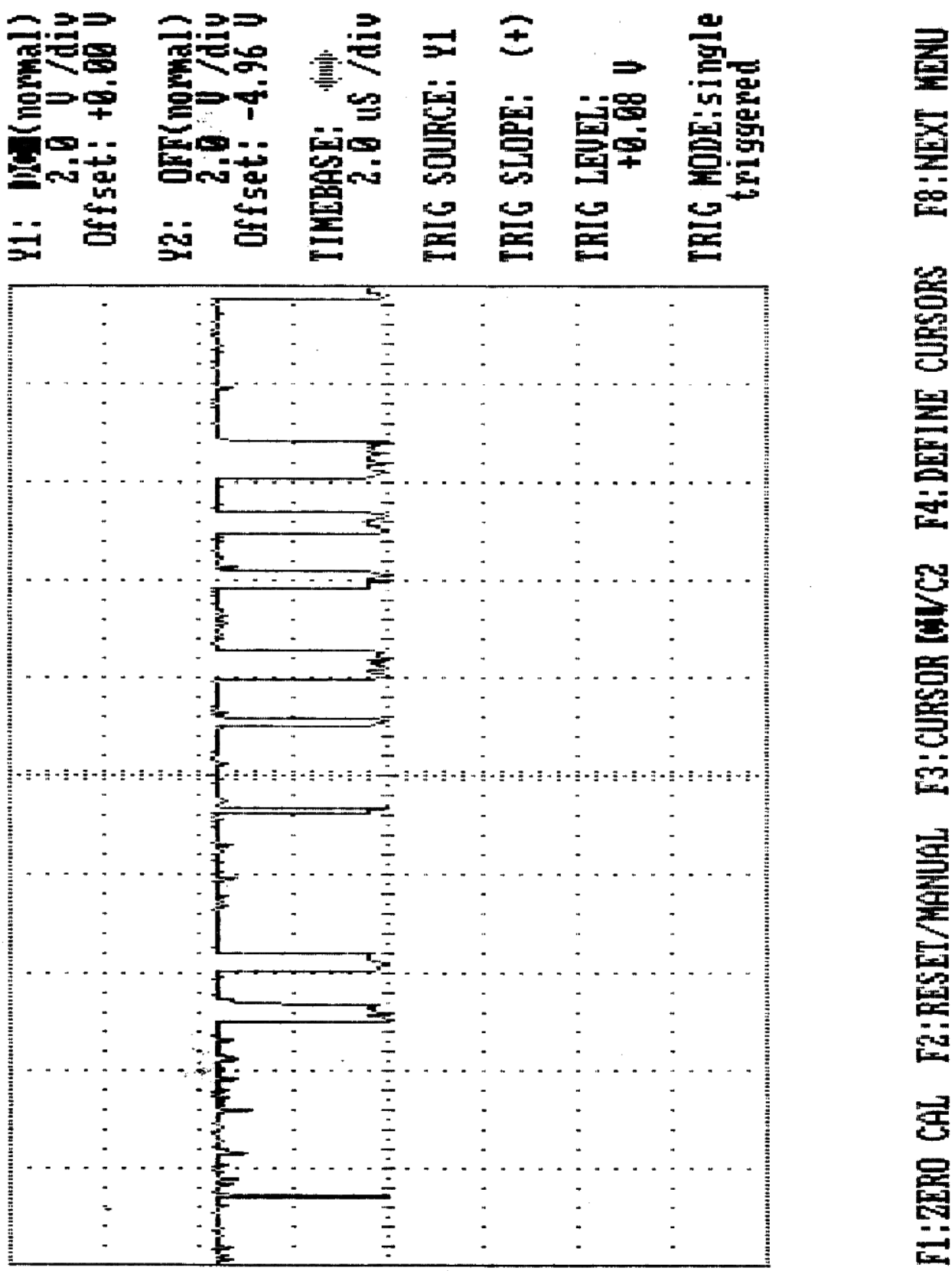

Y1: [redacted](normal)
2.0 V /div
Offset: +0.00 V
Y2: OFF(normal)
100 mV /div
Offset: +0.0mV
TIMEBASE:
1.0 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.00 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR [redacted]/C2 F4:DEFINE CURSORS F8:NEXT MENU

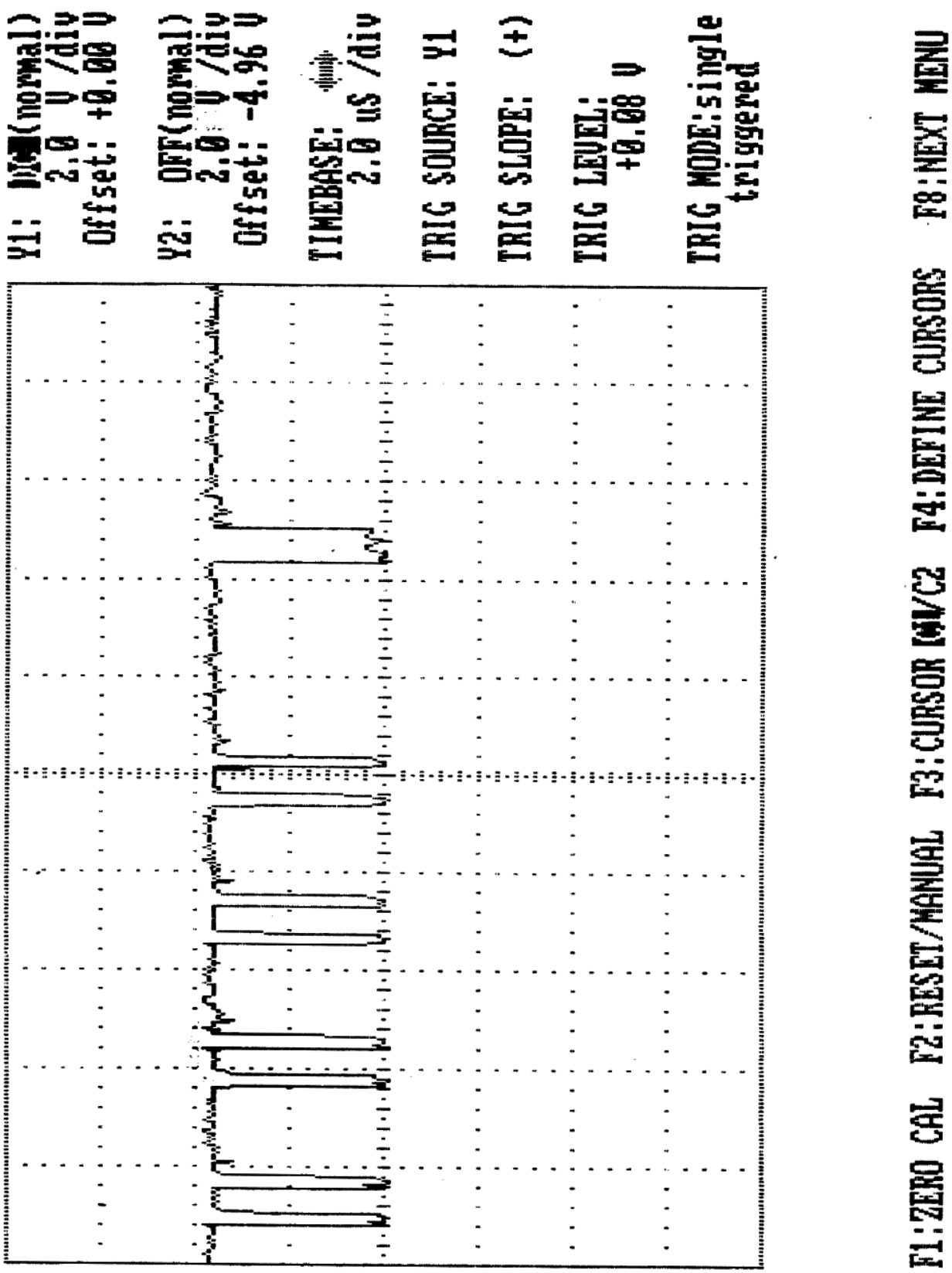

Y1: DC(normal)
2.0 V /div
Offset: +0.00 V
Y2: OFF(normal)
100 mV /div
Offset: +0.0mV
TIMEBASE:
1.0 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.00 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR M/C2 F4:DEFINE CURSORS F8:NEXT MENU

