

TECHNICAL REFERENCE MANUAL MODEL PK-232 DATA CONTROLLER

ADVANCED ELECTRONICS APPLICATIONS, INC.

(Preliminary Release)

EM2004

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:

- o Reorient the antenna of the device receiving interference.
- o Relocate the computer with respect to this device.
- o Plug the computer into a 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 inaccuracies 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 any obligations whatever.

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

Copyright Advanced Electronics Applications, Inc., 1987. All rights reserved. No part of this manual may be reproduced or used in any form or by any means without prior written permission from the copyright owner.

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 functions 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.

Acknowledgements

AEA, Inc. gratefully acknowledges the Tucson Amateur Packet Radio Corporation, Tucson, AZ, for permission to include excerpts from their TNC-2 documentation in this manual.

Norm Sternberg (W2JUP), Barbara Argilo and Joe Schimmel (W2HPM) developed, wrote and edited this Technical Reference Manual for the PK-232 on Tandy 1000HD and 1000SX computers using IBM's DisplayWrite 3 V1.1 program.

Our very special thanks to Dr. Alan Chandler (K6RFK), Steve Stuart (N6IA), Steve Zopfi (KZ7G), Paul Newland (AD7I), Phil Karn (KA9Q) and Jeff Jacobsen (WA7MBL) for their invaluable help.

AEA, Inc. dedicates itself to the development of digital radio communications.

BATTERY BACK-UP

NOTE: 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.

- o 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.
- o 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.
- o 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

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

CHAPTER 2 - FUNCTIONAL DESCRIPTION

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

CHAPTER 3 - THEORY OF OPERATION

3.1	System Diagrams.....	3-1
3.1.1	Block Diagrams.....	3-1
3.1.2	Logic Diagrams.....	3-1
3.1.3	Schematic Diagrams.....	3-1
3.2	Analog Subsystem.....	3-2
3.2.1	Receive Function.....	3-2
3.2.1.1	Receive Circuits.....	3-3
3.2.2	Transmit Function.....	3-4
3.2.2.1	Transmit Circuits.....	3-4
3.3	Digital Subsystem.....	3-5
3.3.1	Z80A Central Processing Unit.....	3-5
3.3.2	Memory.....	3-5
3.3.3	8536 CIO Counter/Timer and Parallel I/O Unit.....	3-6
3.6.4	8530 SCC Serial Communications Controller.....	3-8

CHAPTER 4 - HOST MODE AND SPECIAL APPLICATIONS

<u>Paragraph</u>	<u>Page</u>
4.1 Introduction to Host Mode.....	4-1
4.1.1 Why Do We Need a Host Mode.....	4-1
4.1.2 How Does Host Mode Help Us?.....	4-2
4.1.3 Entering Host Mode.....	4-2
4.1.4 Leaving Host Mode.....	4-3
4.1.5 The Host Mode Dialog.....	4-3
4.1.6 Host Mode Recovery.....	4-3
4.2 Host Computer Commands.....	4-4
4.2.1 Unsupported Commands.....	4-4
4.2.2 Host Mode Mnemonic Indicators.....	4-5
4.2.3 CONNECT and DISCONNECT.....	4-6
4.2.4 ON/OFF Booleans or Switches.....	4-6
4.2.5 TXDELAY.....	4-6
4.2.6 SENDPAC.....	4-6
4.2.7 Interrogation or Query Commands.....	4-6
4.3 PK-232 Responses.....	4-7
4.3.1 Responses to Interrogation or Query Commands.....	4-8
4.3.2 OPMODE response.....	4-8
4.3.3 Link Status Request Response.....	4-9
4.3.4 Callsign Formats.....	4-9
4.4 Sending Data to the PK-232.....	4-10
4.4.1 Data Polling.....	4-10
4.4.2 The HPOLL Command.....	4-11
4.4.3 Special Case in AMTOR.....	4-11
4.4.4 Link Messages.....	4-11
4.5 Host Mode and Special Packet Applications.....	4-12
4.6 Raw HDLC.....	4-12
4.7 "KISS" TNC Asynchronous Packet Protocol.....	4-13
4.7.1 Starting "KISS" TNC Operation.....	4-13
4.7.2 "KISS" TNC Special Characters.....	4-13
4.7.3 "KISS" TNC Frame Structure.....	4-14
4.7.4 "KISS" TNC Commands.....	4-13
4.7.4.1 TXDELAY: CTL = \$01.....	4-14
4.7.4.2 PERSISTENCE: CTL = \$02.....	4-15
4.7.4.3 SLOTTIME: CTL = \$03.....	4-15
4.7.4.4 TXTAIL: CTL = \$04.....	4-16
4.7.4.5 FULLDUP: CTL = \$05.....	4-16
4.7.4.6 HOST OFF: CTL = \$FF.....	4-16
4.7.4.7 DATA: CTL = \$00.....	4-16
4.8 Maximum Block Size.....	4-17
4.9 MEMORY, I/O and ADDRESS Commands	4-17
4.9.1 MEMORY Command.....	4-17
4.9.2 ADDRESS Command.....	4-18
4.9.3 I/O Command.....	4-18
4.10 Converse and Transparent Modes.....	4-19
4.11 MHEARD Command in Host Mode.....	4-19
4.12 Software Release Date Code.....	4-19
4.13 Product Type Code.....	4-20

TABLE OF CONTENTS

CHAPTER 1 - INTRODUCTION

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

CHAPTER 2 - FUNCTIONAL DESCRIPTION

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

CHAPTER 3 - THEORY OF OPERATION

3.1	System Diagrams.....	3-1
3.1.1	Block Diagrams.....	3-1
3.1.2	Logic Diagrams.....	3-1
3.1.3	Schematic Diagrams.....	3-1
3.2	Analog Subsystem.....	3-2
3.2.1	Receive Function.....	3-2
3.2.1.1	Receive Circuits.....	3-3
3.2.2	Transmit Function.....	3-4
3.2.2.1	Transmit Circuits.....	3-4
3.3	Digital Subsystem.....	3-5
3.3.1	Z80A Central Processing Unit.....	3-5
3.3.2	Memory.....	3-5
3.3.3	8536 CIO Counter/Timer and Parallel I/O Unit.....	3-6
3.6.4	8530 SCC Serial Communications Controller.....	3-8

CHAPTER 4 - HOST MODE AND SPECIAL APPLICATIONS

<u>Paragraph</u>	<u>Page</u>	
4.1	Introduction to Host Mode.....	4-1
4.1.1	Why Do We Need a Host Mode.....	4-1
4.1.2	How Does Host Mode Help Us?.....	4-2
4.1.3	Entering Host Mode.....	4-2
4.1.4	Leaving Host Mode.....	4-3
4.1.5	The Host Mode Dialog.....	4-3
4.1.6	Host Mode Recovery.....	4-3
4.2	Host Computer Commands.....	4-4
4.2.1	Unsupported Commands.....	4-4
4.2.2	Host Mode Mnemonic Indicators.....	4-5
4.2.3	CONNECT and DISCONNECT.....	4-6
4.2.4	ON/OFF Booleans or Switches.....	4-6
4.2.5	TXDELAY.....	4-6
4.2.6	SENDPAC.....	4-6
4.2.7	Interrogation or Query Commands.....	4-6
4.3	PK-232 Responses.....	4-7
4.3.1	Responses to Interrogation or Query Commands.....	4-8
4.3.2	OPMODE response.....	4-8
4.3.3	Link Status Request Response.....	4-9
4.3.4	Callsign Formats.....	4-9
4.4	Sending Data to the PK-232.....	4-10
4.4.1	Data Polling.....	4-10
4.4.2	The HPOLL Command.....	4-11
4.4.3	Special Case in AMTOR.....	4-11
4.4.4	Link Messages.....	4-11
4.5	Host Mode and Special Packet Applications.....	4-12
4.6	Raw HDLC.....	4-12
4.7	"KISS" TNC Asynchronous Packet Protocol.....	4-13
4.7.1	Starting "KISS" TNC Operation.....	4-13
4.7.2	"KISS" TNC Special Characters.....	4-13
4.7.3	"KISS" TNC Frame Structure.....	4-14
4.7.4	"KISS" TNC Commands.....	4-13
4.7.4.1	TXDELAY: CTL = \$01.....	4-14
4.7.4.2	PERSISTENCE: CTL = \$02.....	4-15
4.7.4.3	SLOTTIME: CTL = \$03.....	4-15
4.7.4.4	TXTAIL: CTL = \$04.....	4-16
4.7.4.5	FULLDUP: CTL = \$05.....	4-16
4.7.4.6	HOST OFF: CTL = \$FF.....	4-16
4.7.4.7	DATA: CTL = \$00.....	4-16
4.8	Maximum Block Size.....	4-17
4.9	MEMORY, I/O and ADDRESS Commands	4-17
4.9.1	MEMORY Command.....	4-17
4.9.2	ADDRESS Command.....	4-18
4.9.3	I/O Command.....	4-18
4.10	Converse and Transparent Modes.....	4-19
4.11	MHEARD Command in Host Mode.....	4-19
4.12	Software Release Date Code.....	4-19
4.13	Product Type Code.....	4-20

CHAPTER 5 - MECHANICAL ASSEMBLY AND DISASSEMBLY

5.1	Cover Removal.....	5-1
5.2	Circuit Board Removal.....	5-1
5.3	Circuit Board Assembly.....	5-2
5.4	Cover Assembly.....	5-2

CHAPTER 6 - ADJUSTMENTS

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

CHAPTER 7 - TROUBLESHOOTING

7.1	Introduction.....	7-1
7.2	General Tests.....	7-1
7.2.1	Power Supply.....	7-2
7.2.2	Obvious Problems.....	7-2
7.2.3	Assembly Problems.....	7-2
7.2.4	Cabling Problems.....	7-2
7.3	Specific Symptoms.....	7-3
7.3.1	Symptom: Controller appears dead.....	7-3
7.3.2	Symptom: Modem cannot be calibrated.....	7-4
7.3.3	Symptom: Transmitter cannot be keyed.....	7-5
7.3.4	Symptom: Transmitted signals not copyable by other stations..	7-5
7.3.5	Symptom: Received signals not copyable	7-5
7.4	Terminal Interface Troubleshooting.....	7-6
7.4.1	Symptom: Controller does not communicate with the terminal...	7-6
7.4.2	Symptom: Controller signs on with mutilated data.....	7-6
7.4.3	Symptom: Controller does not respond or accept commands.....	7-7

APPENDIX A - AX.25 LEVEL 2 PROTOCOL

APPENDIX B - "KISS" TNC SPECIFICATION

APPENDIX C - DRAWINGS

APPENDIX D - WAVEFORMS

(This page intentionally left blank)

CHAPTER 1

INTRODUCTION

1.1 Introduction

This Technical Reference manual will assist you to maintain, repair, and adjust your PK232 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 a 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 applications developers.

1.3 General Description

AEA's Model PK232 is a multi-mode protocol converter and data controller that includes self-contained modems for all modes.

The PK232 sends and receives Morse, Baudot and ASCII RTTY, facsimile, AMTOR/SITOR and AX.25 packet. The PK-232 converts these signals to ASCII data and sends the data to your terminal via an EIA standard RS232 (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 PK232.

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, 1550
HF (except CW)	Center frequency 2210 Hz, 1350
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

CHAPTER 1

INTRODUCTION

1.1 Introduction

This Technical Reference manual will assist you to maintain, repair, and adjust your PK232 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 a 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 applications developers.

1.3 General Description

AEA's Model PK232 is a multi-mode protocol converter and data controller that includes self-contained modems for all modes.

The PK232 sends and receives Morse, Baudot and ASCII RTTY, facsimile, AMTOR/SITOR and AX.25 packet. The PK-232 converts these signals to ASCII data and sends the data to your terminal via an EIA standard RS232 (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 PK232.

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, 1550
HF (except CW)	Center frequency 2210 Hz, 1350
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-232-C 25-pin DB25 connector
Serial Input/Output	RS-232-C with full hardware and software handshake on wires 1 through 8 and 20
Parallel Input/Output	Centronics-compatible with handshake, circuits available in non-standard configuration of DB-25 Serial Port.
Terminal Data Rates	Autobaud 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 (Data Carrier Detect)																								
Status and Mode Indicators:	<table> <thead> <tr> <th><u>Mode Group</u></th> <th><u>Status Group</u></th> </tr> </thead> <tbody> <tr> <td>BAUDOT</td> <td>STBY</td> </tr> <tr> <td>ASCII</td> <td>PHASE</td> </tr> <tr> <td>PKT</td> <td>IDLE</td> </tr> <tr> <td>MORSE</td> <td>ERROR/CONV</td> </tr> <tr> <td>CHECK</td> <td>OVER</td> </tr> <tr> <td>FEC</td> <td>TFC/TRANS</td> </tr> <tr> <td>ARQ</td> <td>RQ/CMD</td> </tr> <tr> <td>MODE L</td> <td>CON</td> </tr> <tr> <td>STBY</td> <td>STA</td> </tr> <tr> <td></td> <td>MULT</td> </tr> <tr> <td></td> <td>SEND</td> </tr> </tbody> </table>	<u>Mode Group</u>	<u>Status Group</u>	BAUDOT	STBY	ASCII	PHASE	PKT	IDLE	MORSE	ERROR/CONV	CHECK	OVER	FEC	TFC/TRANS	ARQ	RQ/CMD	MODE L	CON	STBY	STA		MULT		SEND
<u>Mode Group</u>	<u>Status Group</u>																								
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' x 8.25' x 2.5' (279.4 mm X 209.6 mm X 63.5 mm) Weight 3 pounds (1.36 kilograms)

(This page intentionally left blank)

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-232-C 25-pin DB25 connector
Serial Input/Output	RS-232-C with full hardware and software handshake on wires 1 through 8 and 20
Parallel Input/Output	Centronics-compatible with handshake, circuits available in non-standard configuration of DB-25 Serial Port.
Terminal Data Rates	Autobaud 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 (Data Carrier Detect)																								
Status and Mode Indicators:	<table> <thead> <tr> <th><u>Mode Group</u></th> <th><u>Status Group</u></th> </tr> </thead> <tbody> <tr> <td>BAUDOT</td> <td>STBY</td> </tr> <tr> <td>ASCII</td> <td>PHASE</td> </tr> <tr> <td>PKT</td> <td>IDLE</td> </tr> <tr> <td>MORSE</td> <td>ERROR/CONV</td> </tr> <tr> <td>CHECK</td> <td>OVER</td> </tr> <tr> <td>FEC</td> <td>TFC/TRANS</td> </tr> <tr> <td>ARQ</td> <td>RQ/CMD</td> </tr> <tr> <td>MODE L</td> <td>CON</td> </tr> <tr> <td>STBY</td> <td>STA</td> </tr> <tr> <td></td> <td>MULT</td> </tr> <tr> <td></td> <td>SEND</td> </tr> </tbody> </table>	<u>Mode Group</u>	<u>Status Group</u>	BAUDOT	STBY	ASCII	PHASE	PKT	IDLE	MORSE	ERROR/CONV	CHECK	OVER	FEC	TFC/TRANS	ARQ	RQ/CMD	MODE L	CON	STBY	STA		MULT		SEND
<u>Mode Group</u>	<u>Status Group</u>																								
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' x 8.25' x 2.5' (279.4 mm X 209.6 mm X 63.5 mm) Weight 3 pounds (1.36 kilograms)

(This page intentionally left blank)

CHAPTER 2

FUNCTIONAL DESCRIPTION

2.1 Major Sections

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

- o Analog
- o Digital
- o Input/Output (I/O)
- o Display
- o 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 and 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 or 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:

- o DIGITAL SECTION
- o ANALOG SECTION
- o I/O SECTION
- o DISPLAY SECTION
- o POWER SECTION

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:

- o the microprocessor
- o Read Only Memory (ROM)
- o Random Access Memory (RAM)
- o clock and control logic
- o part of the display logic
- o 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:

- o mode-dependant bandpass filters
- o MARK and SPACE resonators
- o audio frequency discriminator
- o mode-dependant low pass filter
- o comparator circuitry
- o AFSK generator
- o watchdog timer
- o transmitter audio driver
- o keying circuitry
- o radio switching
- o 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 RADIO1 or RADIO2 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 signals 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 a 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 commanded 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, R167 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 48K of EPROM at U2 and U3, and 16K of static RAM at U4 and U5.

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

EPROM U3 is a 27128 occupying 16K bytes from \$8000 to \$BFFF.

The read-write memory consists of two 6264 8K 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 transitions 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 ports B and C work together for two functions. Most of the time they drive the front-panel LEDs through decoder chips. If PRCON is ON, they interface with the parallel printer.

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 LEDs 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 pins 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.

3.6.4 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 at the packet baud rate. The divide-by-32 function is performed by the 74LS393 at U8.

The U7 TRXCA output (pin 14) is the 32x 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 to the U7 RTXCA input (pin 12) as the 1x 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 (10K) and C55 (47 uf).

For this scheme to work properly, the DTRA output is toggled every pass through 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 U16 (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 Z8030/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.

(This page intentionally left blank)

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 of 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 one 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 on 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 communicating 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 flow 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.

- o To ensure correct entry into Host mode, send the 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 (CTRL-A).
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, go 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  CTRL-A  O     H     O     N     CTRL-W

```

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:

- o Each block starts with the SOH (Start of Header) character, which is hex \$01 or CTRL-A.
- o 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 ends with hex \$17 or CTRL-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), ASCII \$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 sends blocks to the PK-232 using these 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, do not send 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 Host 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 AMTOR	AC ARQ
AO ARQTM0	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 MSG	C1 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 DISCONN
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 ILFPACK	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	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 PERSIST	PC PRCON	PF PRFAX	PO PROUT
PY PRYPE	RW RAWHDL	RB RBAUD	RC RCVE	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 SLOTTIME	SQ SQUELCH	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 host 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 = \$4x, where x is the channel 0-9.

4.2.4 ON/OFF Booleans or Switches

To set an 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 Query Commands

Query commands are similar to the human or verbose mode in that the two-letter command to the PK-232 is not 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 CHSWITCH character. Only channel 0 is used in all communications modes 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 connectees
 \$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 Query 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 BEACON or PACTIME: E or A, space, and a number
- for CONMODE: C or T
- for OPMODE: see below
- for all other commands: the same as in the verbose mode, except for CONNECT, which is discussed later 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 w x ETB
ARQ Listen:	SOH \$4F O P A L w R ETB
FEC:	SOH \$4F O P F E w x ETB
SELFEC:	SOH \$4F O P S E w x ETB
FAX:	SOH \$4F O P F A v x ETB

Pactor where: *P t 0 R 1000*

- | | |
|---------------------|-------------------|
| v = \$30 if Standby | x = S if transmit |
| \$31 if Phasing | R if receive |
| \$32 if Change-over | |
| \$33 if Idle | |
| \$34 if Traffic | |
| \$35 if Error | |
| \$36 if RQ | |
| \$30 if Standby | |
| \$31 if Sync | |

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 connectee, 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.

47	SOH	\$2F	. . .	ETB	echoed data
48.57.57	SOH	\$3x	. . .	ETB	data
63	SOH	\$3F	. . .	ETB	monitored data
64.65.65	SOH	\$4x	. . .	ETB	link status
60.65.89	SOH	\$5x	. . .	ETB	link messages
95	SOH	\$5F	X X .	ETB	status errors
95	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	\$20	(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	ETB	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 modes other than packet, received data is formed into blocks one character at a time, for a total of fyour 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 modes 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 only 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 must 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 includes 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 possibly 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 for \$10.

4.7 "KISS" TNC Asynchronous Packet Protocol

The PK-232 provides a simple, asynchronous, computer-to-TNC protocol for a raw HDLC TNC 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 exceptions:

- o The frame delimiter characters.
- o The commands available.
- o The method of determining the proper time to transmit.
- o 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
RAHDLC     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-character 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.

PERSISTENCE is best described 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 after 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:

```
SC0 $02 n SC0
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 = (255+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:

```
SC0 $03 n SC0
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 milli-seconds.

4.7.4.5 **FULLDUP: CTL = \$05**

The FULLDUP 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.

- o 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.
- o 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 MONITOR set to 6.

This worst-case configuration produces a maximum of $136 + 256$, or 392 bytes, not including SOH, CTL, DLE and ETB. Byte stuffing could 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 containing 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 IO commands work with the ADDRESS command to permit host access to the PK-232's memory and I/O locations.

4.9.1 MEMORY Command

To use the Memory command:

- o Set the memory address into the ADDRESS command.
- o Use the MEMORY command without arguments to read memory locations one after another.
- o 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 \$C000-\$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 IO 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 IO 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 RR0:

- 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, then 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 Date 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 returned 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 are 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**

- o Disconnect all connectors from the rear of the PK-232.
- o Remove six black anodized #6 Phillips head screws from the sides and rear of the cover.
- o Slide the cover towards the rear of the unit until the connectors clear the cutouts on the rear of the cover.
- o Lift the cover and carefully tip it rearward. 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.

- o 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.
- o With a small blade screwdriver, loosen the screw holding the control knob to the shaft of the THRESHOLD control and remove the knob.
- o Remove the control nut and control washer.
- o Remove six black anodized #6 Phillips head screws holding the circuit board to the chassis. Some units may have four nickel plated screws in the board corners and two black anodized screws in the other two positions.
- o Unsolder the red and black leads where they enter the circuit board. This can be done from the top of the circuit board.
- o 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 bent.

- o 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.
- o Fasten the circuit board to the chassis with the six #6 black anodized Phillips head screws.
- o Fasten the control shaft to the front panel using the control washer and control nut.
- o Turn the shaft of the control 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.
- o Tighten the knob securely on the shaft.
- o Place the cover on edge next to the chassis and solder the black lead from the battery holder to the land closest to the switch assembly.
- o Solder the red lead to the land closest to JP-1.
- o Replace the jumper on JP-1.

5.4 Cover Assembly

- o 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.
- o 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 PK-232 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 (counter-clockwise). 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.

6.2.1 AFSK Generator (Transmit) Adjustments

While in the calibration routine use:

- o SPACE BAR to toggle between mark and space,
 - o 'H' to toggle between VHF (1200/2200) and HF (2110/2310),
 - 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 RADIO1 (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' (without the quotation marks) followed by the <return> key.
4. Adjust R167 for a reading of 1200 +/- 10 Hz as displayed 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 a reading of 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) Adjustments

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 Tests

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 now 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.

The monitor should echo these characters back.

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>.

The monitor should echo these same characters.

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 measurements 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 pins on integrated circuits when connecting meter or oscilloscope probes to the board. It is good practice 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

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

- o Are they close to their nominal values?
- o Do all the integrated circuits in the suspected area have the proper voltage on their power pins?
- o Is there excessive ripple in any of the DC voltage lines?
- o If so, verify the regulator and associated components, working backwards toward the input power source.
- o 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 symptoms.

- o Are any components discolored?
- o Does something smell burned?
- o Do any of the parts seem 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 circuit 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 the 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 delating with the apparent problems.

7.3.1 Symptom: Controller appears dead

If the controller responds to initial power application with the MULT, STA, SEND and CON LEDs 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 millamperes continuous 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 Verify 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 transients appear as ringing and 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 Pin 22
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 on 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 factor 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 signal is present, but you cannot successfully 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 Verify 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 the 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 transmissions when using an SSB transmitter:

- o Verify 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 5.
- 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-232C 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 pins 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 (hex) and XFLOW to OFF, the controller will not send data to the terminal unless its DTR is asserted.

If the 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 U19 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.

(This page intentionally left blank)

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 individuals for comment and as a reference for software development. Changes should be expected. Please check the AMRAD Newsletter for announcements of later versions.

History

Over the years there have been several protocols suggested for use at layer 2 of the ISO Open System Interface Reference Model (OSI-RM) over Amateur Radio. The one 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 its 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 protocol and see if the protocol could be "enhanced" to work properly over the amateur radio environment. After reviewing the various protocols around and talking with people in the computer networking industry, we decided to push the X.25 standard, modified to allow a larger address field. At about this time, a group of amateurs in New Jersey 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 this protocol. In order to not step on anyone's toes, it was decided to call the protocol AX.25, which stands for Amateur X.25.

The next step in the evolution of AX.25 was that in October of 1982, AMSAT hosted a gathering of some of the 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 compatible protocol. AX.25 was then modified 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

This protocol conforms with the ISO Recommendations 3309, 4335 (including DAD 1&2) and 6256 high-level data link control (HDLC) and uses some terminology found in these documents.

This protocol also conforms with ANSI X3.66, describing ADCCP, balanced mode.

This protocol is written to work equally well in either half- or full-duplex amateur radio environments.