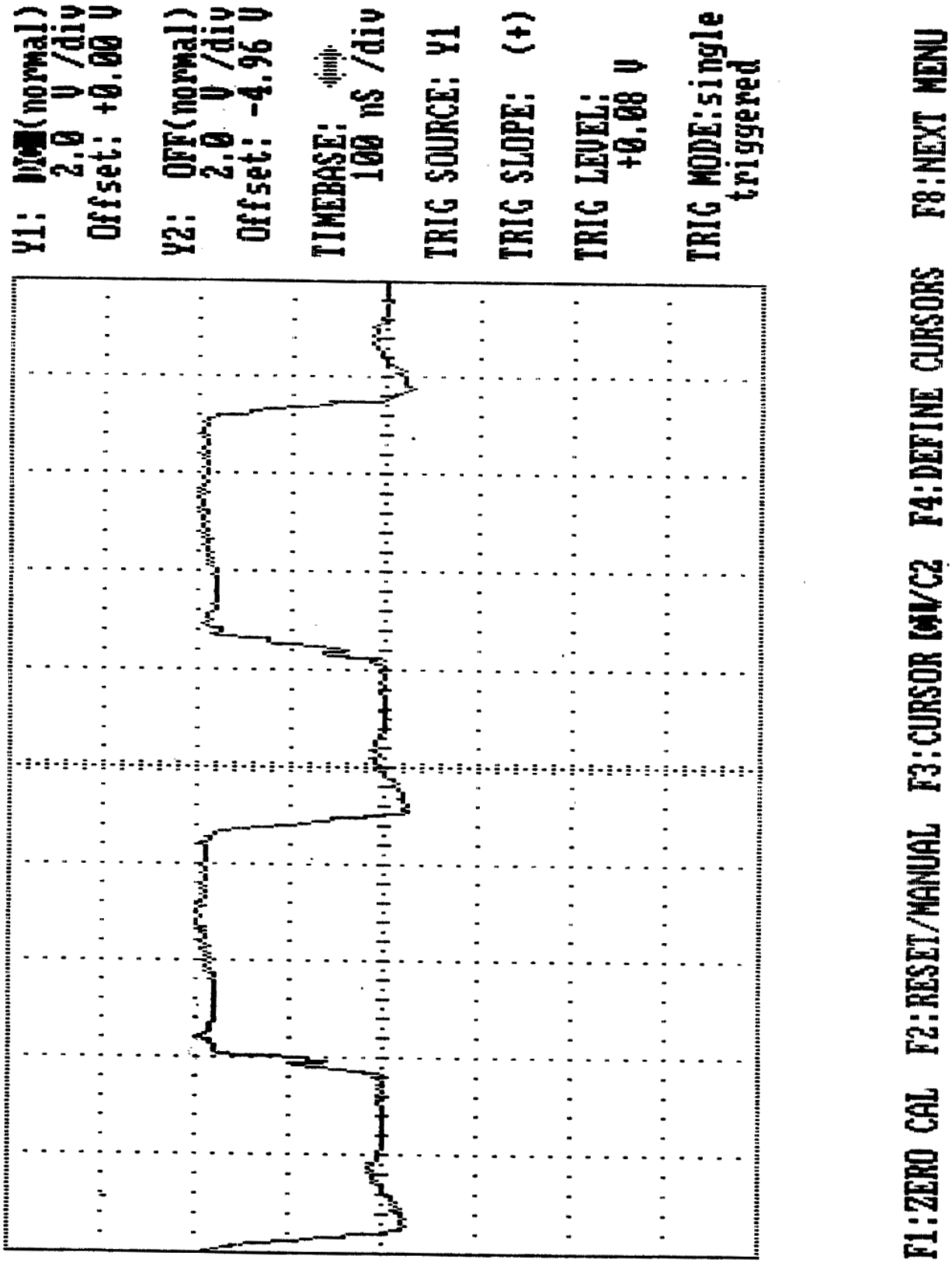

Y1: DC(normal)
2.0 V /div
Offset: +0.00 V
Y2: OFF(normal)
100 mV /div
Offset: +0.0mV
TIMEBASE:
2.0 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.08 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR M/C2 F4:DEFINE CURSORS F8:NEXT MENU

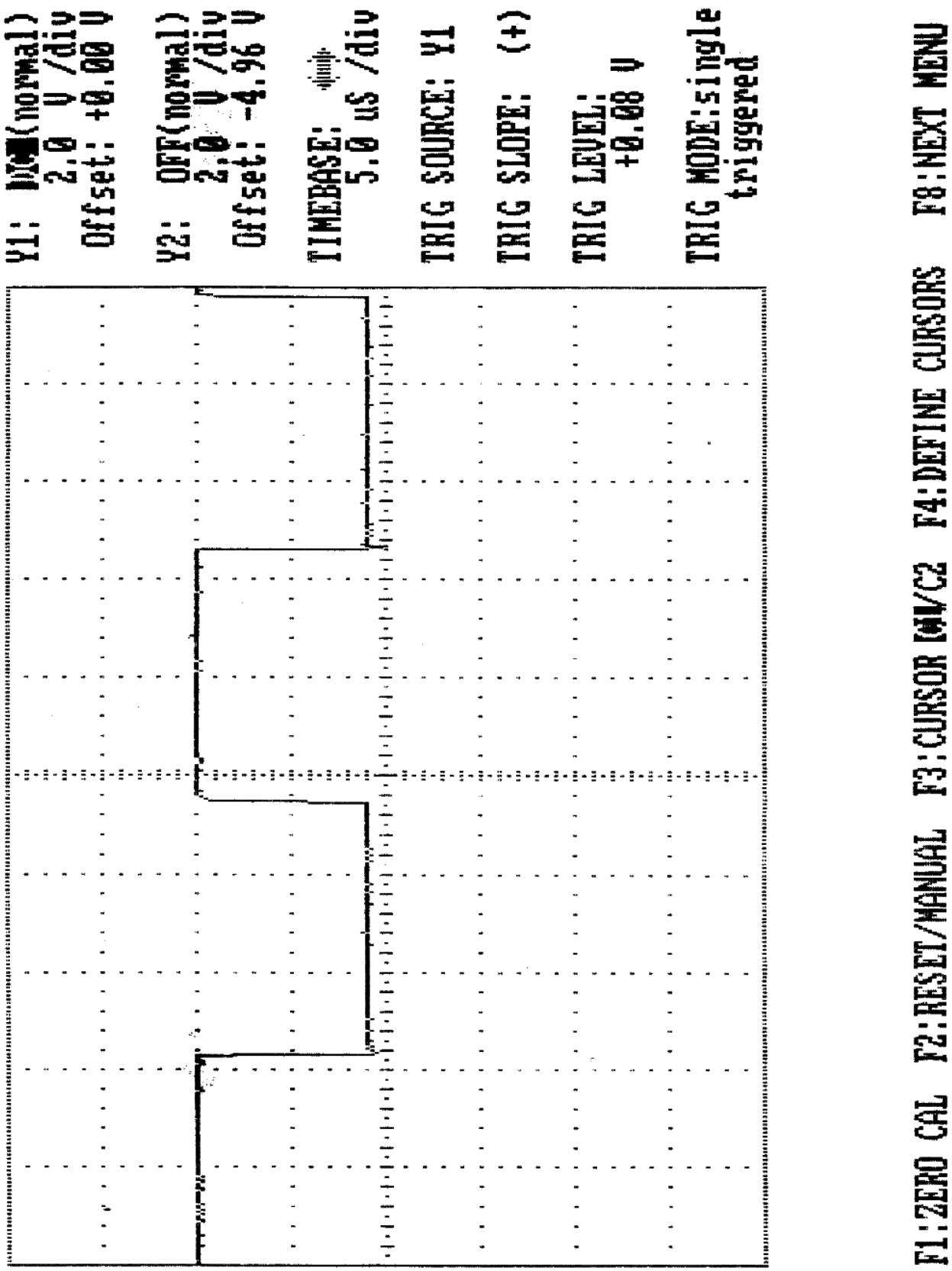

Y1: DC(normal)
2.0 V /div
Offset: +0.00 V
Y2: OFF(normal)
100 mV /div
Offset: +0.0mV
TIMEBASE:
2.0 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.08 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR C1/C2 F4:DEFINE CURSORS F8:NEXT MENU

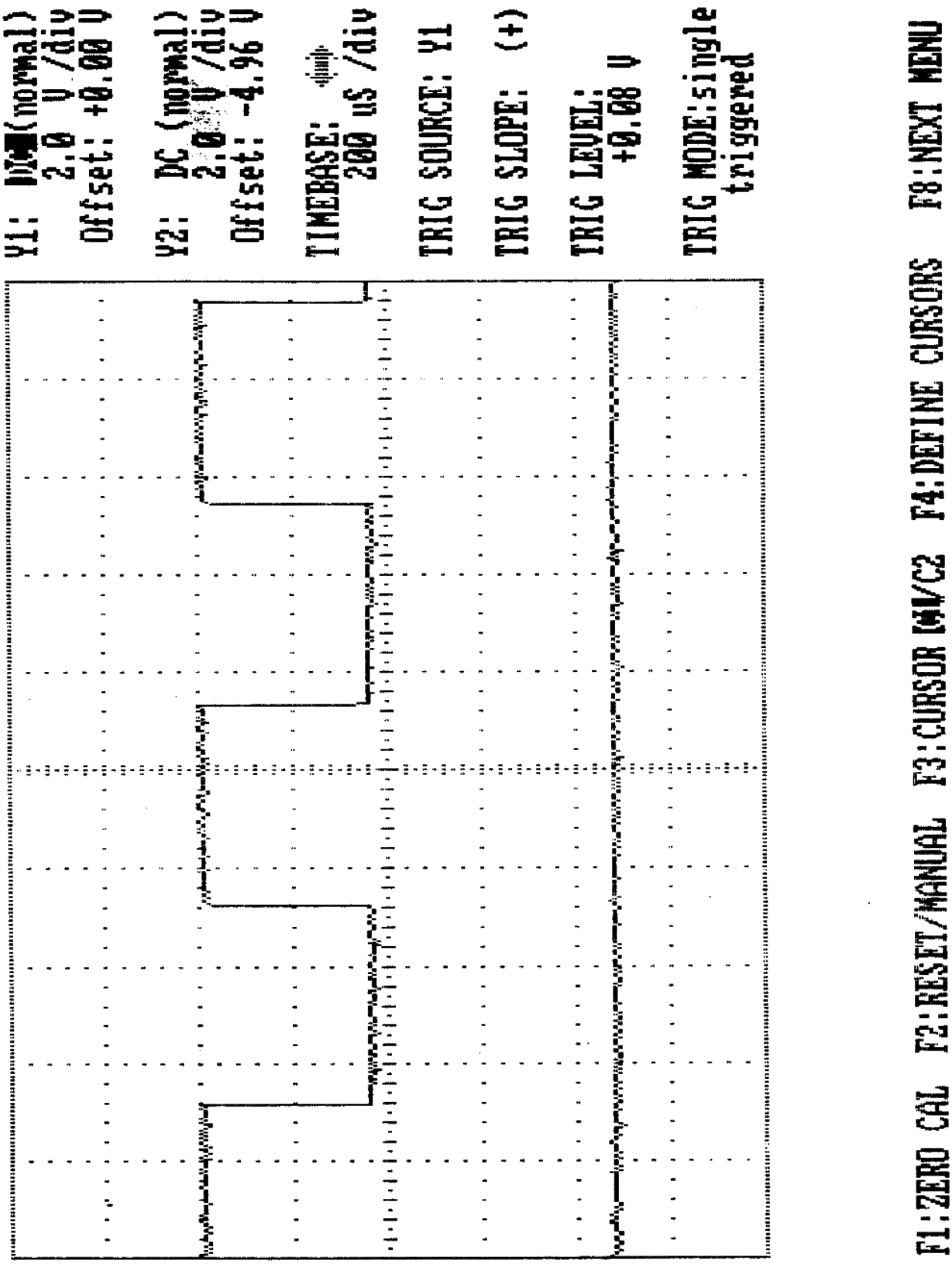

Y1: DC (normal)
2.0 V /div
Offset: +0.00 V
Y2: DC (normal)
2.0 V /div
Offset: -4.96 V
TIMEBASE:
2.0 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.08 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR V/C2 F4:DEFINE CURSORS F8:NEXT MENU