This protocol has been written to work equally well for either point-to-point connections, or connections made thru a larger device, such as a metropolitan network controller (MNC).

This protocol does allow the establishment of more than one layer 2 (link layer) connection per device, if the device is so capable.

This protocol also 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.

Most layer 2 protocols assume that one large device (generally called a DCE, or data circuit-terminating equipment) is connected to several smaller devices (usually called a DTE, or data terminating equipment). AX.25 assumes that both ends of the link are balanced, thereby eliminating the two different classes of device.

Frame Structure

Level 2 packet radio transmissions are sent in small blocks, called frames. These frames are made up of smaller parts, called fields. Fig. 1 shows how the three types of frames are made up. Fig. 1 shows the frames in the same bit order that most packet articles show them. Unfortunately, this method has led to some confusion, since the least-significant bit (LSB) is to the left rather than to the right, as most people would ordinarily assume. I am pointing this out early in this paper to prevent mass confusion as I progress. Later on, I will switch to a hopefully more understandable way of showing the frame and its components.

Field Definitions

The frame is made up of several parts, called fields. Each of these fields is made up of an integral number of octets (or bytes), and serves a specific function.

Flag Field

Since amateur packet radio is a bit-oriented protocol, the only way to tell when one frame is over and another is starting for sure is to delimit each frame with a certain bit sequence both at the beginning and the end. This is the job of the flag field. A flag consists of a zero followed by six ones followed by another zero, or 01111110 (7E hex). Due to the bit stuffing mentioned above, the only time this sequence is allowed is at the beginning and end of a legitimate frame.

Address Field

The address field is used to identify both where the frame came from and what the destination of it is. In the CCITT recommendation X.25, this field is only one octet long. This permits at most 256 users per level 2 channel, and since some bits of this field were used for other purposes, the real number of users were about thirty per level 2 channel. Both the HDLC and ADCCP recommendations allowed the address field to be extended, so we decided to extend the address field per their recommendations in the amateur version of X.25 to include the callsigns of both the destination and source amateur radio stations.

The method used to extend the address field will be described shortly.

Control Field

The control field is used to identify the type of frame and control several attributes of the level 2 connection. It is one octet in length, and its encoding will be discussed in a following section.

PID-Field

The Protocol Identifier (PID) field is used only in information frames, and identifies what kind of layer 3 protocol, if any, is in use. Its encoding is as follows:

M	L	
S	S	
B	B	
xx00xxxx		Reserved at the moment.
xx01yyyy		AX.25 layer 3 implemented.
xx10yyyy		AX.25 layer 3 implemented.
11110000		No layer 3 implemented.
11111111		Escape character. Next byte contains more PID information.

Where:

1. An x indicates a "don't care" bit.
2. A y indicates all combinations used.

Information Field

The information field is used to convey the actual 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 should be an even multiple of octets long. Any information in the I field should be passed along the link totally transparently, except for any zero-bit insertion necessary to prevent flags from accidentally appearing in the I field.

Frame Check Sequence

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

Bit Stuffing

In order to assure that the flag sequence mentioned above doesn't accidentally appear anywhere else in a frame, as the frame is being sent it should be monitored, and if more than five contiguous ones are detected, a zero bit should be added between the fifth and sixth ones, eliminating the possibility of a flag appearing in the frame other than where it belongs. The receiver of five ones, a zero, and more ones should automatically eliminate the inserted zero before passing the data on.

Bit Order of Transmission

With the exception of the FCS field, all other fields in an AX.25 frame should be sent starting with the least-significant bit. In accordance with HDLC practices, the FCS should be sent most-significant bit first.

Frame Abort

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

Invalid Frames

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

Address Field Encoding

The address field of all frames should be encoded with both the destination and source amateur callsigns of the frame. If a level 2 amateur "repeater" is to be used, its callsign should also be in the address field. AX.25 follows the HDLC recommended method of extending the address field in order to fit all this information into the address field.

Basically, the way the HDLC address field is extended beyond one octet is to reserve the least-significant bit of each octet for what is called an "extender bit". This bit is set to zero if the next octet contains more address field information, and is set to one if this is the last octet. To make room for this extender bit, the amateur radio call sign information is shifted one bit to the left.

The actual encoding techniques for both non-repeater and repeater operation follows.

Non-Repeater Address-Field Encoding

If a level 2 repeater is not being used, the address field is encoded as shown in Fig. 2. The destination address is the callsign of the amateur radio station that the frame is addressed to, while the source address contains the amateur call sign who sent the frame. These call signs are the call signs of the two ends of a level 2 AX.25 link only, not of any other station, such as the destination of a packet going thru an intermediary link. Those addresses should be in a higher layer, not layer 2.

A1 thru A14 are the fourteen octets that make up the two address sub-fields of the address field. The destination sub-address is seven octets long (A1 thru A7), and is sent first. This will allow the receivers of the frame time to check the destination address sub-field and see if the frame is for them while the rest of the frame is being received. The source address sub-field is then sent in octets A8 thru A14. Both of these sub-fields are encoded in the same manner, except for the last octet having the HDLC address extender bit set. Since they are basically the same, only the destination sub-address will be outlined.

There is an extra octet at the end of each address sub-field that allows room for a Secondary-Station Identifier (SSID) and three additional bits for future expansion. The SSID field allows an Amateur Radio operator to have more than one packet radio station. This is useful when an amateur wants to put up a repeater in addition to his regular station for example.

Appendix A shows a typical AX.25 frame in the non-repeater mode.

Destination Sub-Field Encoding

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

Octet	ASCII	Bin. Data	Hex Data
A1	W	1101011101	AE
A2	B	1100001001	84
A3	4	1011010001	68
A4	J	1100101001	94
A5	F	1100011001	8C
A6	I	1100100101	92
A7	SSID	10RRSSID01	

Bit Position--> 76543210

Fig. 3. Destination Field Encoding

Where:

1. The top octet (A1) is the first octet sent (sort of like popping it off the top of the stack), with bit 0 of each octet being the first bit sent, and bit being the last bit sent.
2. The first (low order or bit 0) bit of each octet is the HDLC extender 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. If they aren't implemented, they should be set to one.
4. The characters of the callsign should be standard seven-bit ASCII (upper case only) before being shifted left to make room for the extender bit. If the callsign is less than six characters long, it should be padded at the trailing end with ASCII spaces between the end of the callsign and the SSID octet.
5. The SSID portion of the last octet has been intentionally left vague at this point, and is left up to the individual station to assign. The only recommended restriction is to reserve the all-ones condition (1111) for an all-call SSID in case one wants to reach an amateur but doesn't know what SSID that amateur operates under.