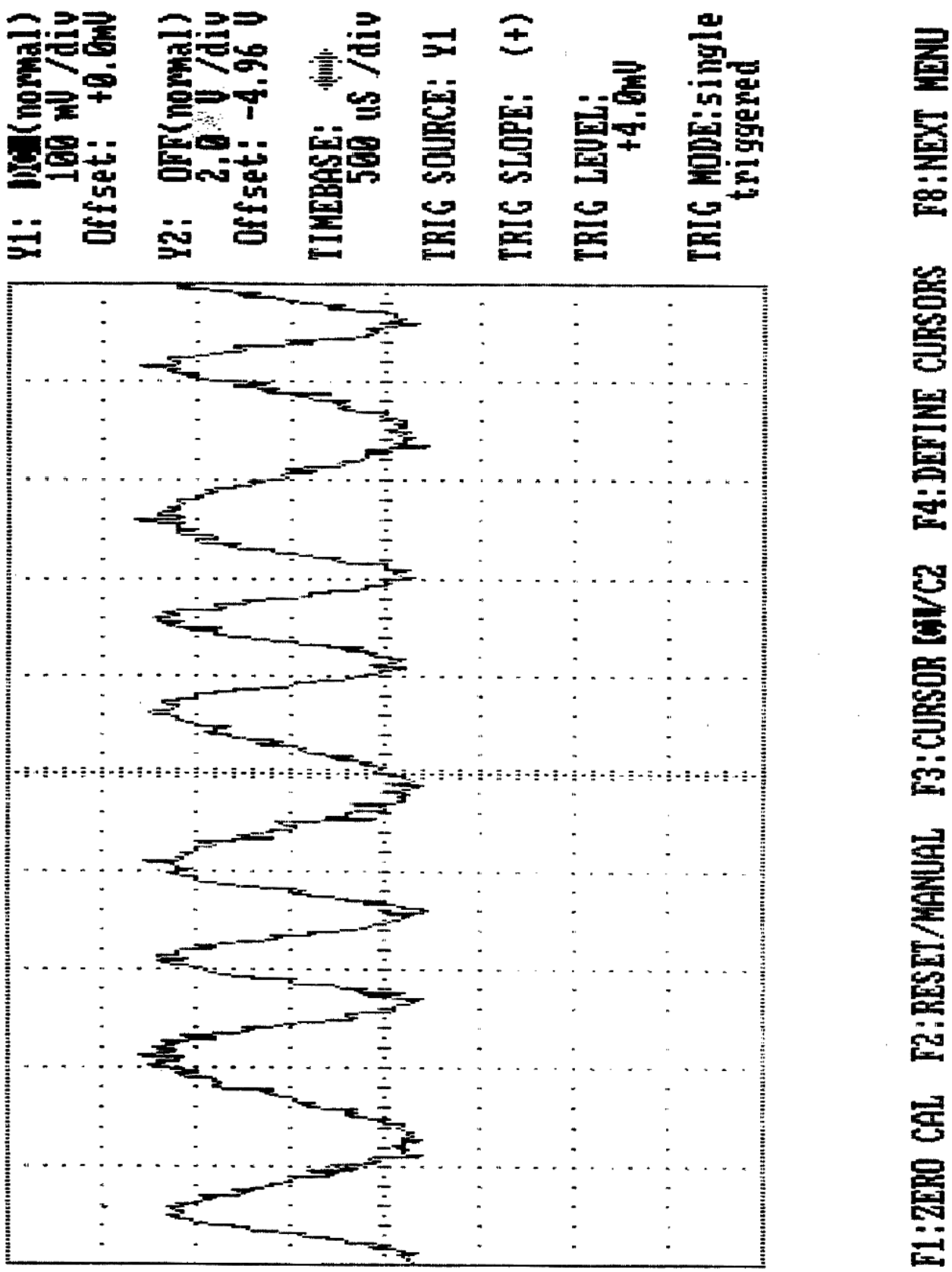

Y1: DC(normal)
2.0 V/div
Offset: +0.00 V
Y2: OFF(normal)
2.0 V/div
Offset: -4.96 V
TIMEBASE:
2.0 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.08 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR C1/C2 F4:DEFINE CURSORS F8:NEXT MENU

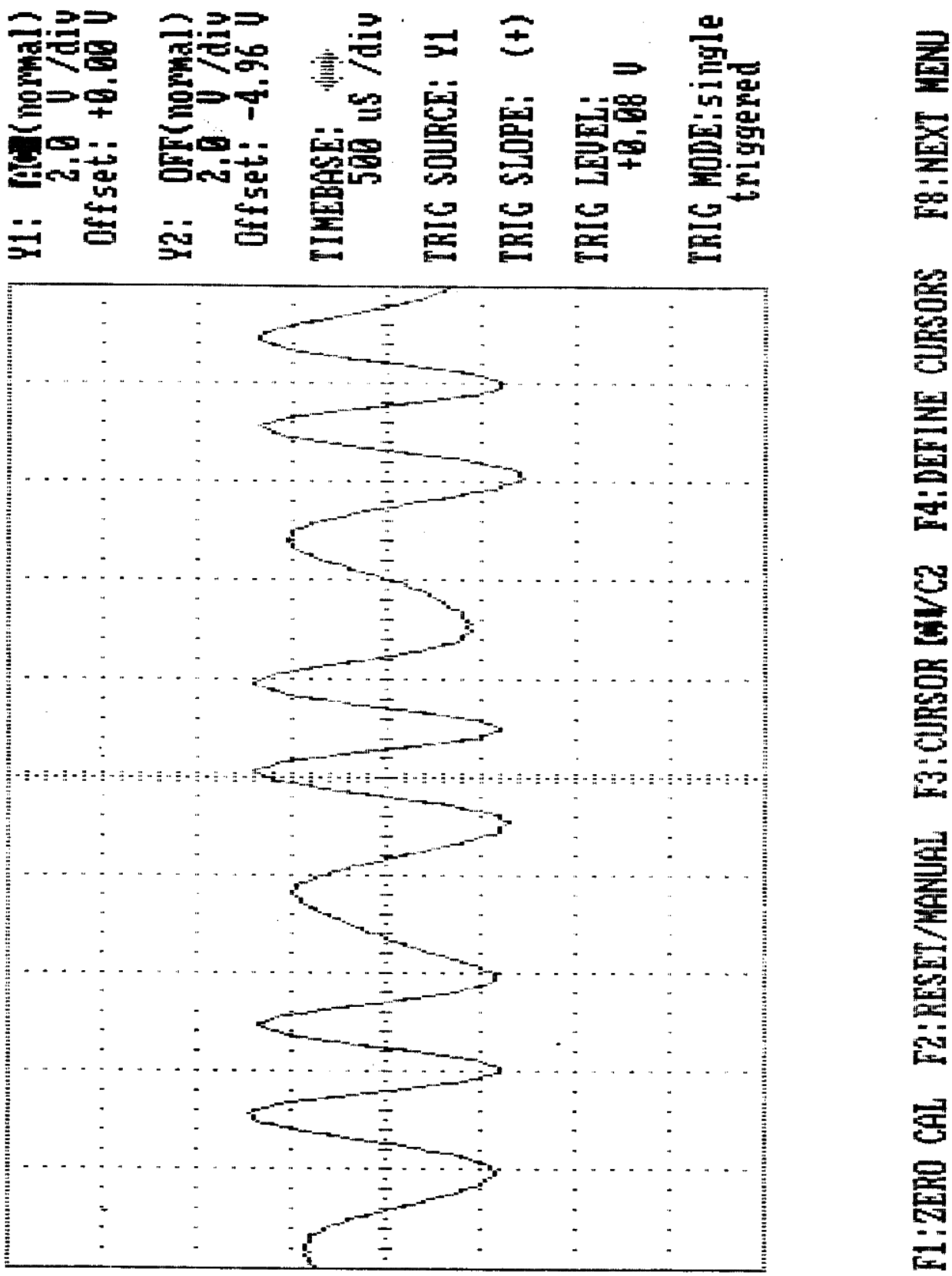

Y1: ON(normal)
2.0 V/div
Offset: +0.00 V
Y2: OFF(normal)
2.0 V/div
Offset: -4.96 V
TIMEBASE:
2.0 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.08 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR ON/C2 F4:DEFINE CURSORS F8:NEXT MENU