Level 2 Repeater Address-Field Encoding

If a frame is to go thru a level 2 amateur packet repeater, there is an additional address sub-field added to the end of the address field. This additional sub-field contains the call sign of the repeater to be used. This will allow more than one repeater to share the same RF channel, which has been a problem with the older protocols. If this field exists, the last octet of the source sub-field has its extender bit set to zero, indicating that more address-field data follows. The repeater address sub-field is encoded in the same manner as the destination and source address sub-fields, except for one bit in the last octet, called the "H" bit. The H bit is used to indicate whether a frame has been repeated or not. This is necessary to prevent someone from potentially receiving two identical frames, the one going to the repeater, and the one coming back from the repeater. Fig. 4 shows how the repeater address sub-field is encoded.

Appendix B is an example of complete frame on its way back from a repeater.

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

Fig 4. Repeater Address Encoding

Where:

1. The top octet is the first octet sent, with bit 0 being sent first, bit 7 sent last of each octet.
2. As with the source and destination address sub-fields discussed above, bit of each octet is the HDLC address extender bit, which is set to zero on all but the last address octet (A21) where it is set to one.
3. The "R" bits are reserved just like in the source and destination sub-fields.
4. The "H" bit is the has-been-repeated bit. It is set to zero on a non repeated frame, and set to one by the repeater when the frame has been repeated.

It should be noted that some of the advantages of this addressing scheme are mentioned in Appendix C.

Control Field Formats

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

The control fields used in AX.25 use the CCITT X.25 control fields for balanced operation, 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.

Control Field Type	Control Field Bits
	7 6 5 4 3 2 1 0
I Frame	N(R) IP/FI N(S) 0
S Frame	N(R) IP/FI S SI 0 1
U Frame	M M M IP/FI M MI 1 1

Fig. 5. Control Field Formats

Where:

1. Bit 0 is the first bit sent, bit 7 is the last bit sent of the control field.
2. N(S) is the send sequence number (bit 2 is the LSB).
3. N(R) is the receive sequence number (bit is the LSB).
4. The "S" bits are the supervisory function bits, and their encoding is discussed below.
5. The "M" bits are the unnumbered frame modifier bits and their encoding is discussed below.
6. The P/F bit is the Poll/Final bit. Its function is described in more detail shortly.

Control Field Definitions

Information Frame Control Field

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 the section regarding flow control.

Supervisory Frame Control Field

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 don't have an information field, the sender's send variable and the receiver's receive variable are not incremented for S frames.

Unnumbered Frame Control Field

Unnumbered frames are distinguished by having both bits 0 and 1 set to one. U frames are responsible for maintaining control over the link beyond what is accomplished with S frames. They are also responsible for the establishment and tearing down of the link. U frames also allow for the transmission and reception of information outside of the normal flow control. Some U frames may contain information fields.

Control Field Parameters

Sequence Numbers and Variables

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

Send State Variable V(S)

The send state variable is an internal variable that 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.

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.

Receive State Variable V(R)

The receive state variable is an internal variable that 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.

Received Sequence Number N(R)

The received sequence number is in both 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 I frames up to and including N(R)-1.

Poll/Final (P/F) Bit

The P/F bit may be 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.

Control Field Encoding

Information Frame Control Field

The information frame control field is encoded as shown in Fig. 6. These frames are sequentially numbered to maintain control of their passage over the link level connection.

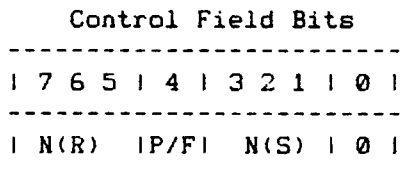


Fig. 6. I Frame Control Field

Supervisory Frame Control Field

The supervisory frame control fields are encoded as shown in Fig. 7. In AX.25, S frames are used only as responses to other frames.

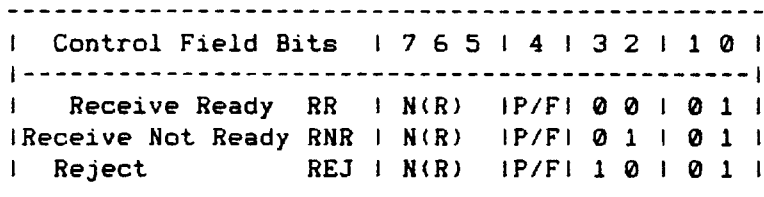


Fig. 7. S frame control Fields

Receive Ready (RR) 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 $N(R)-1$.
3. To clear a previously set busy condition created by an RNR command having been sent.

It should be noted that the status of the other side of the link can be requested by setting the poll bit.

Receive Not Ready (RNR) Response

Receive not ready is used to indicate to the sender of I frames that receiver 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 in between 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 P/F bit can be used within the RNR frame to interrogate the status of the other side of the link.

Reject (REJ) 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.

As with the other supervisory responses, the P/F bit may be used in the REJ frame.

Unnumbered Type Frames

Unnumbered frame control fields are either commands or responses. This standard follows X.25 as much as possible. The only deviation from X.25 is in the addition of the Unnumbered Information (UI) frame from ADCCP. X.25 is designed to work with in full-duplex systems with only one main device (DCE) and potentially many users (DTEs).

Amateur Radio packet systems differ greatly on both of these respects. Not only is Amateur Radio packet networking done in a half-duplex rf environment, but many DCE/DTE links may be sharing the same channel. Many amateurs have rejected the use of X.25 as a result of these problems. X.25 can easily be enhanced so that it will perform properly over amateur radio.

Fig. 8 shows the layout of U frames implemented within this standard.

Control Field	Type	Control Field Bits							
		7	6	5	4	3	2	1	0
Set Asynchronous	Cmd	0	0	1	P	1	1	1	1
Balanced Mode-SABM									
Disconnect-DISC	Cmd	0	1	0	P	0	0	1	1
Disconnected Mode-DM	Res	0	0	0	P/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	Res	0	0	0	P/F	0	0	1	1

Fig. 8. U Frame Control Fields

Set Asynchronous Balanced Mode (SABM) Command

The SABM command is used to place 2 stations in the asynchronous balanced mode. This is a balanced mode of operation known as LAP B where DCEs and DTEs are treated as equals.

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

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. Any outstanding I frames will remain outstanding.

Disconnected Mode (DM) Response

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

A DCE or DTE in the disconnected mode will respond to any command other than a SABM with DM response with the P/F bit set to 1.

Unnumbered Acknowledge (UA) Response

The UA response frame is sent to acknowledge the reception and acceptance of a U frame command. A received command is not actually processed until the UA response frame is sent. An information field is not permitted in a UA frame.

Unnumbered Information (UI) Frame

The unnumbered information frame 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 wiped out, there is no way to recover it.

The UI frame is not defined in X.25. It has been taken from ADCCP to allow uncontrolled information to flow thru the link without interfering with a next higher layer.

Link Error Recovery

There are several link-level errors that are recoverable without tearing down the connection. These error situations may occur as a result of malfunctions within the DTE or DCE, or if transmission errors occur.

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.

Device Busy Condition

When a DTE or DCE becomes temporarily busy, such as when receive buffers are full, it will send a receive not ready (RNR) frame. This tells the other side of the link that the device 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.

Send Sequence Number Error

If the send sequence number, $N(S)$, of an otherwise error-free received I 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 can still be performed, such as checking the P/F bit. Because of this updating, the retransmitted I frame may have an updated P bit and $N(R)$.

Reject (REJ) Error Recovery

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

A device receiving the REJ command will clear the error by sending over the I frame indicated in $N(R)$ of the REJ command frame.

Time-out Error Recovery

When a transmission abnormality wipes out a single I frame, or the last I frame of a group, there is no way of telling this immediately, since the receiver does not necessarily know something was sent until another frame is sent resulting in an out-of-sequence error. To cope with this situation better, some form of time-out delay will be incorporated by the sender after it sends out a frame. This time-out timer is started at the time a frame is sent, and stopped by the reception of an acknowledgement for the sent frame. If the timer times out before an acknowledgement is received, any unacknowledged frames are retransmitted. The delay is an agreed-upon amount that will vary with the type of rf medium and signaling speed used.

Rejection Error

A rejection error condition occurs when an error-free received frame has one of the following problems:

1. An invalid command or response control field.
2. An invalid frame format.
3. An Invalid N(R).
4. An information field that exceeds the maximum the device can accept.

Once a rejection error occurs, no more I frames are accepted (with the exception of the P/F bit still usable) until the error is resolved. The error condition is reported to the other side of the link by sending a FRMR response frame.

Primary/Secondary versus Balanced Operation

There are two basic classes of link-level connections. The first, known as Link Access Procedure (or LAP) is often called an unbalanced service where the DCE is considered the primary (or master) devices and the DTEs are considered secondary (or slave) devices. The second class of service is known as LAPB, Link Access Procedure Balanced. In this service both devices are treated as equals as far as connection requests and other types of commands. There is still only one DCE and potentially many DTEs, but both ends can command the link equally.

Primary/Secondary (LAP) Operation

LAP is the older style of link control, where most of the intelligence was assumed to be in a large main frame (the DCE) and the end users were just using smart terminals (the DTEs). Since network software can have a lot of overhead, it made sense at the time to put most of the overhead in the big computer, and just enough smarts to make the link work in the terminals.

Balanced (LAPB) Operation

LAPB is a slightly modified version of LAP. It has been changed to allow the two sides of a link to operate in a more balanced manner. In the official version of X.25 there is still only one DCE to potentially many DTEs, but the two can operate more as equals than master and slave.

LAPB is what this document describes for use over Amateur Radio packet networks. Even when there is a network controller overseeing the network operation, the balanced link procedure will enhance operation.

Connection Operation

In amateur radio network operations, it would be very helpful if one level 2 protocol would work with the various RF systems in use. An example of this is the difference in operation between a simple two-station link, and multiple stations operating thru a network controller. Obviously, when a network controller exists, it should be considered the DCE, while the other stations connecting to it would be the DTEs. A simple two-station connection is another matter. To this type of connection the station requesting a connection should always be considered the DTE, while the device that is receiving the connection request should operate as the DCE. This simple rule should eliminate any ambiguity that might otherwise occur under these conditions.

NOTE There are a couple minor changes from the official X.25 standard in the protocol recommended here. These changes are done only as absolutely necessary to work over the shared RF media. Since X.25 was written to work so that one DCE talked with many DTEs over a closed network, it cannot properly cope with a channel where there may be many DCEs linked to many DTEs. Some amateurs have thrown X.25 out because of this problem. It seems to take just a couple minor changes in the initial link set-up procedure to make X.25 work properly over amateur radio. Where these changes are made, both the original X.25 procedure and the recommended amateur procedure will be noted.

LAPB Procedures

The following describes the procedures used to set-up, use, and disconnect a balanced link between a DTE and DCE. These procedures have been taken from X.25 and conform very closely to that standard, except where it was necessary to change due to the radio environment.

Address Field Operation

All transmitted frames shall have address fields conforming to above-mentioned rules. All frames should have both the destination device and the source device addresses in the address field, with the destination address coming first. This will allow 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 however, if point to multi-point operation is allowed. This will be discussed further under link operations.

LAPB Connection Establishment

When a device (either a DCE or DTE) wishes to connect to another device, it will send SABM command frame to that device and start a time-out timer (T1). If the other device is there and able to connect, it will answer with a UA response frame and at the same time 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 device requesting the connection to abort the T1 timer and set its internal state variables to 0 also.

If the other device doesn't respond before T1 times out, the device requesting the connection will resend the SABM frame, and start T1 running again. This trying to establish a connection will continue until the requesting device has tried unsuccessfully a number of times. That number (N1) is variable, depending on the frequency of operation, type of transmission (eg. terrestrial vs. satellite), and the signaling speed in use. N1 will be discussed in another section.

Information Transfer

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

Sending of I Frames

Whenever a station 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.

The station should not transmit any more 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 exceeded and errors could result.

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

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

Receiving I Frames

If a device 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 manner:

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 an 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 an RR frame with $N(R)$ equal to $V(R)$.

If the device 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 station receiving the busy condition indication should poll the sender of the busy indication periodically until the busy condition disappears.

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.

When an I frame is received with a correct FCS, but its send sequence number does not match the current receiver's receive state variable, the frame should be discarded and a REJ frame should be sent with a receive sequence number equal to one higher (modulo 8) than the last correctly received I frame. Any out-of-sequence received I frames should be handled in this manner. The received state variable and poll bit in such a discarded frame should be checked before throwing it away, and take any action needed depending on the condition of them.

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 acknowledgment of outstanding sent I frames. The T1 timer should be reset if the received frame actually acknowledges previously unacknowledged frames. If the T1 timer is reset, 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.

Receiving Reject

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

1. If the device 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 device is operating on a full duplex channel transmitting a U or S frame when it receives a REJ frame, it may finish sending the U or S frame and then retransmit the I frame(s).