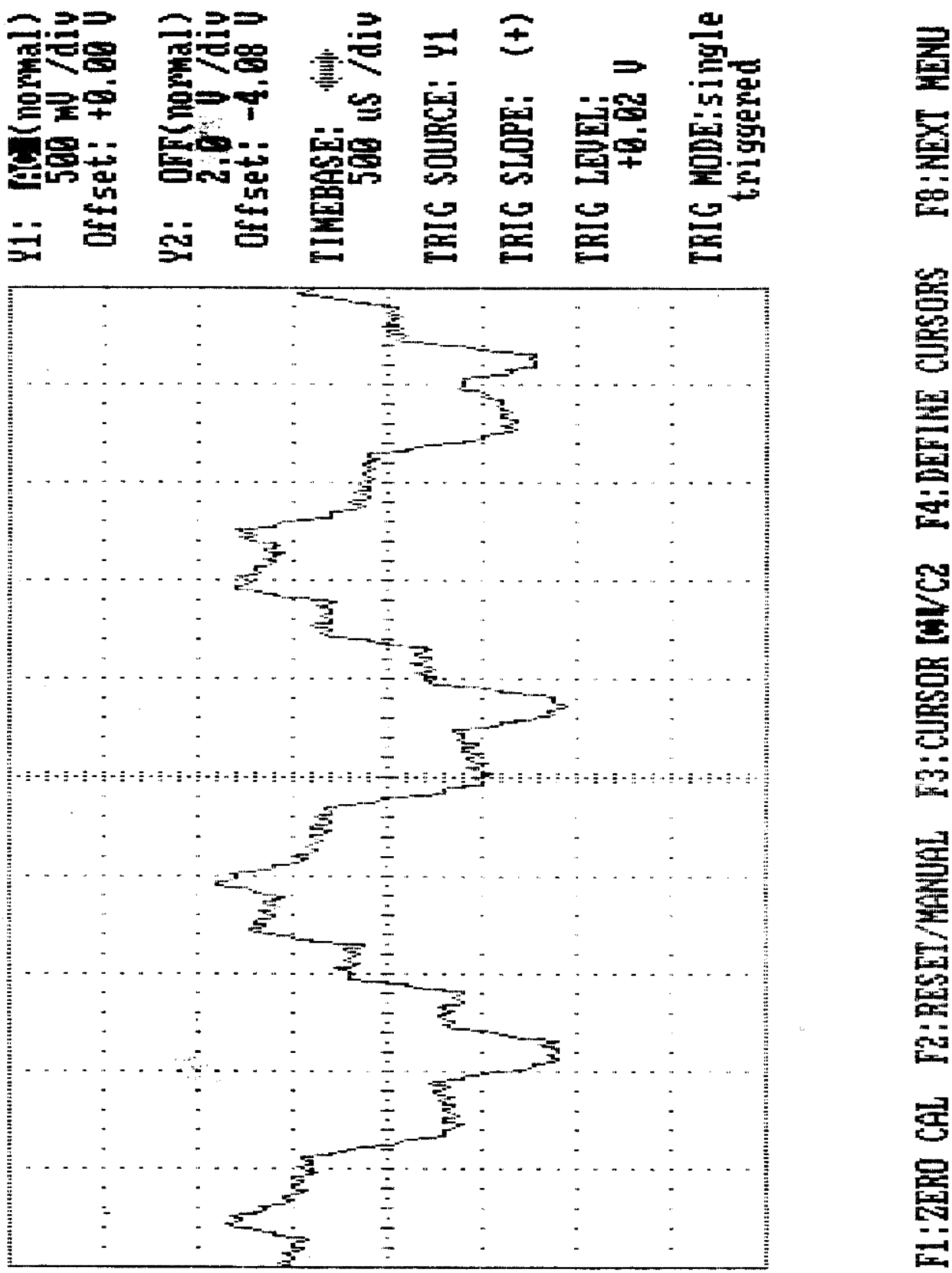

Y1: ON(normal)
2.0 V /div
Offset: +0.00 V
Y2: OFF(normal)
2.0 V /div
Offset: -4.96 V
TIMEBASE:
100 nS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.08 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR C1/C2 F4:DEFINE CURSORS F8:NEXT MENU

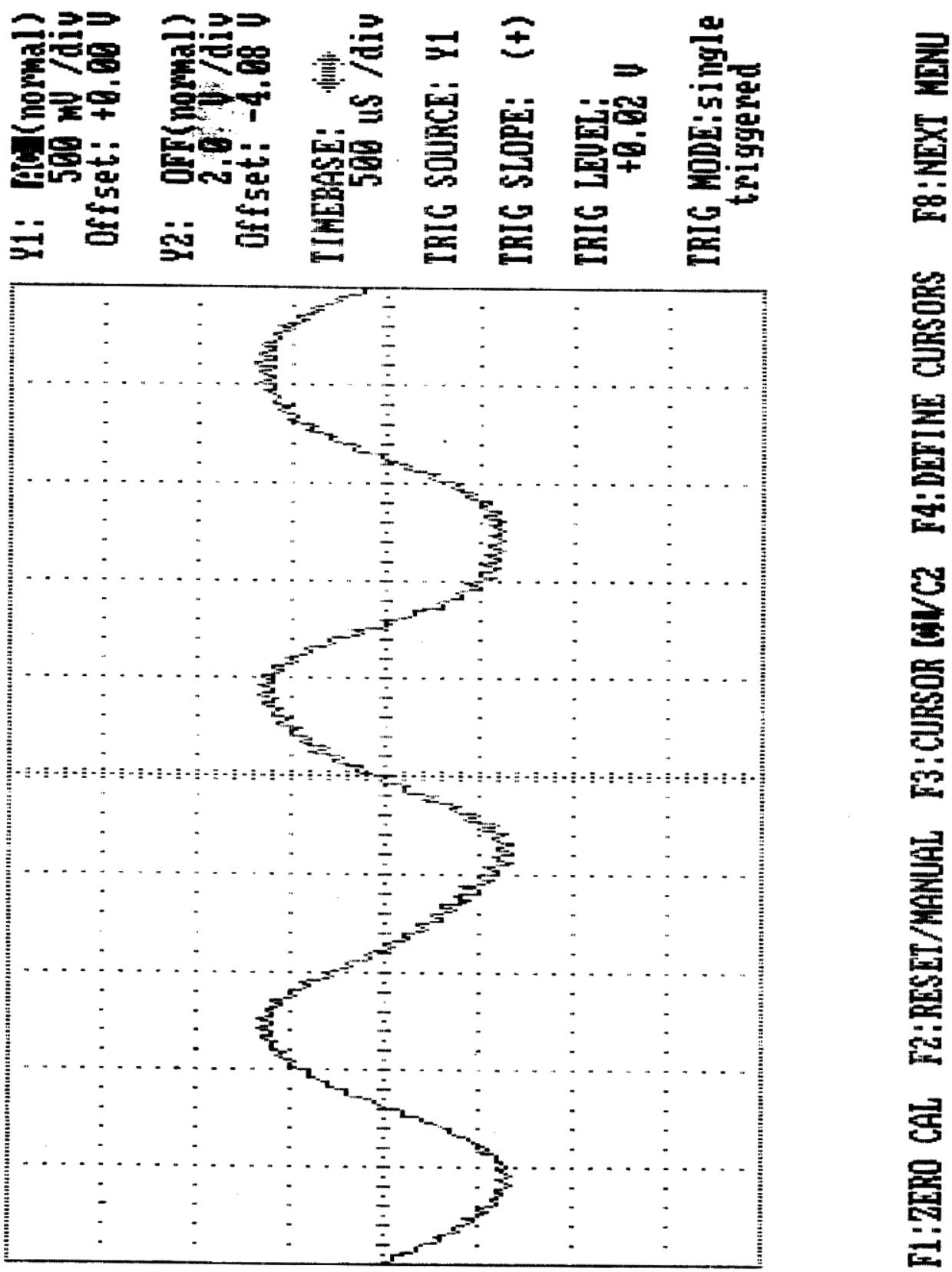

Y1: DC(normal)
2.0 V /div
Offset: +0.00 V
Y2: OFF(normal)
2.0 V /div
Offset: -4.96 V
TIMEBASE:
5.0 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.08 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR M/C2 F4:DEFINE CURSORS F8:NEXT MENU