3. If the device 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 device may send just the one I frame outstanding, or it may send more than one if any 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 device receives a REJ frame with the poll bit set, it should respond with either an RR or RNR frame with the final bit set before retransmitting the outstanding I frame(s).

Receiving an RNR Frame

Whenever a device receives an RNR frame, it may transmit or retransmit the I frame whose send sequence number equals that of the received sequence number indicated in the RNR control field. If timer T1 runs out after the RNR was received, the waiting acknowledgement procedure listed 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 station. No I frames other than the one mentioned above may be sent out before the busy condition is cleared.

Sending a Busy Indication

Whenever a device enters a busy condition, it will indicate this by sending an RNR response at the next opportunity. While the device is in the busy condition, it may receive and process S frames, and if a received S frame has the F bit set to one, the device should send a RNR frame with the F bit set to one at the next possible opportunity. To clear the busy condition, the device 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.

Waiting Acknowledgement

The device should maintain an internal retransmission count variable which is set to zero whenever another I frame is acknowledged (either thru the reception of a UA or RNR frame, or when a received I or S frame has an N(R) higher than the last received N(R), showing the acknowledgement of additional I frames).

Any time the timer T1 runs out, the device will reenter the timer recovery condition, the retransmission count variable will be incremented by one, and another internal variable (X) will be set to the current send state variable value.

The device will then restart the T1 timer, set its receive state variable to the last receive sequence number, and retransmit the corresponding I or S frame with the P bit set to one.

The timer recovery condition is cleared when the device receives a valid S frame with the bit set to one.

If the device receives an S frame with the F bit set to one and N(R) within the range from the current send state variable to X mentioned above inclusive while in the timer recovery condition, this condition will be cleared, and the send state variable will be set to the N(R) received.

If the device receives an S frame with the F bit set to zero but otherwise the same condition as the last paragraph, the timer recovery condition will NOT be cleared. The received N(R) may be used however to update the send state variable. The device may keep the last I frame transmitted (even if it was acknowledged) to be retransmitted with the P bit set to one if timer T1 expires at a later time.

Once the retransmission count variable reaches N2, the device should proceed to the resetting procedures outlined below.

Link Disconnection

When in the information-transfer phase, either device may initiate a link disconnection by sending a DISC frame. It should then start its T1 timer, and wait for a response. If the proper response doesn't come before T1 times out, it should send the DISC frame again and restart T1. If this happens N2 times, the device should enter the disconnected state.

When a DISC frame is received, the receiver should return a UA response frame, and enter the disconnected state.

Disconnected State

After having sent a DISC frame and received a UA, or receiving a DISC and having sent UA, the device will enter the disconnected state.

In the disconnected state, the device may initiate a link set-up as outlined in connection establishment above. 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.

Any station receiving a DISC command while in the disconnected state should send back a DM response frame.

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

When the device enters the disconnected state after an error condition or if it has recovered from an internal error condition by coming up in the disconnected state, it should indicate this by sending a DM response rather than a DISC frame. It should start the T1 timer when the DM is sent, and if T1 times out before getting a SABM or DISC frame back, it should send another DM frame, and restart T1. After retransmitting the DM frame N2 times, the device will remain in the disconnected state, and no other action will be taken.

Resetting Procedure

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

A device shall request a reset by sending an SABM frame. Upon receiving an SABM frame from a station previously connected to, the receiver of an SABM frame should send a UA frame back at the earliest opportunity. Both devices should then set their send and receive state variables to zero. Any busy condition that previously existed will also be cleared.

It is possible to initiate a disconnect procedure instead of resetting the link.

One device may ask the other to reset the link by sending a DM response frame. After the DM frame is sent, the sending device will then enter the disconnected state.

One device may ask the other to initiate link reset by transmitting a FRMR response frame.

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

The device that sent the FRMR frame should start the T1 timer when the FRMR is sent. If above mentioned frames are not received before the timer runs out, the FRMR frame should be retransmitted, and the T1 timer restarted as described in the waiting acknowledgement section above. If the FRMR is sent N2 times without success, the link should be reset.

Rejection Conditions

A device should initiate the link-reset procedure when a frame is received with the correct FCS and address field during the information transfer phase with one or more of the following conditions:

1. The frame is not known as a command or response to the device.
2. The information field is invalid (as an example is longer than 256 octets).

A device will initiate a reset procedure whenever it receives a DM or FRMR response frame during the information transfer phase.

A device may initiate a reset procedure also whenever it receives a UA response frame or if it receives an unsolicited response frame with the F bit set to one.

Collision Recovery

Collisions in a Half-Duplex Environment

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

Collisions in a Full-Duplex Environment

Collisions in a full-duplex environment are not really frame collisions, but have more to do with the devices being pulled in two different directions at the same time.

Collisions of Unnumbered Commands

If the sent and received U command frames are the same, both devices should send a UA response at the earliest opportunity, and both devices should enter the indicates state.

If the sent and received U commands are different, both devices should enter the disconnected state, and transmit a DM frame at the earliest opportunity.

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, so there isn't any confusion when a DM frame is received.

Connectionless Operation

In Amateur Radio circles, there is a type of operation that isn't really feasible using level 2 connections. This operation is the round-table, where several amateurs may be engaged in one conversation. The only way to accomplish this in a connected mode would be to have everyone cross-connected with each other. This would require a separate frame to be sent to each member of the round-table every time someone says something. Obviously, this mode is not practical. The way most amateur packet radio enthusiasts have ended up implementing the round-table operation is outside the AX.25 connection, but still using the AX.25 frame structure. AX.25 does allow a special frame for this operation, called the Unnumbered Information (UI) frame. It is recommended that when this type of operation is in use, the destination address have a code word installed in it to prevent the users of that particular round-table from seeing all frames going thru 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 only receive frames 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 sub-field, so software could be written to automatically display who is making what comments.

Admittedly, this is a kludge to the level 2 AX.25 protocol. This type of operation really belongs at the next layer (layer 3, packet level) of operation, but until layer 3 is implemented, this appears to be an acceptable substitute.