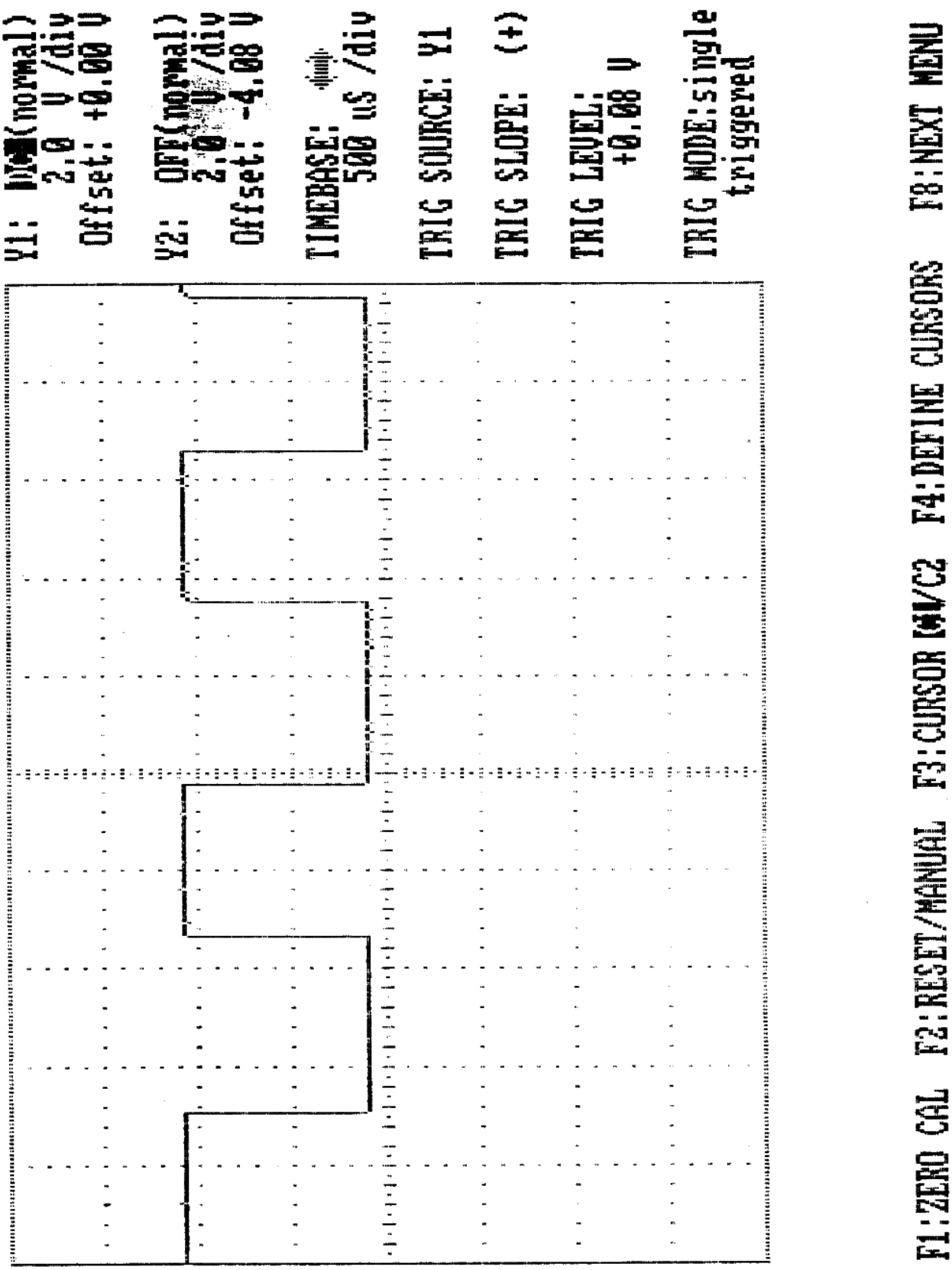

Y1: DC (normal)
2.0 V /div
Offset: +0.00 V
Y2: DC (normal)
2.0 V /div
Offset: -4.96 V
TIMEBASE:
200 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.08 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR M1/C2 F4:DEFINE CURSORS F8:NEXT MENU

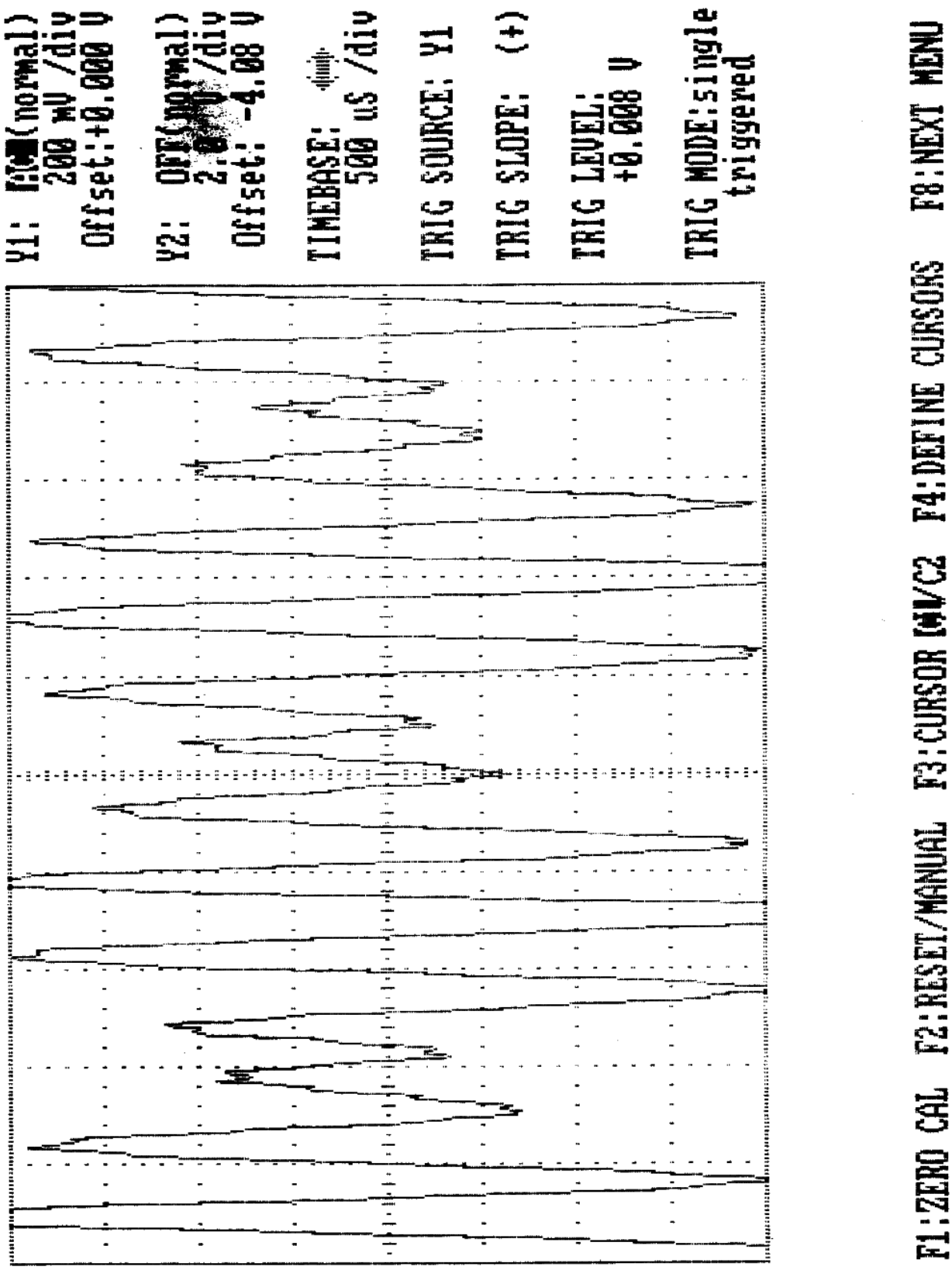

Y1: ON(normal)
100 mV /div
Offset: +0.0mV
Y2: OFF(normal)
2.0 V /div
Offset: -4.96 V
TIMEBASE:
500 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+4.0mV
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR C1/C2 F4:DEFINE CURSORS F8:NEXT MENU