Keep in mind that this mode is connectionless, so all transmitted frames should be of good quality, as there will be no requests for retransmissions of bad frames. Collisions will also occur, with the potential of losing the frames that collided.

List of System Defined Parameters

Timers

It is recommended that there are two timers used to maintain the integrity of the AX.25 layer 2 connection.

The first timer, T1, is used to make sure a device 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 baud rate used at level 1. T1 should be at least twice the time it would take to send a maximum length frame to the other end of the link, and get the proper response frame back from the other end of the link. This would allow time for the other end of the link to do some processing before responding.

The second timer, T2, is used whenever T1 isn't running to make sure that a supervisory frame is sent periodically to maintain link integrity. It also will vary dramatically depending on layer 1 constraints, and is subject for further study.

Maximum Number of Retrys (N2)

The maximum number of retrys is used in conjunction with the T1 timer. It will vary depending on the layer 1 in use, but will generally be sixteen.

Maximum Number of Octets in an I Field (N1)

The maximum number of octets allowed in the I field will be 256. There should also be an even multiple number of octets.

Maximum Number of I Frames Outstanding (k)

The maximum number of outstanding I frames at a time is seven. A smaller number may be used at any time, provided it is agreed upon ahead of time.

First
Bit Sent

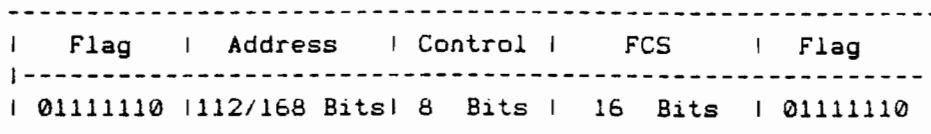


Fig. 1A. U and S Frame Construction

First
Bit Sent

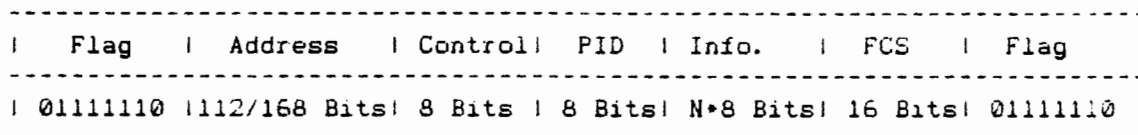


Fig. 1B. Information Frame Construction

First
Octet Sent

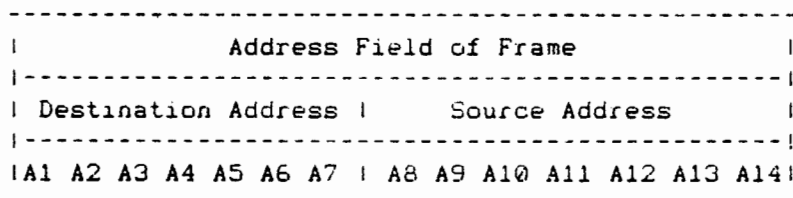


Fig. 2A. Non-Repeater Address Field Encoding

First
Octet Sent

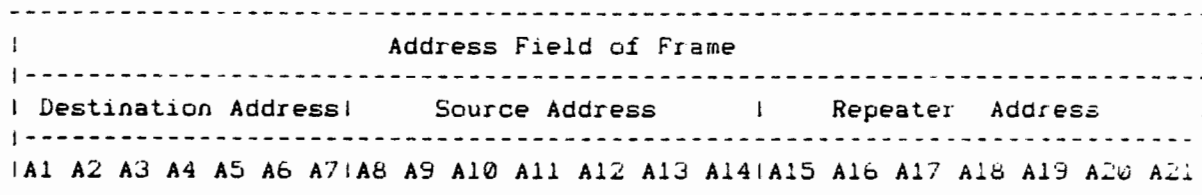


Fig. 2B. Repeater Address Field Encoding

Appendix A. Non-Repeater Frame Example

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	01100000	60
A8	W	10101110	AE
A9	B	10000100	84
A10	4	01101000	68
A11	J	10010100	94
A12	F	10001100	8C
A13	I	10010010	92
A14	SSID	01100001	61
Control	SABM	00111111	3F
PID	none	11110000	F0
FCS	part 1	XXXXXXXX	HH
FCS	part 2	XXXXXXXX	HH
Flag		01111110	7E

Appendix B. Repeater Type Operation

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	0	10011110	9E
A6	space	01000000	40
A7	SSID	01100000	60
A8	W	10101110	AE
A9	B	10000100	84
A10	4	01101000	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	SABM	00111111	3F
PID	none	11110000	F0
FCS	part 1	XXXXXXXX	HH
FCS	part 2	XXXXXXXX	HH
Flag		01111110	7E

Appendix C.

Advantages of the WB4JFI Addressing Scheme

Some of the advantages to using this addressing system are:

1. Every packet station will have a unique fixed address that doesn't change every time a new network is logged into.
2. Relocating to a new area won't cause major (or minor) problems.
3. Allows for more than 62 or 31 users at a time.
4. No local packet guru is needed to assign addresses with attendant concerns of backup and transfer during failure.
5. Direct or network operation requires no change of address.
6. All the problems with dynamic allocation/de-allocation are eliminated.
7. Reduces local co-network interference due to users in overlapping local network rf domains with the same address fields.
8. With every frame having both the destination and source addresses in them, it will be a lot easier to set-up and run multiple connections on the same data channel without having problems arise as to who is sending what frames to whom.
9. In round-table operation, every frame sent will have the source address imbedded in it, allowing automatic printing of the source of the frame.

Appendix D - Layer 2 AX.25 State Table - Part 1 of 2

State	I with Poll	I with out P	RR with Final	RR with out F	REJ with Final	REJ with out F
S1 Disconnected	DM	-	DM	--	DM	-
S2 Link Setup	--	-	-	--	--	-
S3 Frame Reject	FRMR	-	FRMR	--	FRMR	-
S4 Disconnect Rqst	--	-	-	--	--	-
S5 Information Xfr	RK	I	I	I	I	I
S6 REJ Frame Sent	RR, S5	I, S5	I	I	I	I
S7 Waiting Acknow.	RR	I	I, S5	I	I, S5	I
S8 Device Busy	RNR	RNR	I	I	I	I
S9 Remote Device Busy	RR	RR	I, S5	I, S5	I, S5	I, S5
S10 Both Devices Busy	RNR	RNR	I, S8	I, S8	I, S8	I, S8
S11 Waiting acknow. and Device Busy	RNR	RNR	I, S8	I	I, S8	I
S12 Waiting acknow. and Remote Busy	RR	RR	I, S5	I, S7	I, S5	I, S7
S13 Waiting Acknow. Both Devices Busy	RNR	RNR	I, S8	I, S11	I, S8	I, S11
S14 REJ Sent and Device Busy	RNR	RNR	I	I	I	I
S15 REJ Sent and Remote Busy	RR, S9	RR, S9	I, S6	I, S6	I, S6	I, S6
S16 REJ Sent and Both Devices Busy	RNR	RNR	I, S14	I, S14	I, S14	I, S14

Appendix D - Layer 2 AX.25 State Table - Part 2 of 2

State	RNR with Final	RNR with- out F	SABM either	DISC either	UA either	DM either	FRMR either
S1	DM	-	UA, S5	DM	-	SABM, S2	--
S2	--	-	UA	DM, S1	S5	S1	--
S3	FRMR	-	UA, S5	UA, S1	-	SABM, S2	SABM, S2
S4	--	-	DM	UA	S1	S1	--
S5	S9	S9	UA	UA, S1	SABM, S2	SABM, S2	SABM, S2
S6	S15	S15	UA, S5	UA, S1	SABM, S2	SABM, S2	SABM, S2
S7	S9	S12	UA, S5	UA, S1	SABM, S2	SABM, S2	SABM, S2
S8	S10	S10	UA	UA, S1	SABM, S2	SABM, S2	SABM, S2
S9	--	-	UA, S5	UA, S1	SABM, S2	SABM, S2	SABM, S2
S10	--	-	UA, S8	UA, S1	SABM, S2	SABM, S2	SABM, S2
S11	S10	S13	UA, S8	UA, S1	SABM, S2	SABM, S2	SABM, S2
S12	--	-	UA, S5	UA, S1	SABM, S2	SABM, S2	SABM, S2
S13	--	-	UA, S8	UA, S1	SABM, S2	SABM, S2	SABM, S2
S14	S16	S16	UA, S8	UA, S1	SABM, S2	SABM, S2	SABM, S2
S15	--	-	UA, S5	UA, S1	SABM, S2	SABM, S2	SABM, S2
S16	--	-	UA, S5	UA, S1	SABM, S2	SABM, S2	SABM, S2

APPENDIX B

"KISS" TNC SPECIFICATION

By Phil Karn, KA9Q

KISS TNC Specification (Phil Karn, KA9Q)**Introduction**

The KISS TNC protocol permits more efficient use of amateur packet radio controllers with host computers, particularly in the development of experimental protocols and multi-user servers (e.g., "bulletin boards").

Current TNC software was written for human users in mind. Commands and responses designed for human use are inefficient 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 the single host/TNC link. In addition, experimentation with new link-level protocols is greatly hampered; there may be no way to send or receive frames in the desired format without reprogramming the TNC.

The KISS protocol 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. 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 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.

The bulk of AX.25 (or another protocol) must now be executed 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.

Asynchronous Frame Format

The "asynchronous packet protocol" spoken between the host and TNC is very simple, since its only function is delimiting 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.

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 prepending it to 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.

Transparency

Frames are sent in eight-bit binary; if an 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", which causes 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 FESC 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.

The special characters are:

FEND (frame end)	300 (octal)
FESC (frame escape)	333 (octal)
TFEND (transposed frame end)	334 (octal)
TFESC (transposed frame escape)	335 (octal)

(ARPA Internet hackers will recognize this protocol as identical to SLIP, a popular method for sending IP datagrams across ordinary dialup modems).

Control of the Raw 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 parameters, namely the transmitter keyup delay and the transmitter persistence variables.

It is therefore necessary to distinguish between command and data frames on the host/TNC link. This is done by defining the first byte of each asynchronous frame between host and TNC as a "type" indicator. The following types are defined in frames to the TNC:

APPENDIX B - "KISS" TNC SPECIFICATION

Type	Function	Comments
0	Data frame	Rest of frame is data to be sent on the HDLC channel
1	TXDELAY	Second byte is transmitter keyup delay in 10 ms units
2	P	Second byte of frame is persistence parameter, p: 0: $p = (0+1)/256$, 255: $p = (255+1)/256 = 1.0$
3	SLOTTIME	Second byte of frame is slot interval in 10 ms units
4	TXtail	Second byte of frame is time to hold up the TX after the FCS has been sent (the time allowed to send the HDLC flag character; should be at least 2 for 1200 bps operation). In 10 ms units.