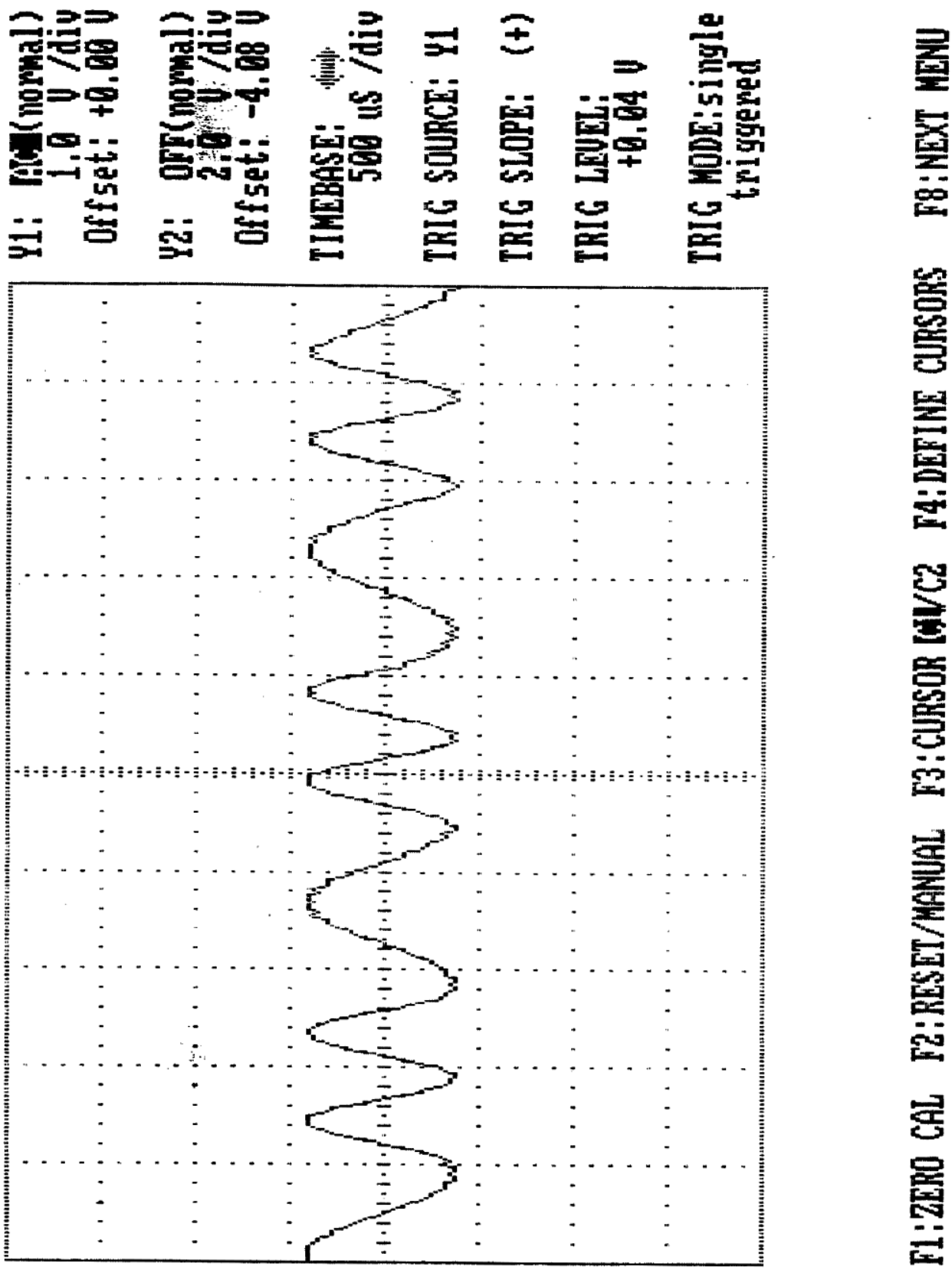

Y1: ON(normal)
2.0 V /div
Offset: +0.00 V
Y2: OFF(normal)
2.0 V /div
Offset: -4.96 V
TIMEBASE:
500 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.08 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR C1/C2 F4:DEFINE CURSORS F8:NEXT MENU

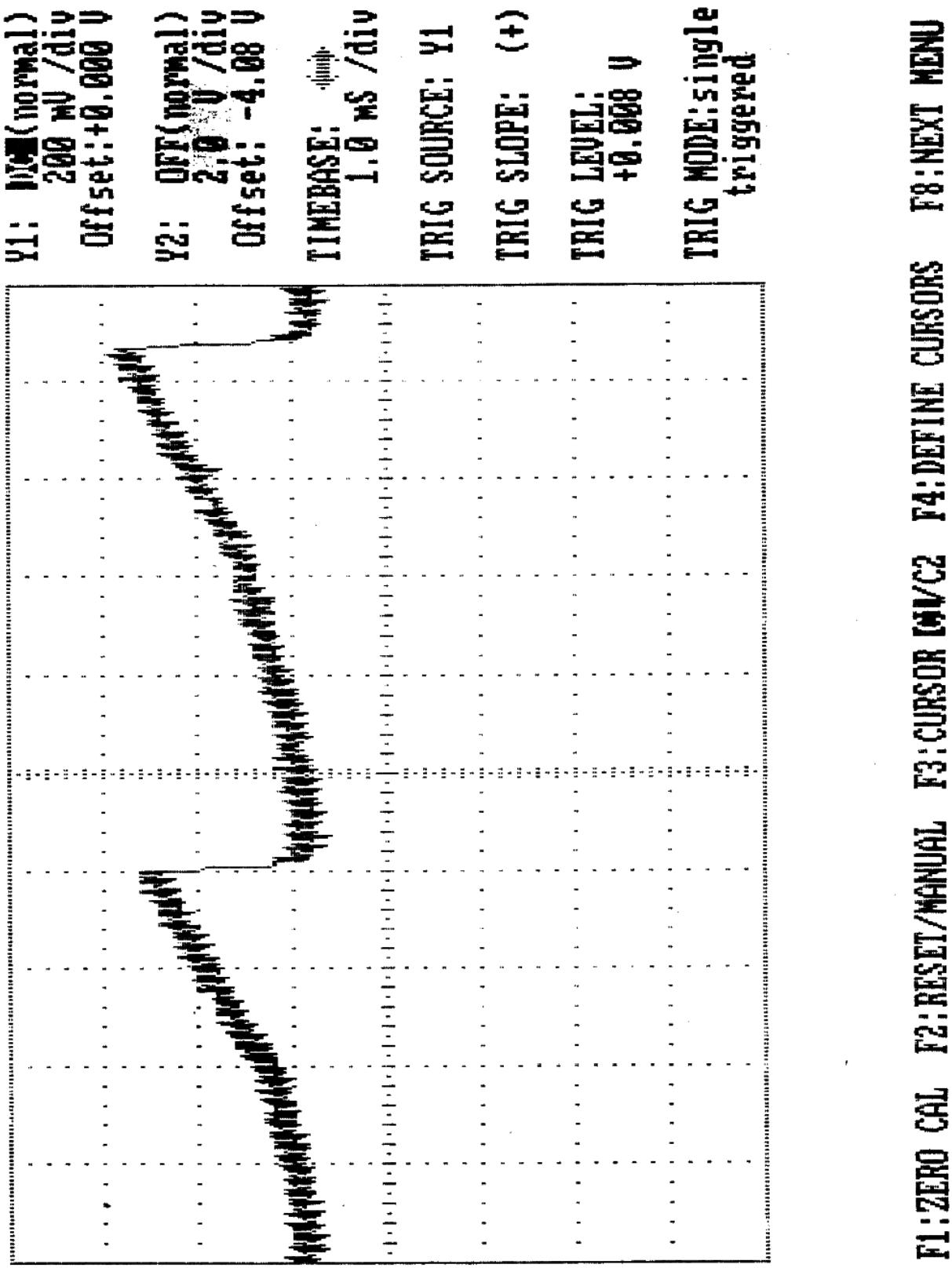

Y1: [illegible](normal)
500 mV /div
Offset: +0.00 V
Y2: OFF(normal)
2.0 V /div
Offset: -4.08 V
TIMEBASE:
500 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.02 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR [illegible]/C2 F4:DEFINE CURSORS F8:NEXT MENU

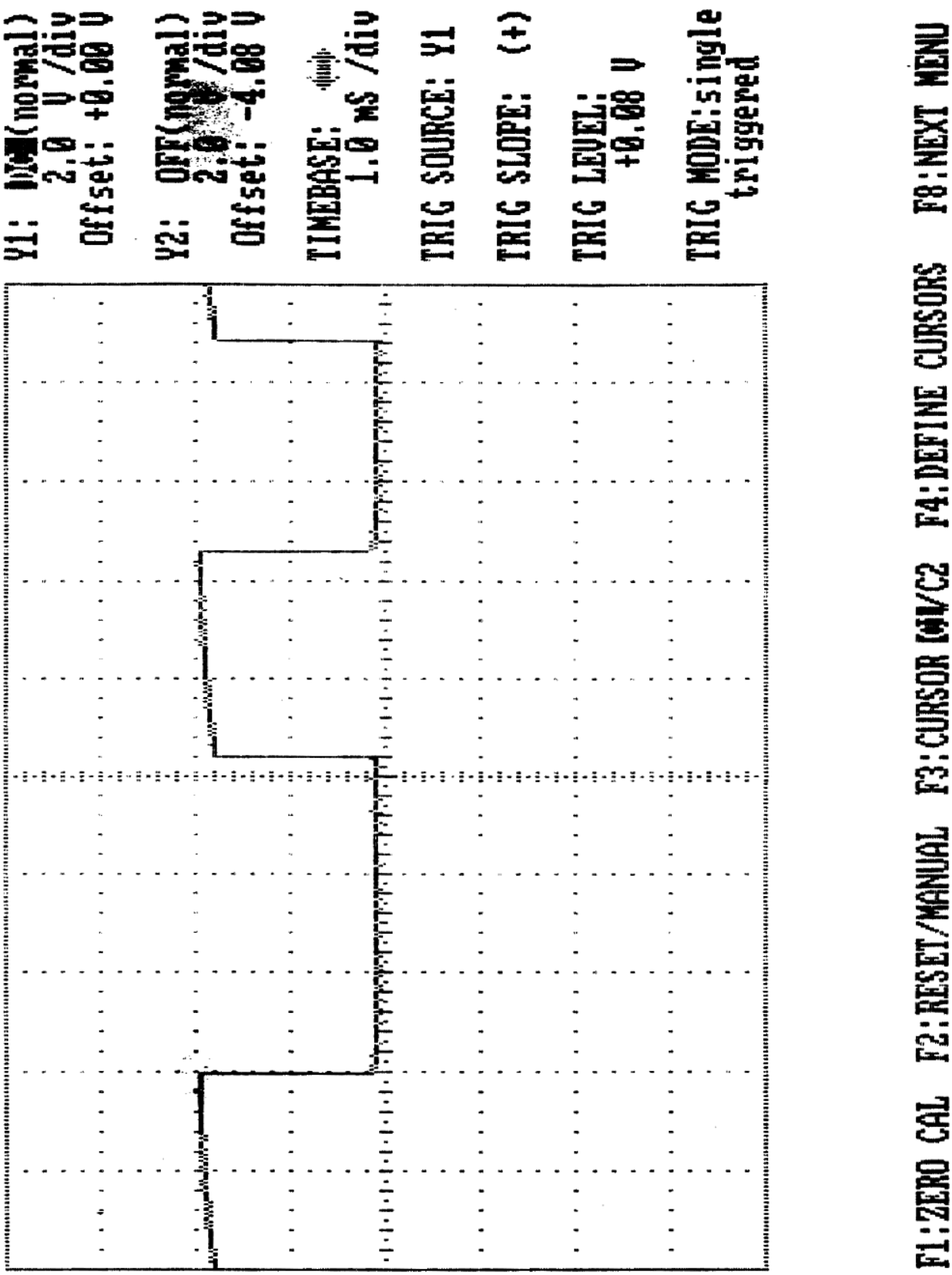

Y1: [illegible](normal)
500 mV /div
Offset: +0.00 V
Y2: OFF(normal)
2.0 V /div
Offset: -4.08 V
TIMEBASE:
500 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.02 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR [illegible]/C2 F4:DEFINE CURSORS F8:NEXT MENU


Y1: ON(normal)
2.0 V/div
Offset: +0.00 V
Y2: OFF(normal)
2.0 V/div
Offset: -4.08 V
TIMEBASE:
500 uS/div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.08 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR M/C2 F4:DEFINE CURSORS F8:NEXT MENU


Y1: [illegible](normal)
200 mV /div
Offset:+0.000 V
Y2: OFF(normal)
2.00 V /div
Offset: -4.08 V
TIMEBASE:
500 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.008 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR [illegible]/C2 F4:DEFINE CURSORS F8:NEXT MENU


Y1: ON(normal)
1.0 V /div
Offset: +0.00 V
Y2: OFF(normal)
2.0 V /div
Offset: -4.08 V
TIMEBASE:
500 uS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.04 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR C1/C2 F4:DEFINE CURSORS F8:NEXT MENU


Y1: DC(normal)
200 mV /div
Offset:+0.000 V
Y2: OFF(normal)
2.0 V /div
Offset: -4.08 V
TIMEBASE:
1.0 mS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.008 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR C1/C2 F4:DEFINE CURSORS F8:NEXT MENU


Y1: ON(normal)
2.0 V/div
Offset: +0.00 V
Y2: OFF(normal)
2.0 V/div
Offset: -4.08 V
TIMEBASE:
1.0 mS /div
TRIG SOURCE: Y1
TRIG SLOPE: (+)
TRIG LEVEL:
+0.08 V
TRIG MODE:single
triggered
F1:ZERO CAL F2:RESET/MANUAL F3:CURSOR C1/C2 F4:DEFINE CURSORS F8:NEXT MENU