The following types are defined in frames to the host:

Type	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.)

Persistence

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

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 after 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.

Comments on this spec are welcome.

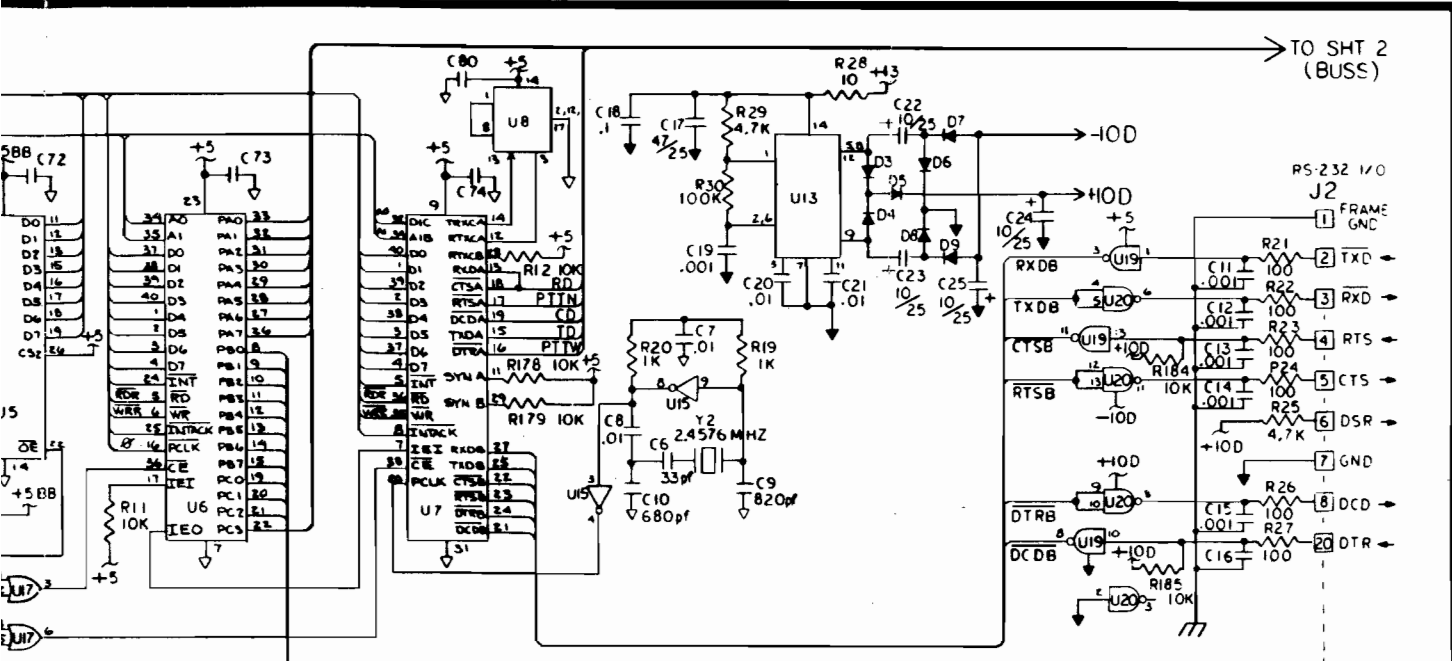
Phil Karn, KA9Q

(This page intentionally left blank)

APPENDIX C - DRAWINGS

APPENDIX D - WAVEFORMS

U20 75188
 U7 7530 ser
 U8 7536 par

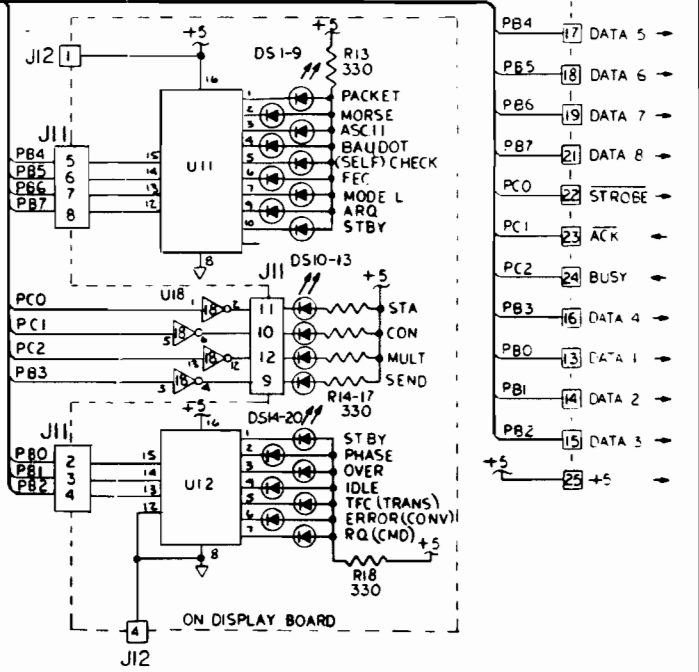


I.C. TABLE

REF. DESIG.	PIN NO.					TYPE
	GND	+13	+5	+5BB	+100	
U1	29	11				280A
U2	14	14	1, 28			27256
U3	14	14	1, 28			27128 (B)
U4, U5	14	14	28			6264
U6	7	23				28536
U7	31	9				28530
U8	2, 7	14				74LS393
U9	7	14				74LS164
U10	8	16				74LS139
U11, U12	8	16				7445
U13	7	14				556
U14	7	14				74LS11
U15, U39	7	14				74LS04
U17	7	14				74LS32
U18, U37, U38	7	14				7405
U19	7	14				MC1489
U20	7	14				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 μ F.
 RESISTORS 1/4 W, 5%.
 HOLES IN 4148.
 NOTCHES SHOWN IN OUT POSITION.
 CAPS ARE .1 BYPASS CAPS.
 USED REF. DESIG:
 36 Y2 DS22
 9 J12 JP6
 1 Q11
 3 SW2
 USED:
 17
 INSTALLED:
 RESERVED FOR FUTURE EXPANSION.

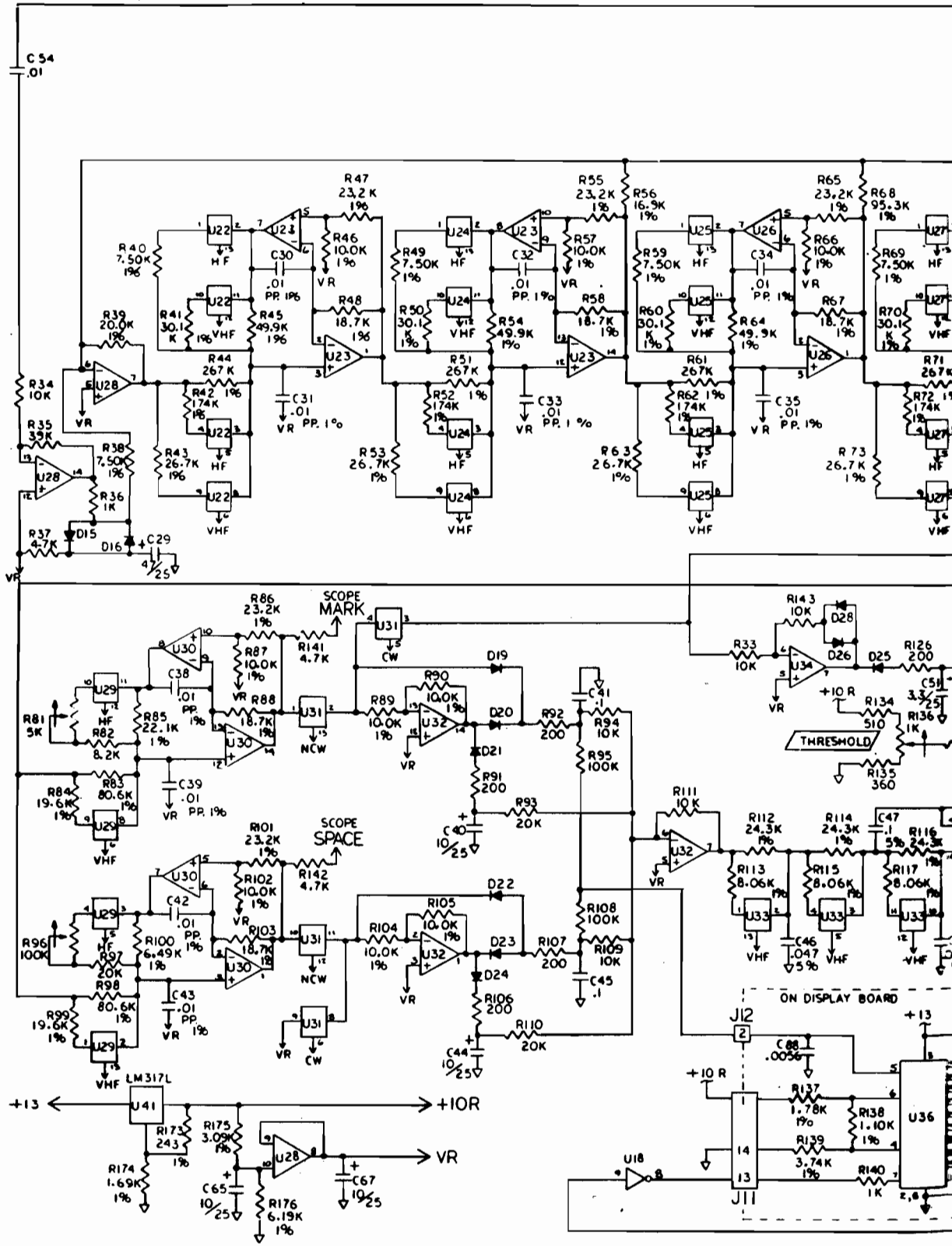


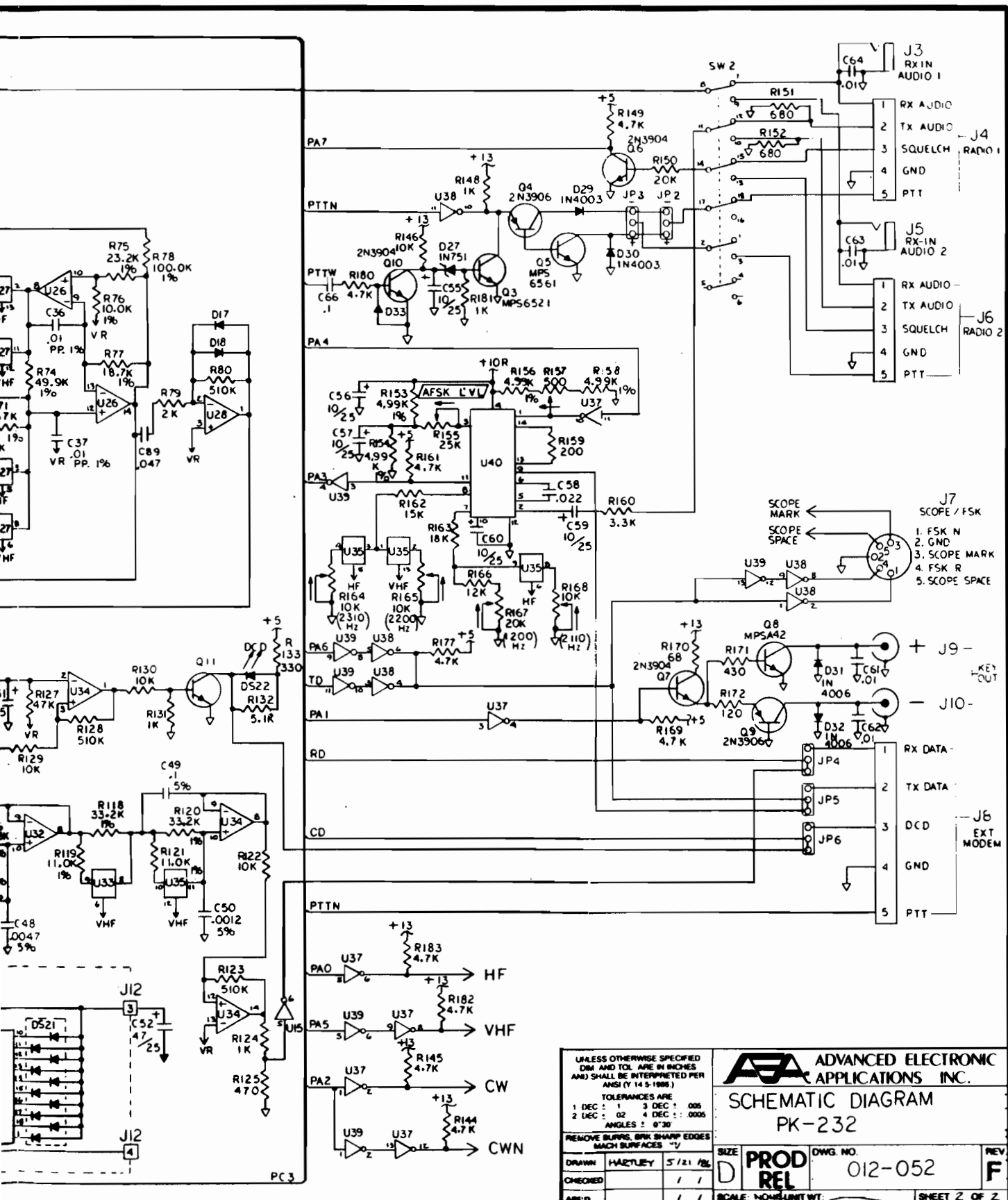
APPENDIX H

UNLESS OTHERWISE SPECIFIED DIM. AND TOL. ARE IN INCHES AND SHALL BE INTERPRETED PER ANSI (Y 14.5-1986)		ADVANCED ELECTRONIC APPLICATIONS INC.	
TOLERANCES ARE 1 DEC : 1 3 DEC : .005 2 DEC : .02 4 DEC : .0005 ANGLES : 0°30'			
REMOVE BURRS, BURN SHARP EDGES ON MACH SURFACES .001" ±		SIZE	DWG. NO.
DRAWN	HACTLEY	D	012-052
CHECKED		REL	
APPD		SCALE:	SHEET 1 OF 2

TO SHT 1
(BUSS) ←

H-2





UNLESS OTHERWISE SPECIFIED
DIM AND TOL ARE IN INCHES
AND SHALL BE INTERPRETED PER
ANSI (Y 14.5-1986)

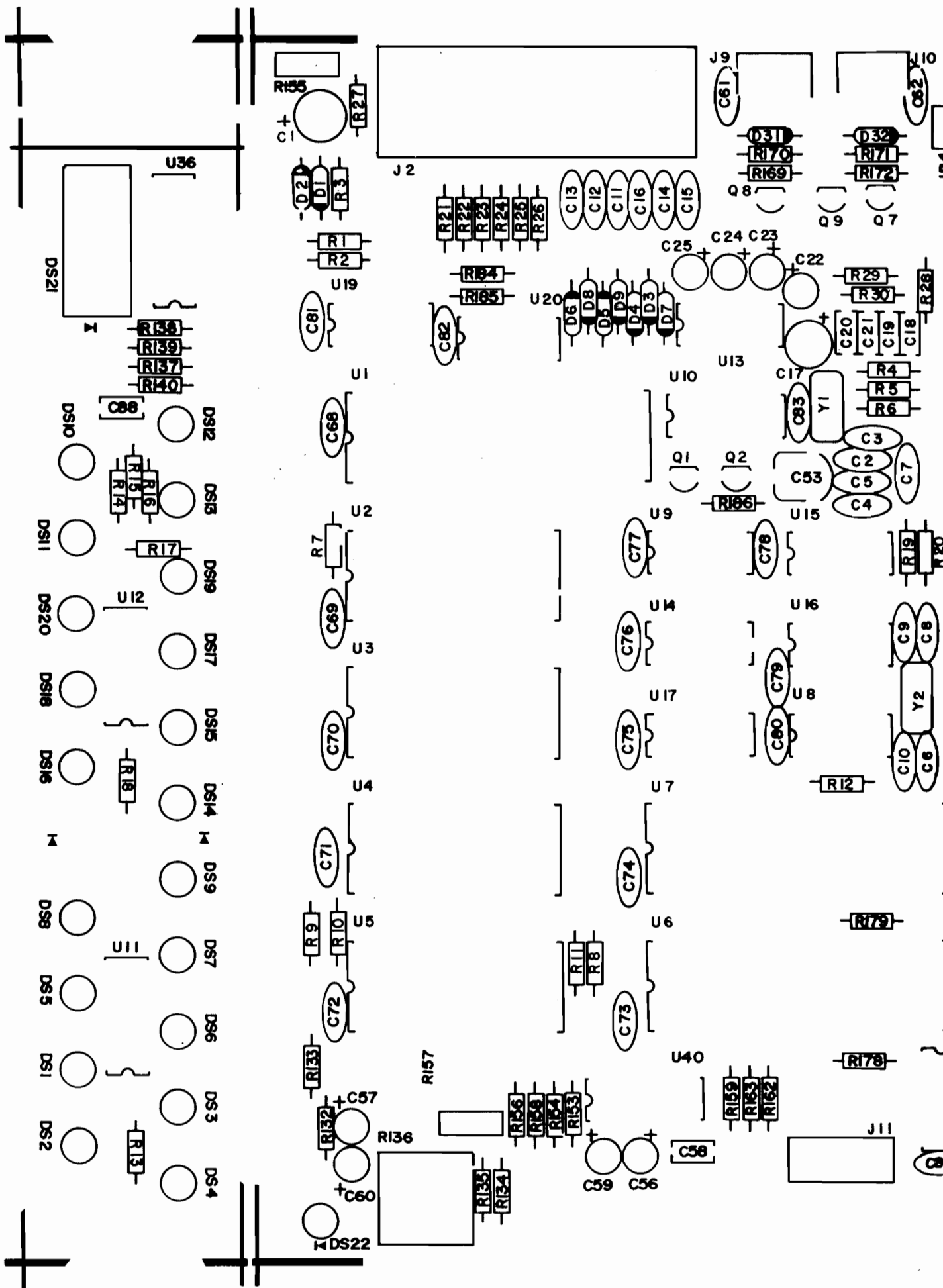
TOLERANCES ARE
1 DEC : 1 3 DEC : .005
2 DEC : .02 4 DEC : .0005
ANGLES : 0°30'

REMOVE BURRS, BRK SHARP EDGES
MACH SURFACES "V"

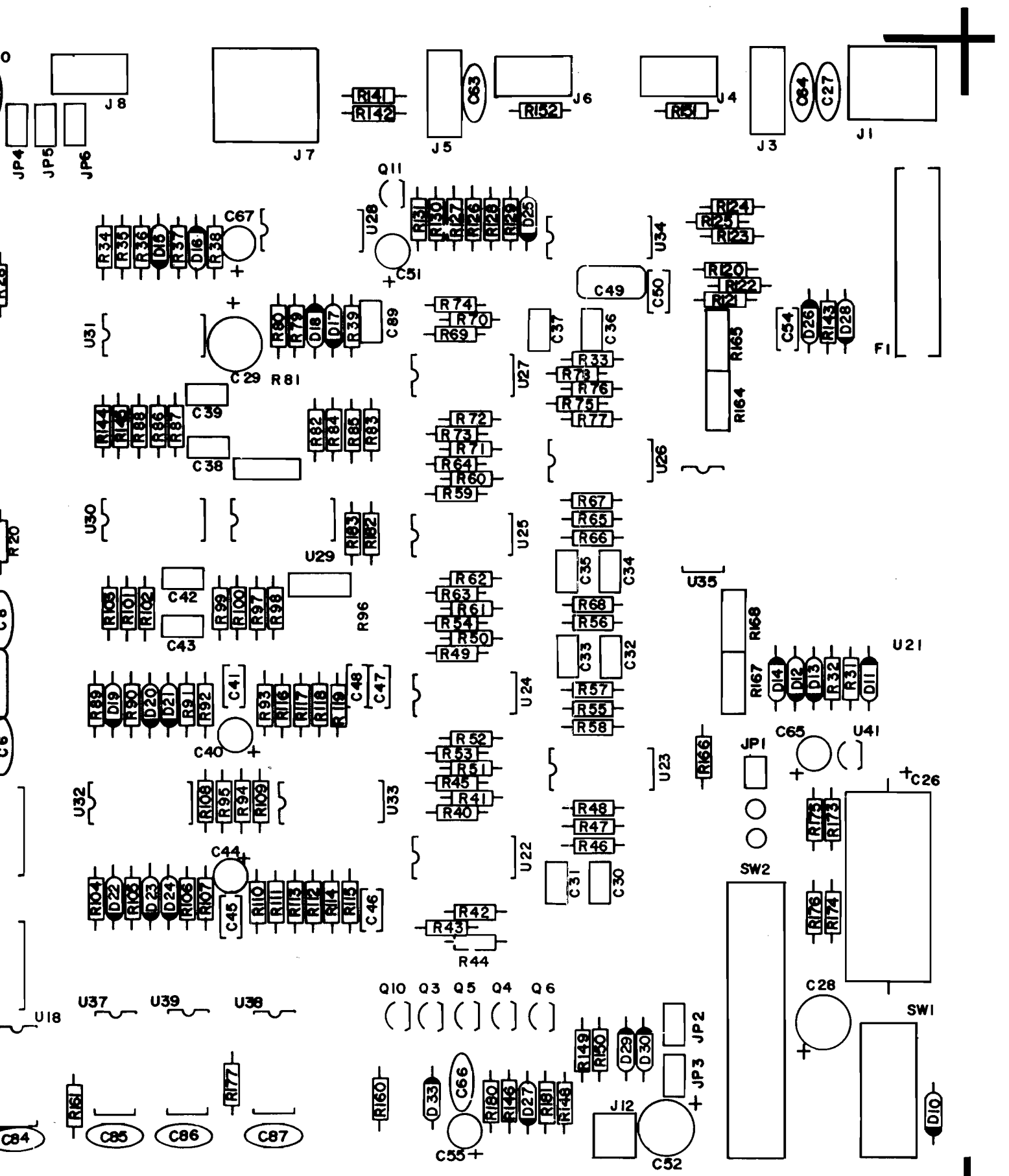
DRAWN	HARTLEY	5/21/86	SIZE	PROD	DWG. NO.	REV
CHECKED			D	REL	012-052	F
APP'D			SCALE: NONE/UNIT WT.			SHEET 2 OF 2

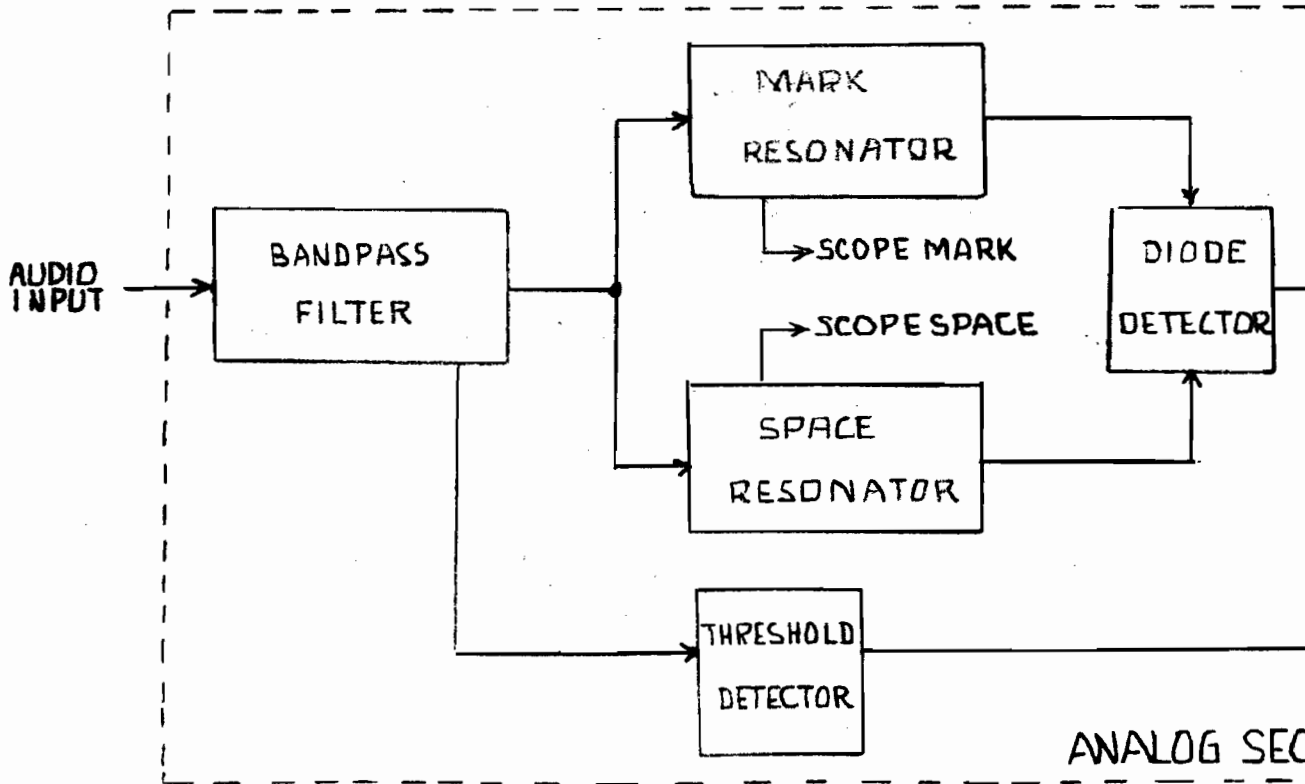
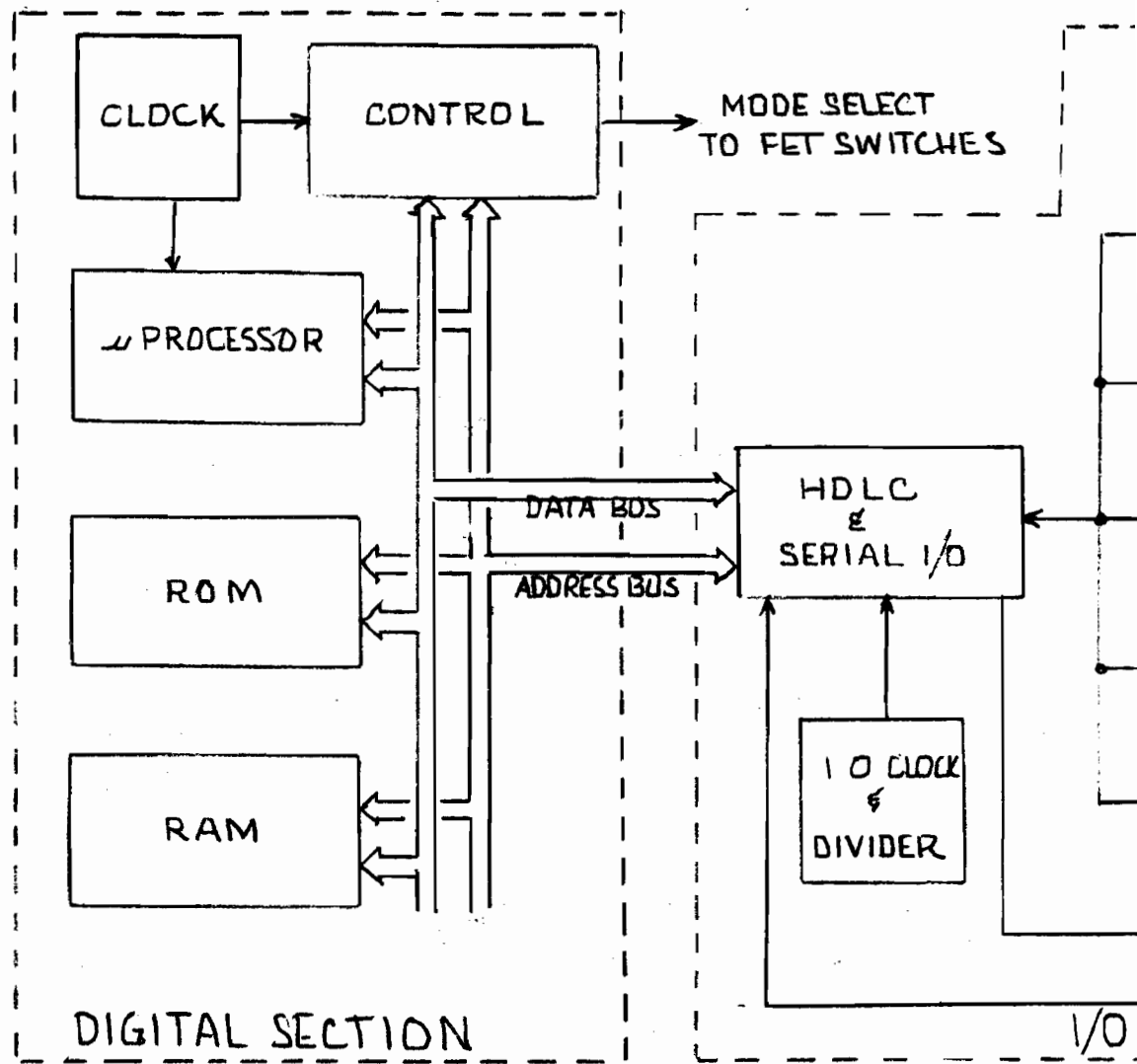
ADVANCED ELECTRONIC APPLICATIONS INC.

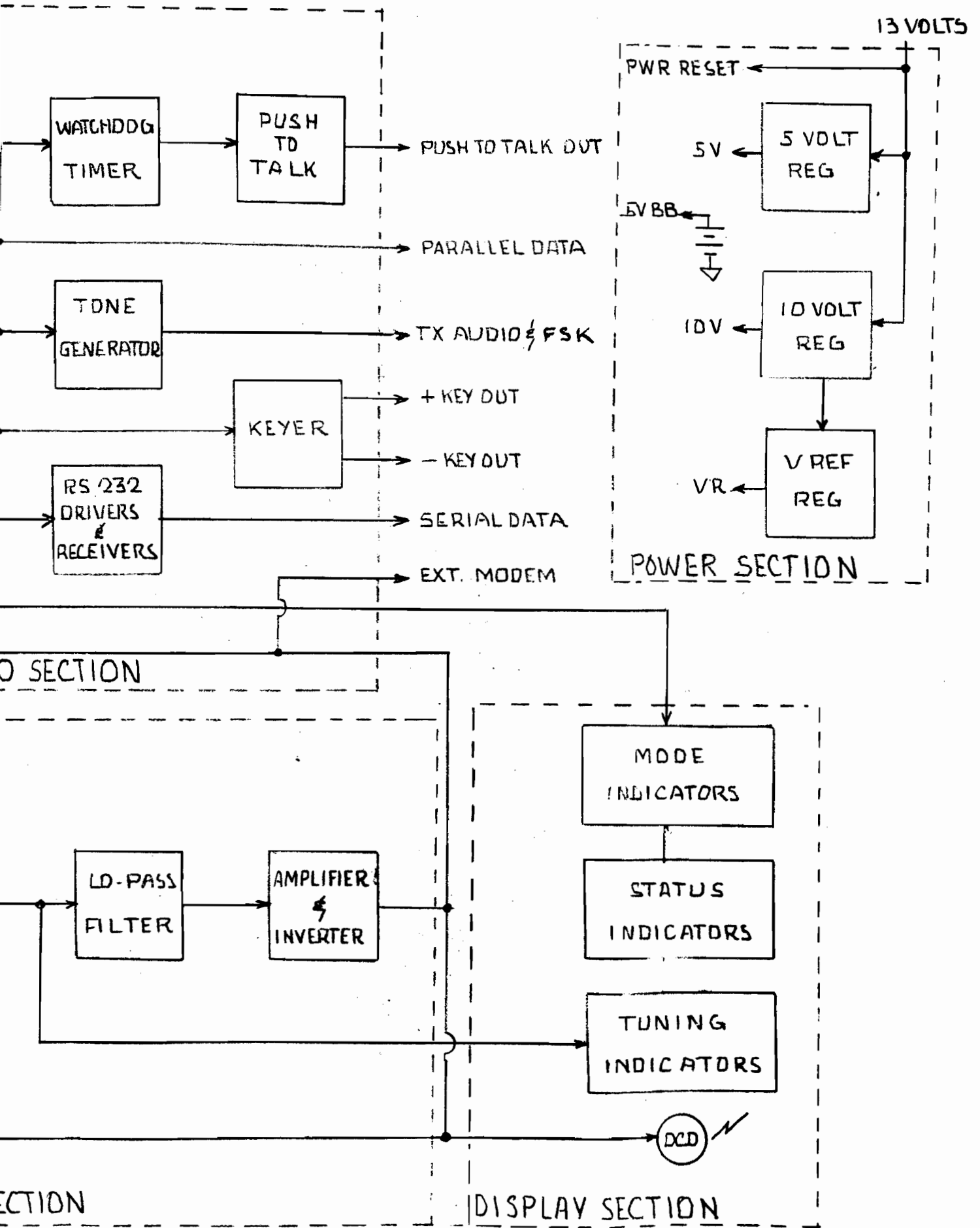
SCHEMATIC DIAGRAM
PK-232



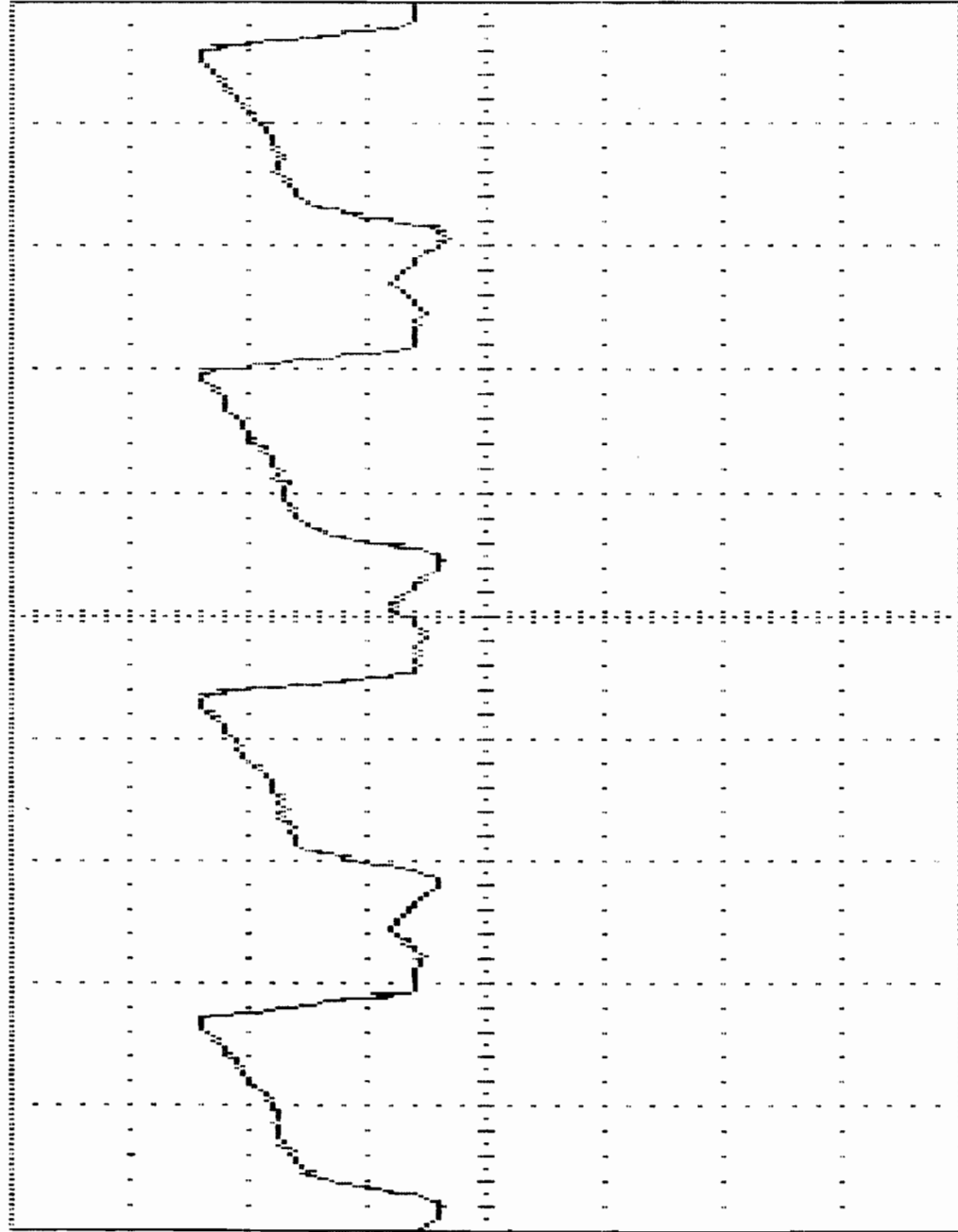
CCMP I.D.







PK-232 BLOCK DIAGRAM



Y1: **IN**(normal)
 2.0 V/div
 Offset: +0.00 V

Y2: OFF(normal)
 100 mV/div
 Offset: +0.0mV

TIMEBASE: $\frac{100}{10}$
 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 1/M/C2 F4: DEFINE CURSORS F8: NEXT MENU

Y1: 100mV (normal)
2.0 V/div
Offset: +0.000 V

Y2: OFF (normal)
100 mV/div
Offset: +0.000 V

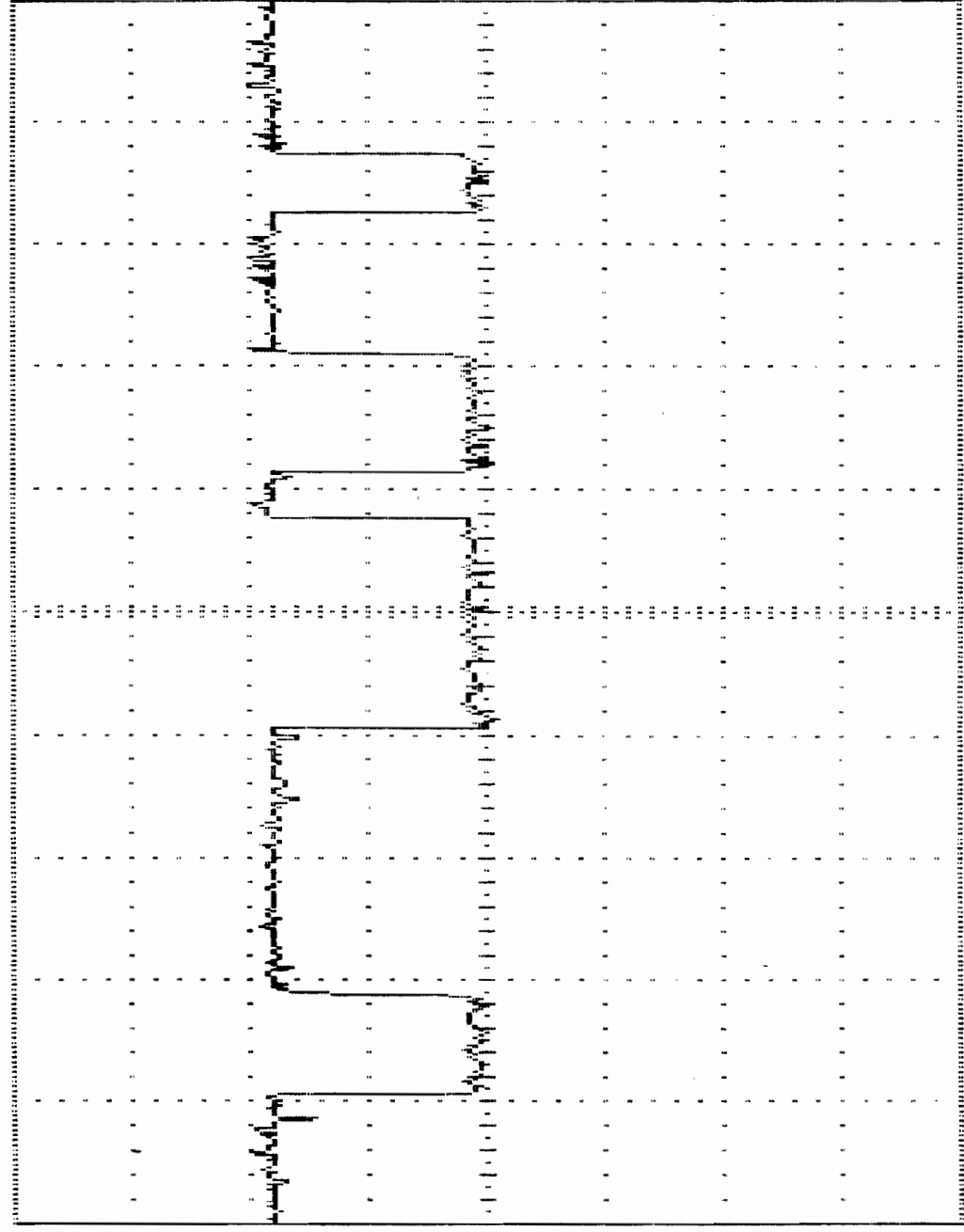
TIMEBASE: 1.0 μ s/div

TRIG SOURCE: Y1

TRIG SLOPE: (+)

TRIG LEVEL: +0.00 V

TRIG MODE: single triggered



F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR MW/C2 F4: DEFINE CURSORS F8: NEXT MENU

2. 01-30 A0 (typical substructure activity)

Y1: I_{X1} (normal)
2.0 V/div
Offset: +0.00 V

Y2: OFF (normal)
100 mV/div
Offset: +0.0mV

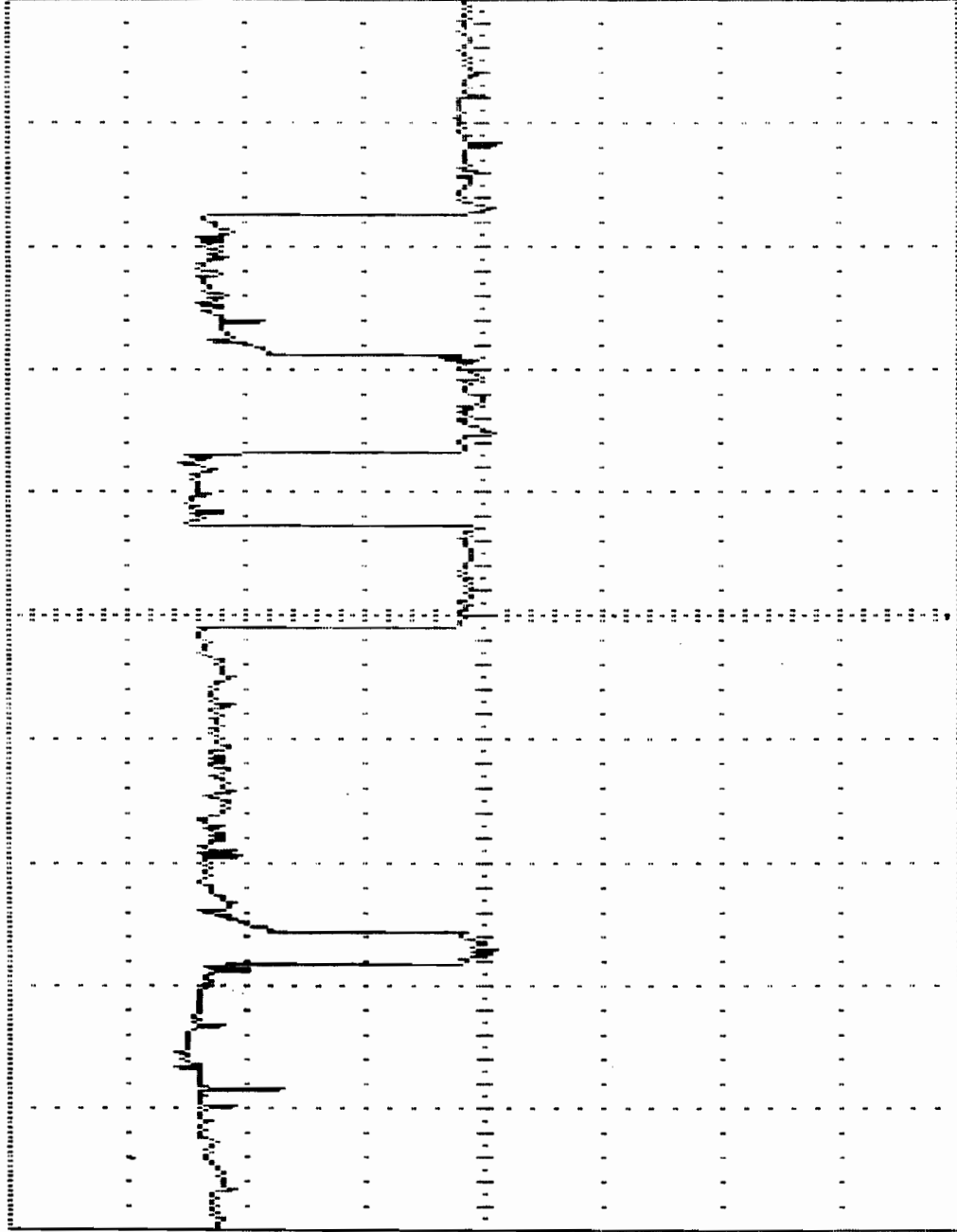
TIMEBASE: μ
1.0 μ s/div

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: 100mV (normal)
2.0 V/div
Offset: +0.00 V

Y2: OFF (normal)
100 mV/div
Offset: +0.0mV

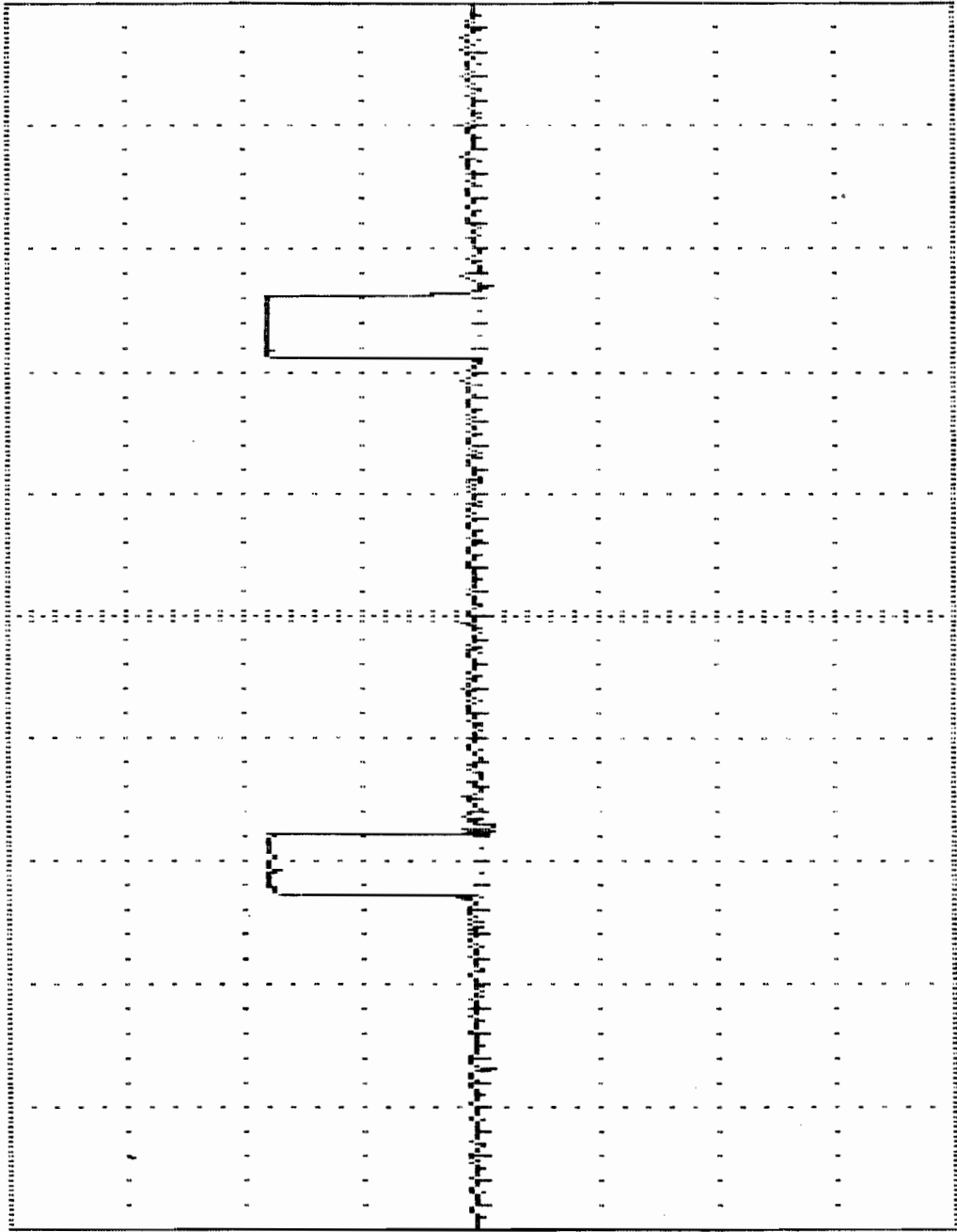
TIMEBASE: $2.0\text{ }\mu\text{s}$ /div

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

4. U20-20 CE (ROM chip enable)

Y1: 100mV (normal)
2.0 V/div
Offset: +0.00 V

Y2: OFF (normal)
100 mV/div
Offset: +0.00V

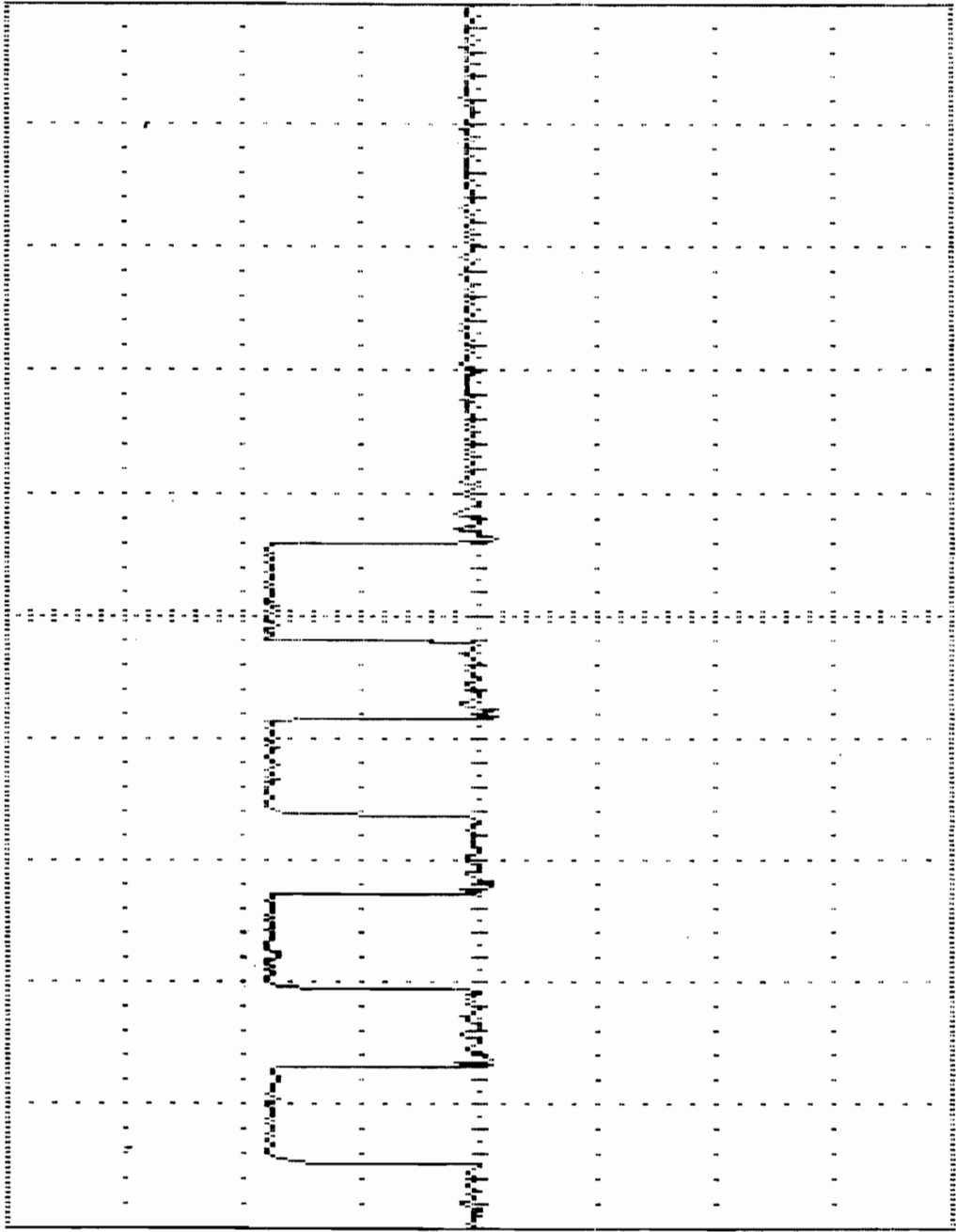
TIMEBASE: μ sec
2.0 us/div

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

5. U2-22 OE (topical output enables activity)

Y1: 10mV (normal)
2.0 V/div
Offset: +0.00 V

Y2: DC (normal)
2.0 V/div
Offset: -4.96 V

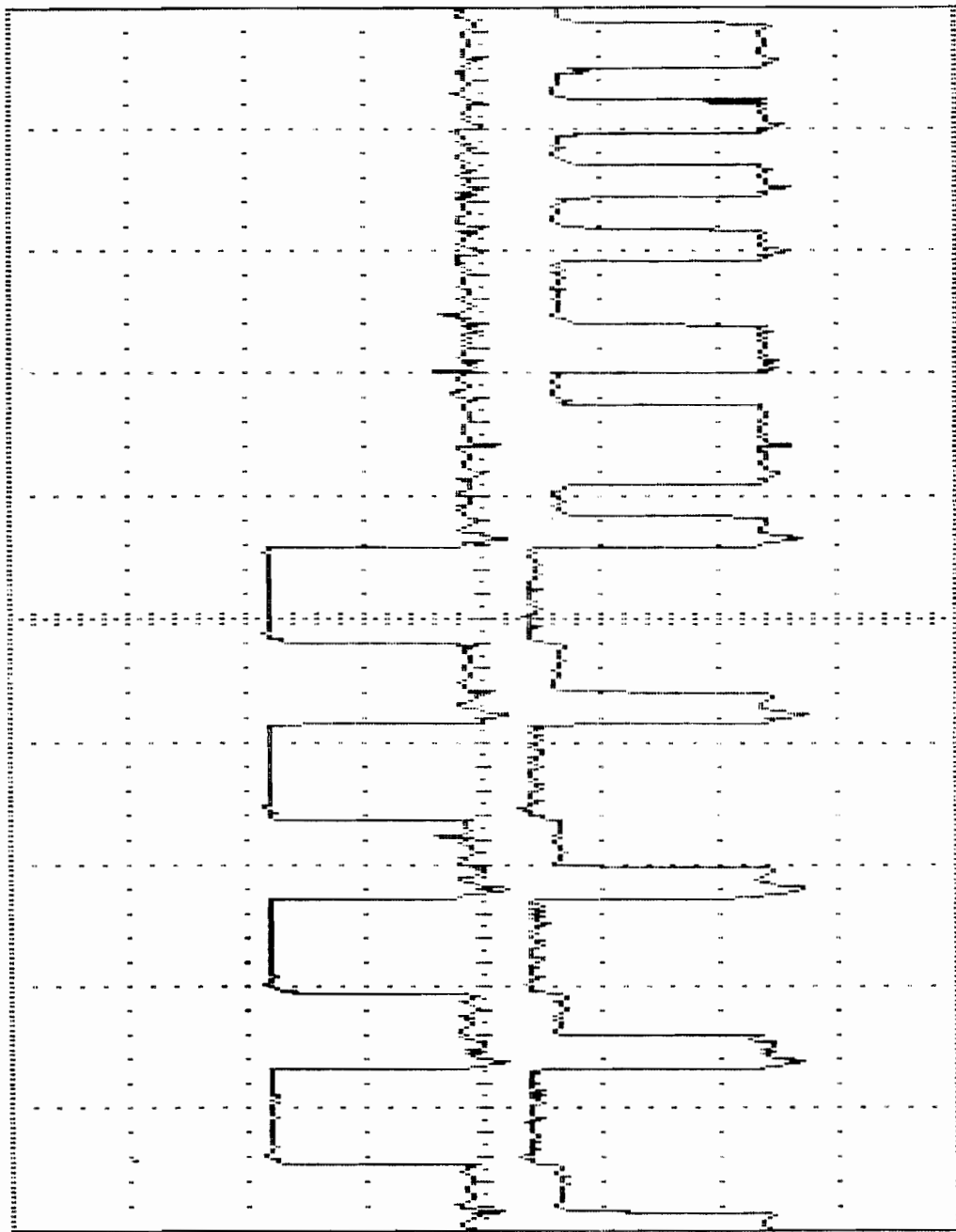
TIMEBASE: $2.0\ \mu\text{s}$ /div

TRIG SOURCE: Y1

TRIG SLOPE: (+)

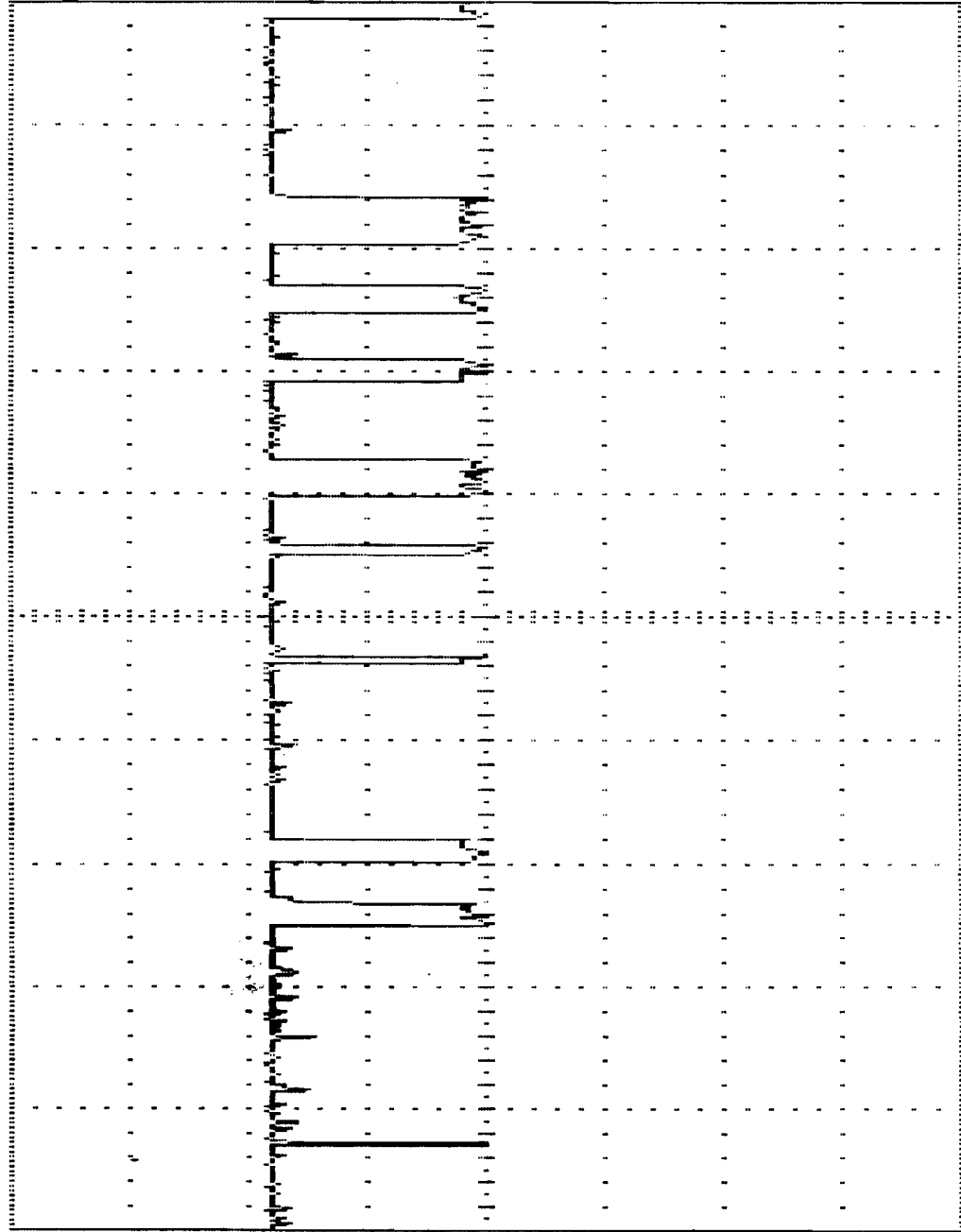
TRIG LEVEL:
+0.08 V

TRIG MODE: single
triggered



F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR M/C2 F4: DEFINE CURSORS F8: NEXT MENU

6. 02-20/02-22 (ce/oe timing)



Y1: NORM (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 M/V/C2 F4: DEFINE CURSORS F8: NEXT MENU

7. U1-21 RD (used to test the trigger)

Y1: IN (normal)
2.0 V/div
Offset: +0.00 V

Y2: OFF (normal)
2.0 V/div
Offset: -4.96 V

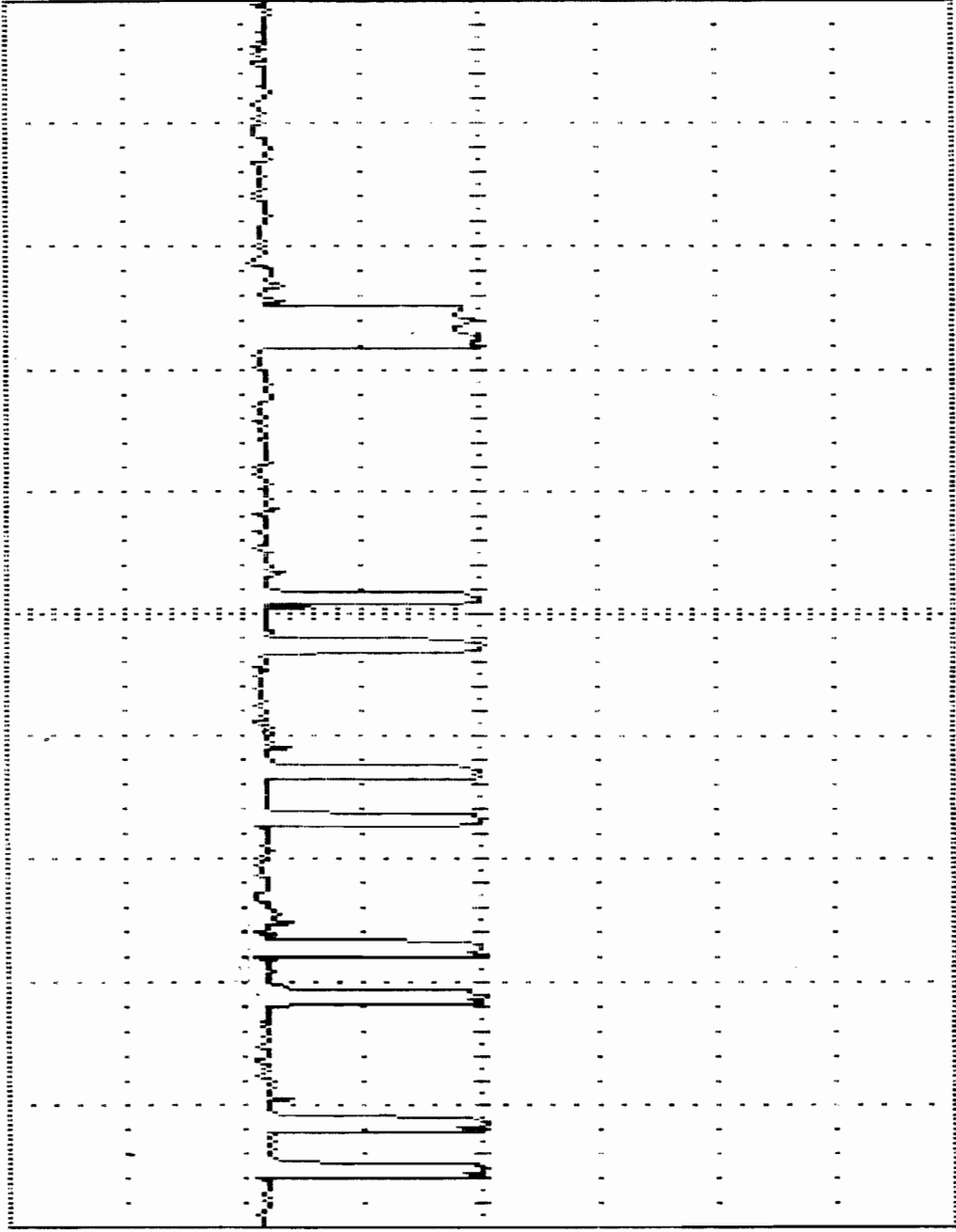
TIMEBASE: $\frac{1}{10}$
2.0 μ S/div

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

8. 01-22 WTR (typical write level activity)

Y1: 1V (normal)
2.0 V/div
Offset: +0.00 V

Y2: OFF (normal)
2.0 V/div
Offset: -4.96 V

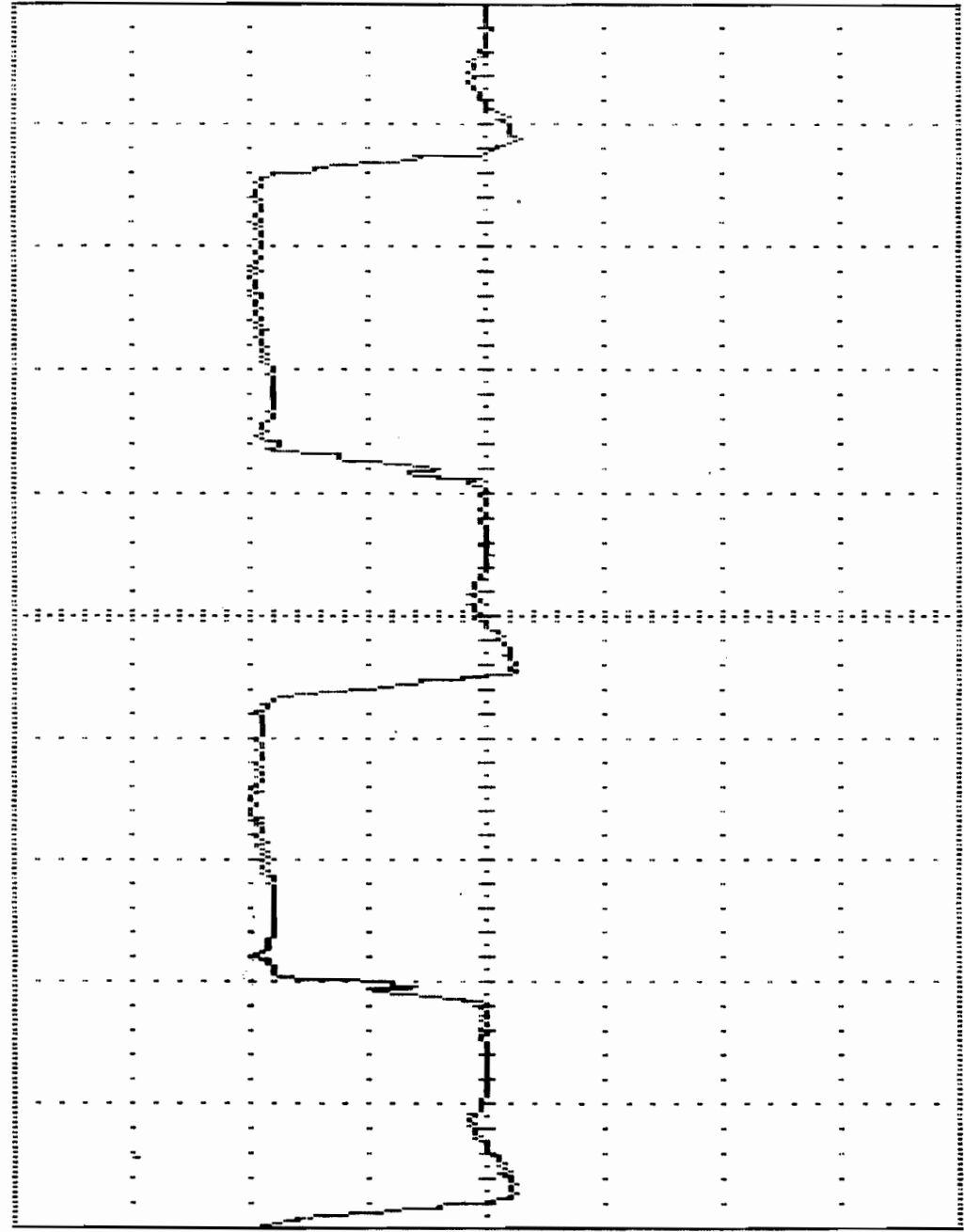
TIMEBASE: 100 ns/div

TRIG SOURCE: Y1

TRIG SLOPE: (+)

TRIG LEVEL: +0.00 V

TRIG MODE: single triggered



F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR M/C2 F4: DEFINE CURSORS F8: NEXT MENU

Y1: **IN**(normal)
2.0 V/div
Offset: +0.00 V

Y2: **OFF**(normal)
2.0 V/div
Offset: -4.96 V

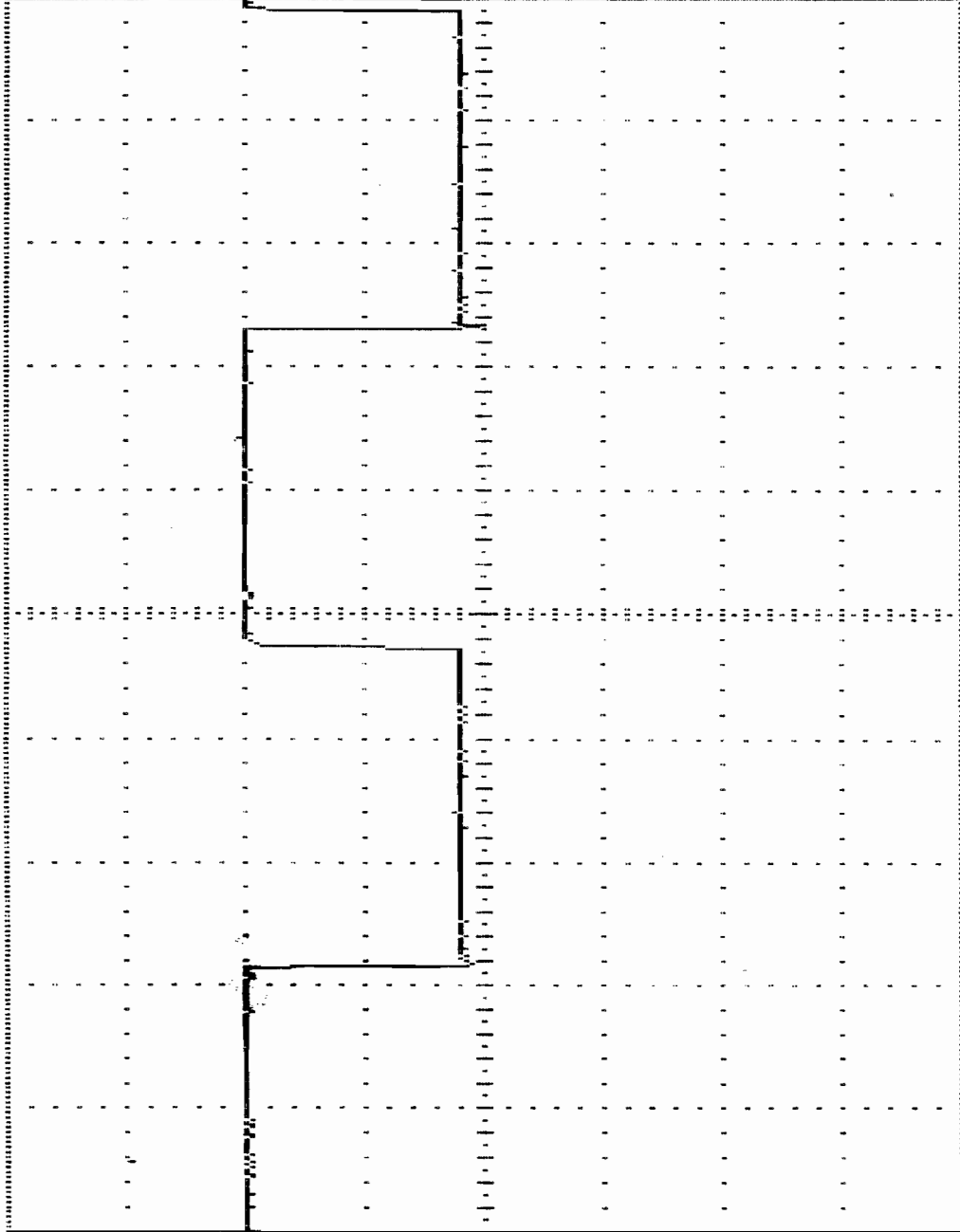
TIMEBASE: μ sec
5.0 us/div

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

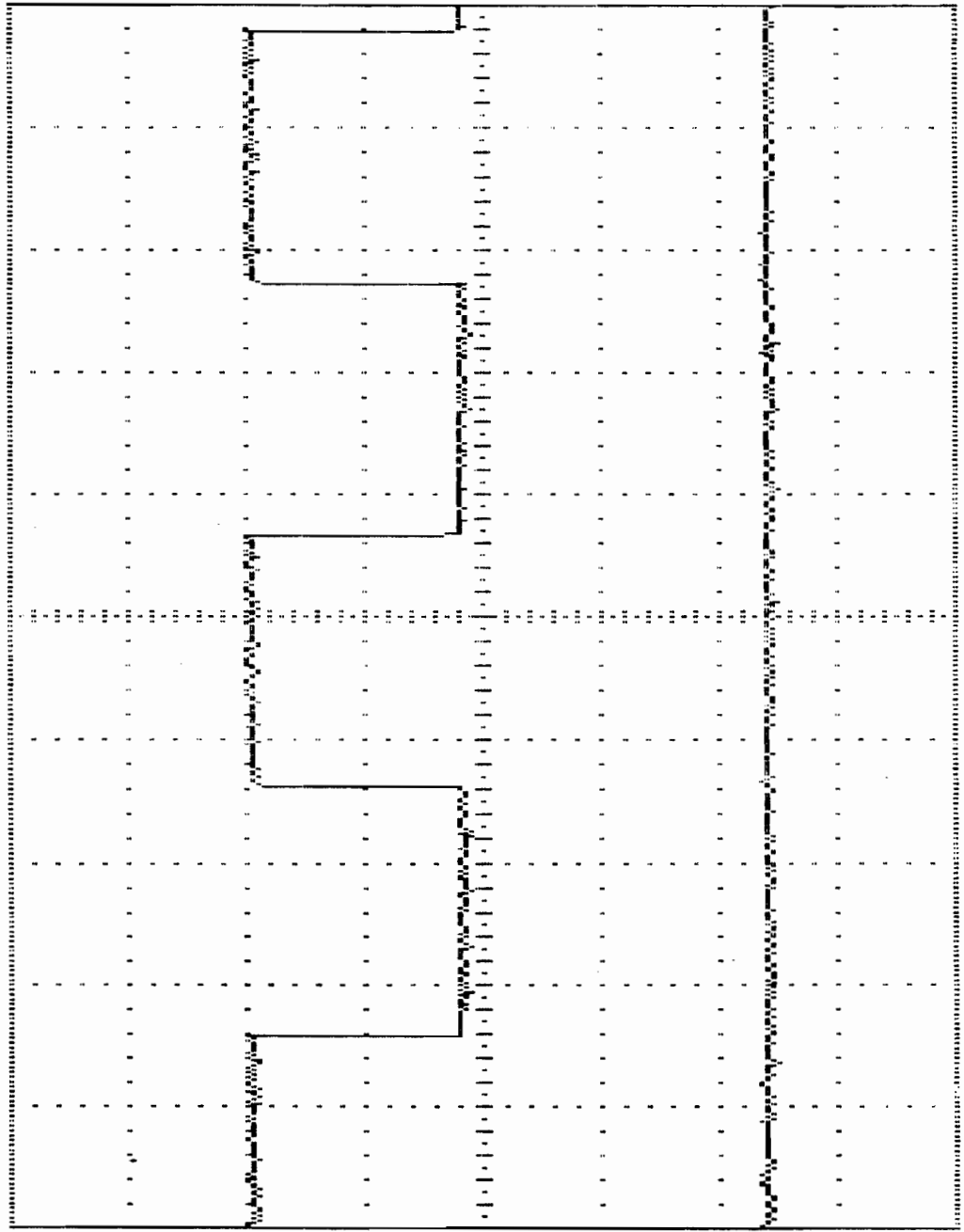
TIMEBASE: μ s/div
200 us/div

TRIG SOURCE: Y1

TRIG SLOPE: (+)

TRIG LEVEL:
+0.08 V

TRIG MODE: single
triggered



F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR MM/C2 F4: DEFINE CURSORS F8: NEXT MENU

Y1: 100 mV/div
100 mV/div
Offset: +0.0mV

Y2: OFF(normal)
2.0 V/div
Offset: -4.96 V

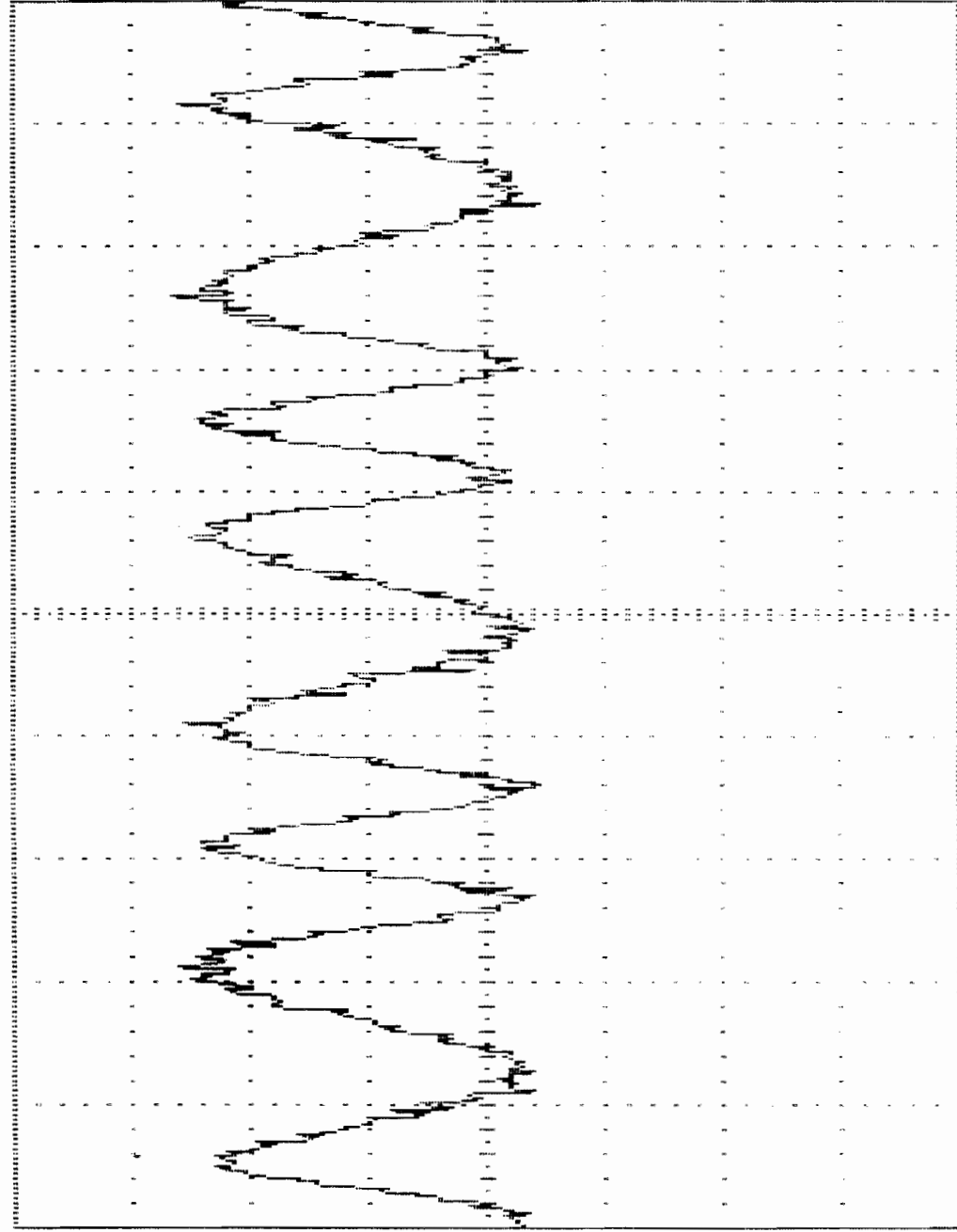
TIMEBASE: 500 us/div

TRIG SOURCE: Y1

TRIG SLOPE: (+)


TRIG LEVEL: +4.0mV

TRIG MODE: single triggered




F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR MW/C2 F4: DEFINE CURSORS F8: NEXT MENU

12. U4-2 TX AUDIO (CAL MODE "D" function 1210/2210 Hz alternating)

Y1:  (normal)
2.0 V/div
Offset: +0.00 V

Y2: OFF (normal)
2.0 V/div
Offset: -4.96 V

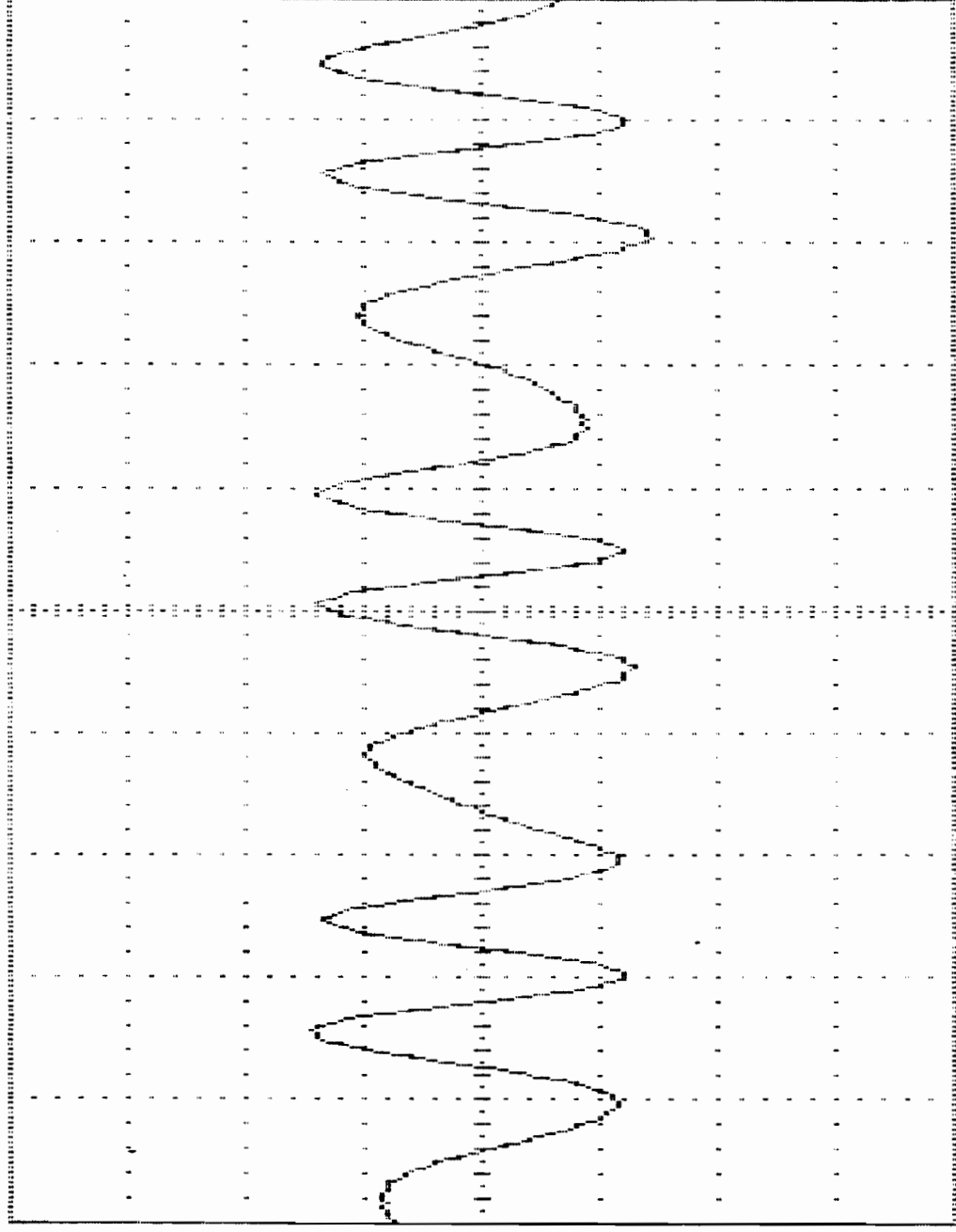
TIMEBASE: 
500 us/div

TRIG SOURCE: Y1

TRIG SLOPE: (+)

TRIG LEVEL:
+0.08 V

TRIG MODE: single
triggered



F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR  C2 F4: DEFINE CURSORS F8: NEXT MENU

13. U26-14 RX HUD10 at bandpass, filter, test

Y1: AC (normal)
500 mV/div
Offset: +0.00 V

Y2: OFF (normal)
2.0 V/div
Offset: -4.08 V

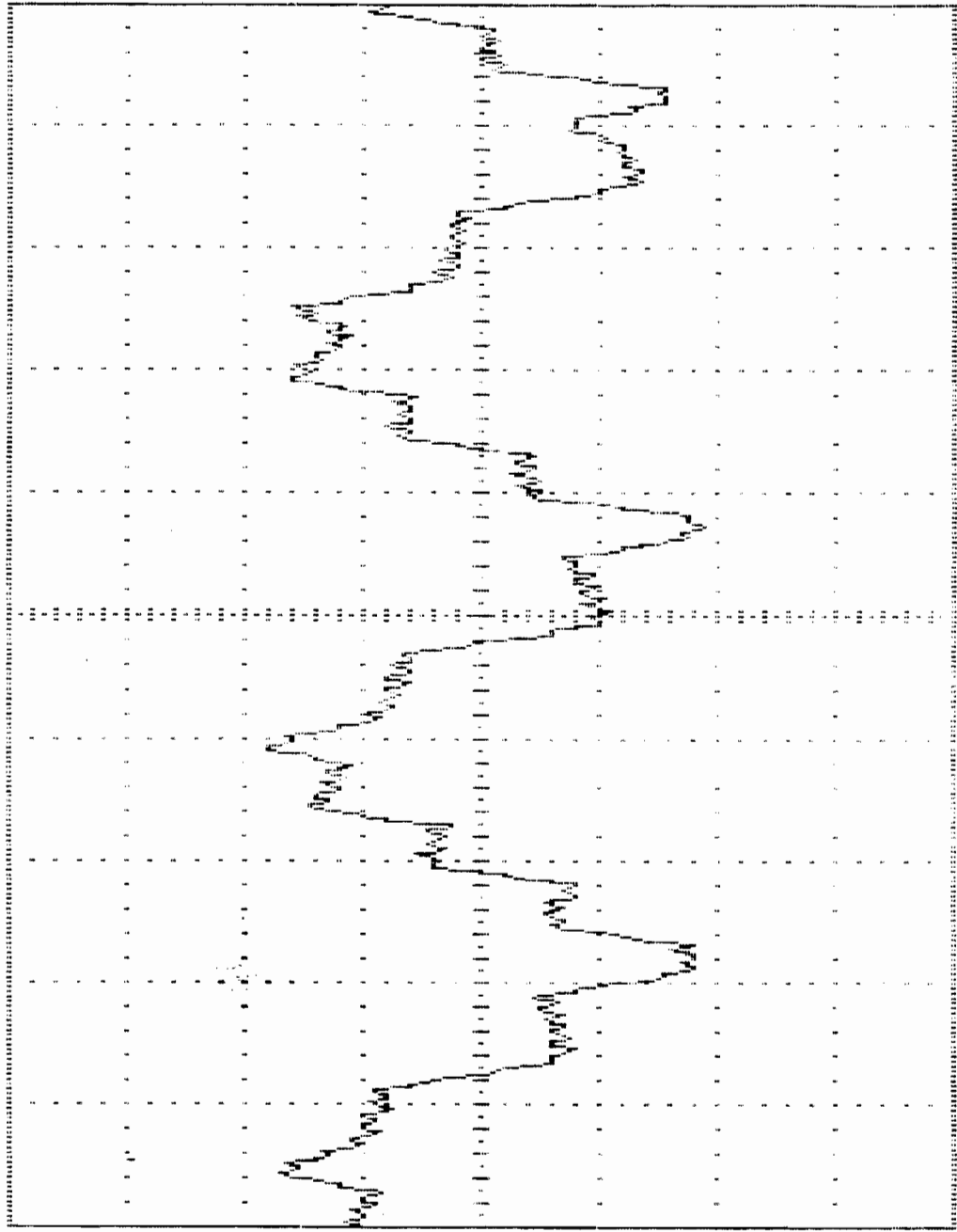
TIMEBASE: μ s/div
500 μ s/div

TRIG SOURCE: Y1

TRIG SLOPE: (+)

TRIG LEVEL:
+0.02 V

TRIG MODE: single
triggered



F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR CH/C2 F4: DEFINE CURSORS F8: NEXT MENU

14. U32-7 Summer output showing plot and long time constants, see 11:11:11

Y1: **Auto**(normal)
500 mV /div
Offset: +0.00 V

Y2: OFF(normal)
2.0 V /div
Offset: -4.08 V

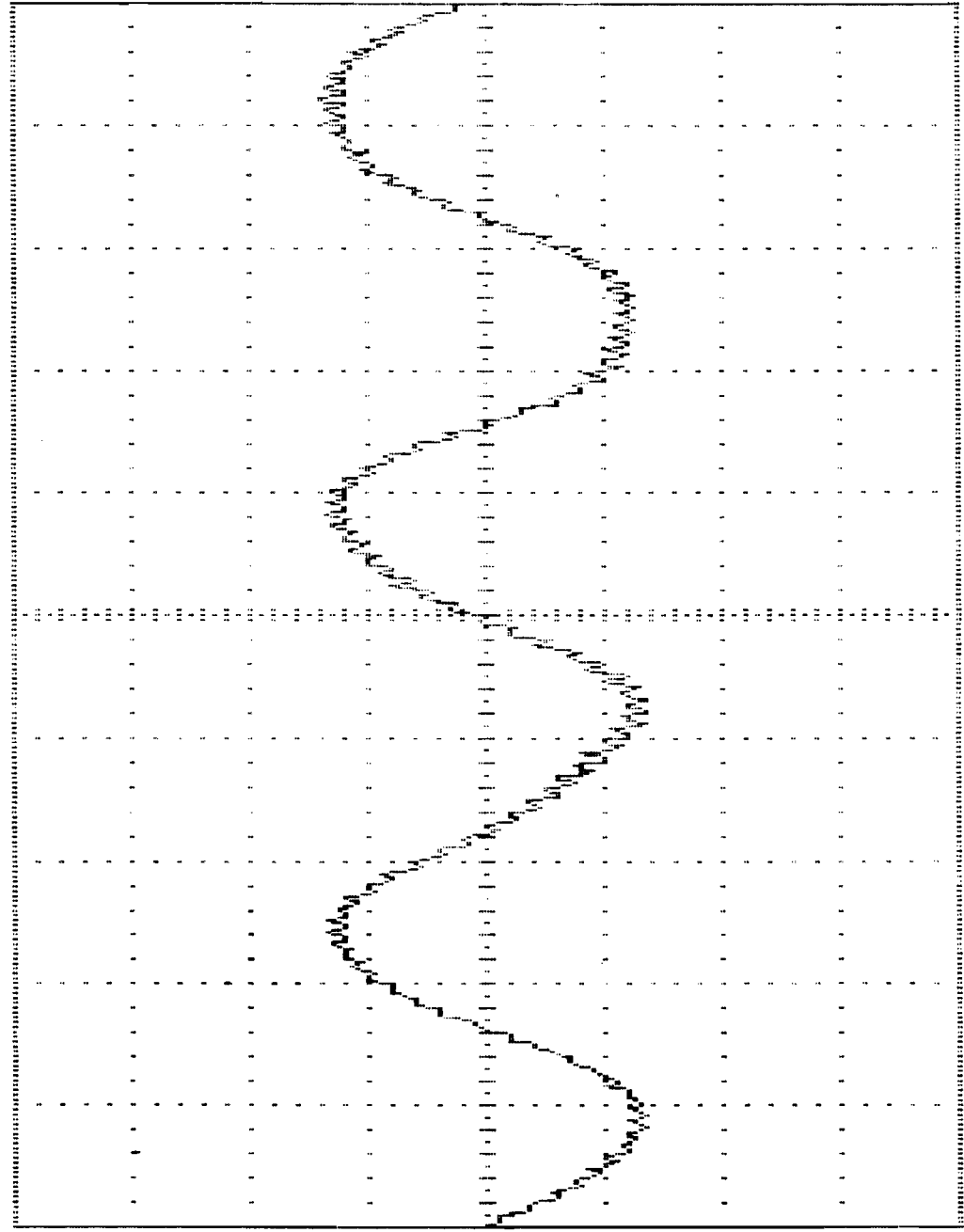
TIMEBASE: **Auto**
500 us /div

TRIG SOURCE: Y1

TRIG SLOPE: (+)

TRIG LEVEL:
+0.02 V

TRIG MODE: single
triggered



F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR ON/C2 F4: DEFINE CURSORS F8: NEXT MENU

15. U34-8 LOWPASS filter output.

Y1: ~~ON~~(normal)
2.0 V/div
Offset: +0.00 V

Y2: OFF(normal)
2.0 V/div
Offset: -4.08 V

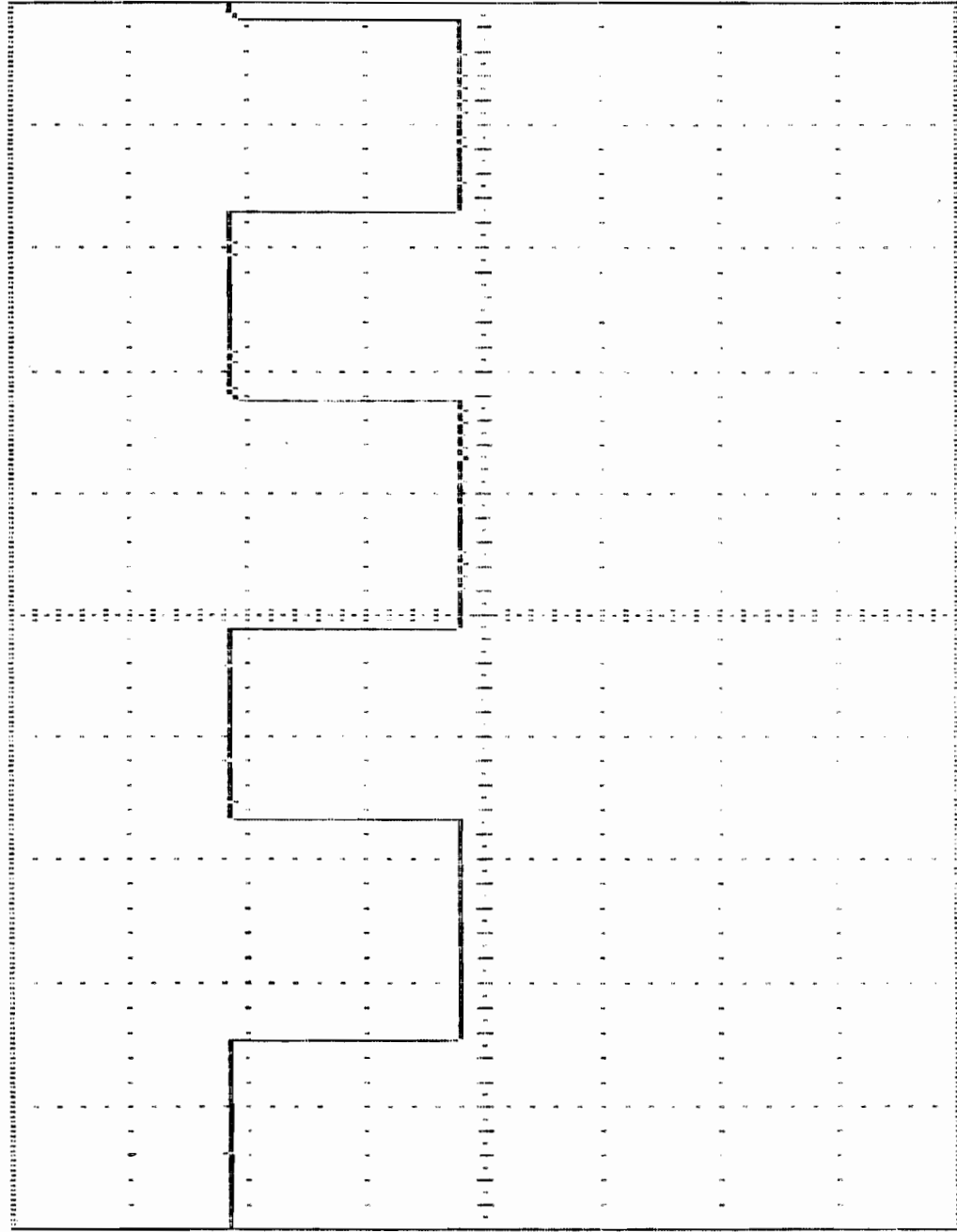
TIMEBASE: μ sec
500 μ s/div

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

16. U15-6 RXDATA inverter/shaper output

Y1: 200 mV/div
Offset: +0.000 V

Y2: 2.0 V/div
Offset: -4.00 V

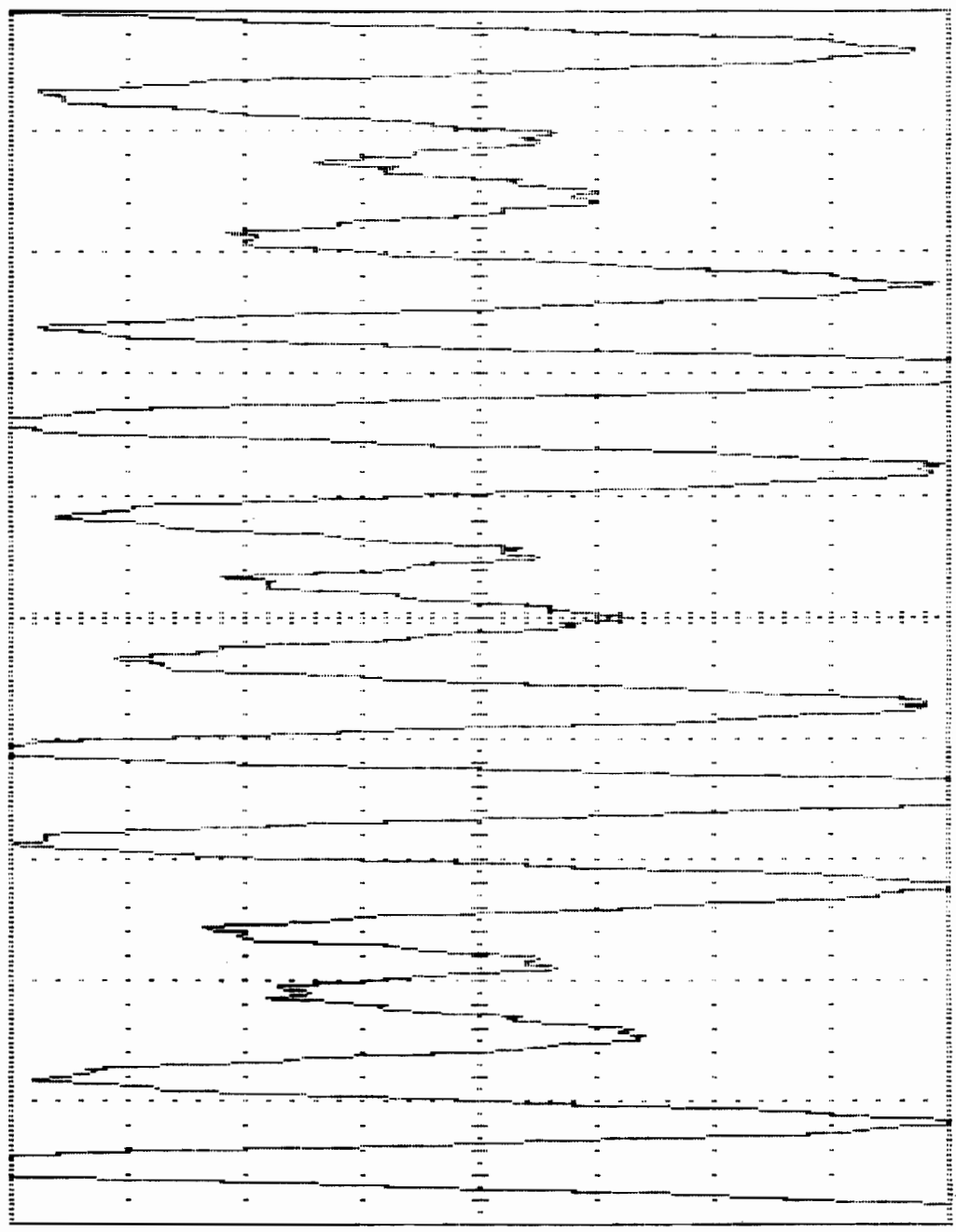
TIMEBASE: 500 uS/div

TRIG SOURCE: Y1

TRIG SLOPE: (+)

TRIG LEVEL: +0.008 V

TRIG MODE: single triggered



F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR M/C2 F4: DEFINE CURSORS F8: NEXT MENU

17. U28-7 INPUT RC and HSW 200-1.7 action on audio signal input.

Y1: (normal)
1.0 V /div
Offset: +0.00 V

Y2: OFF (normal)
2.0 V /div
Offset: -4.08 V

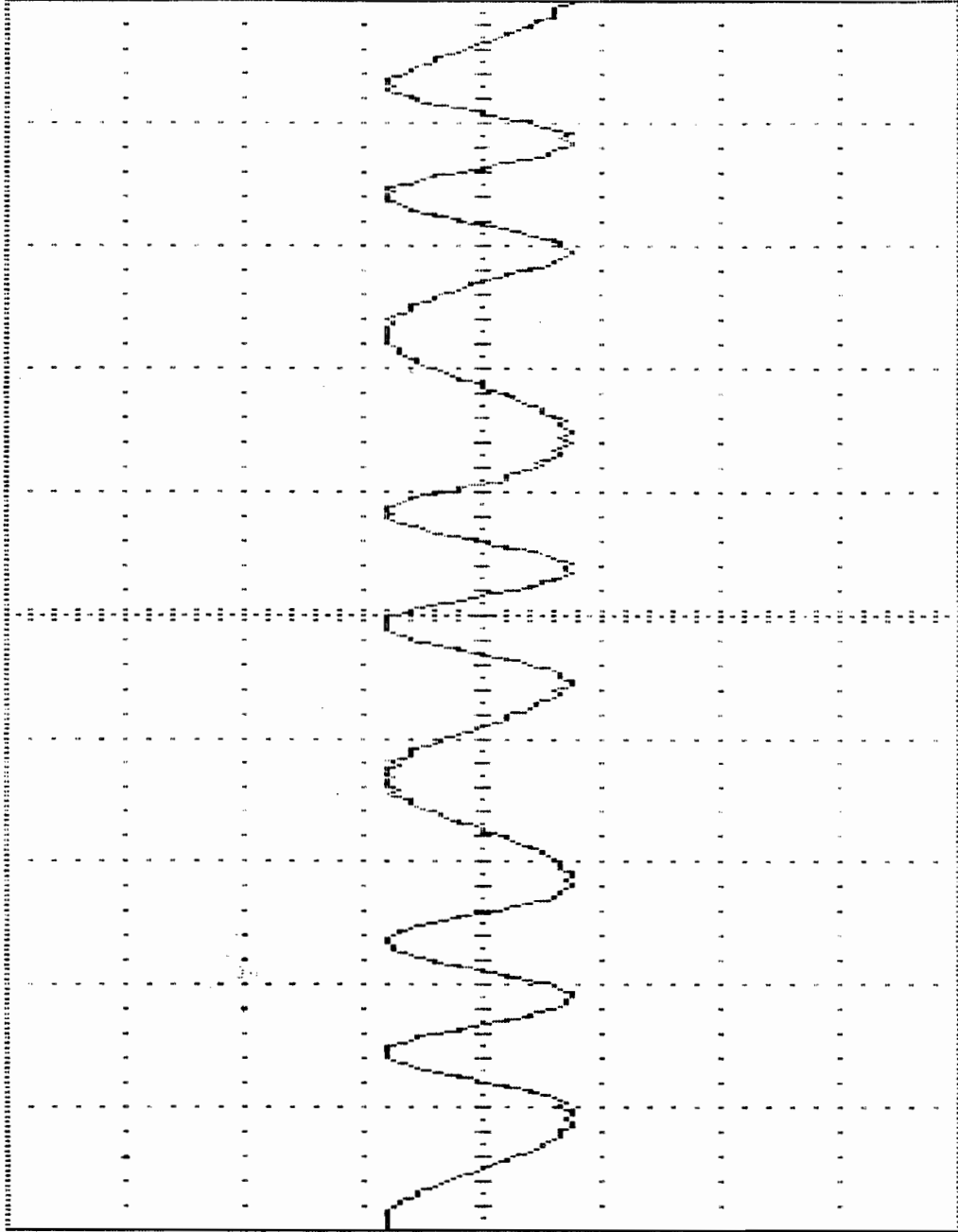
TIMEBASE: (normal)
500 μ S /div

TRIG SOURCE: Y1

TRIG SLOPE: (+)

TRIG LEVEL:
+0.04 V

TRIG MODE: single
triggered



F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR M/C2 F4: DEFINE CURSORS F8: NEXT MENU

18. U40-2 TX audio output from generator

Y1: 200 mV/div
Offset: +0.000 V

Y2: 2.0 V/div
Offset: -4.08 V

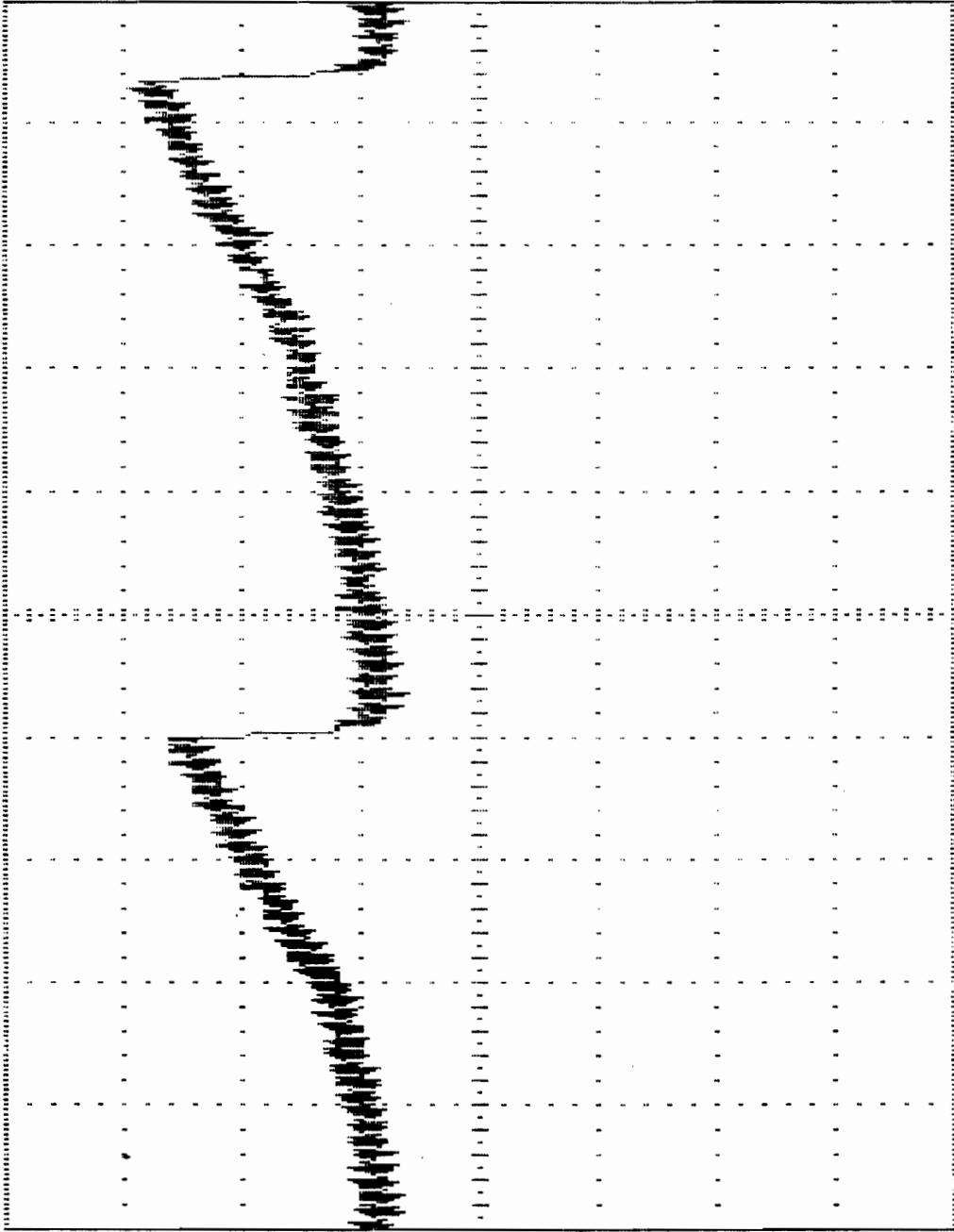
TIMEBASE: 1.0 MS/div

TRIG SOURCE: Y1

TRIG SLOPE: (+)

TRIG LEVEL: +0.008 V

TRIG MODE: single triggered



F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR M/V/C2 F4: DEFINE CURSORS F8: NEXT MENU

19. C55/Q10 WATCHDOG TIMER CHARGE/DISCHARGE

Y1: ~~IN~~(normal)
2.0 V/div
Offset: +0.00 V

Y2: OFF(normal)
2.0 V/div
Offset: -4.08 V

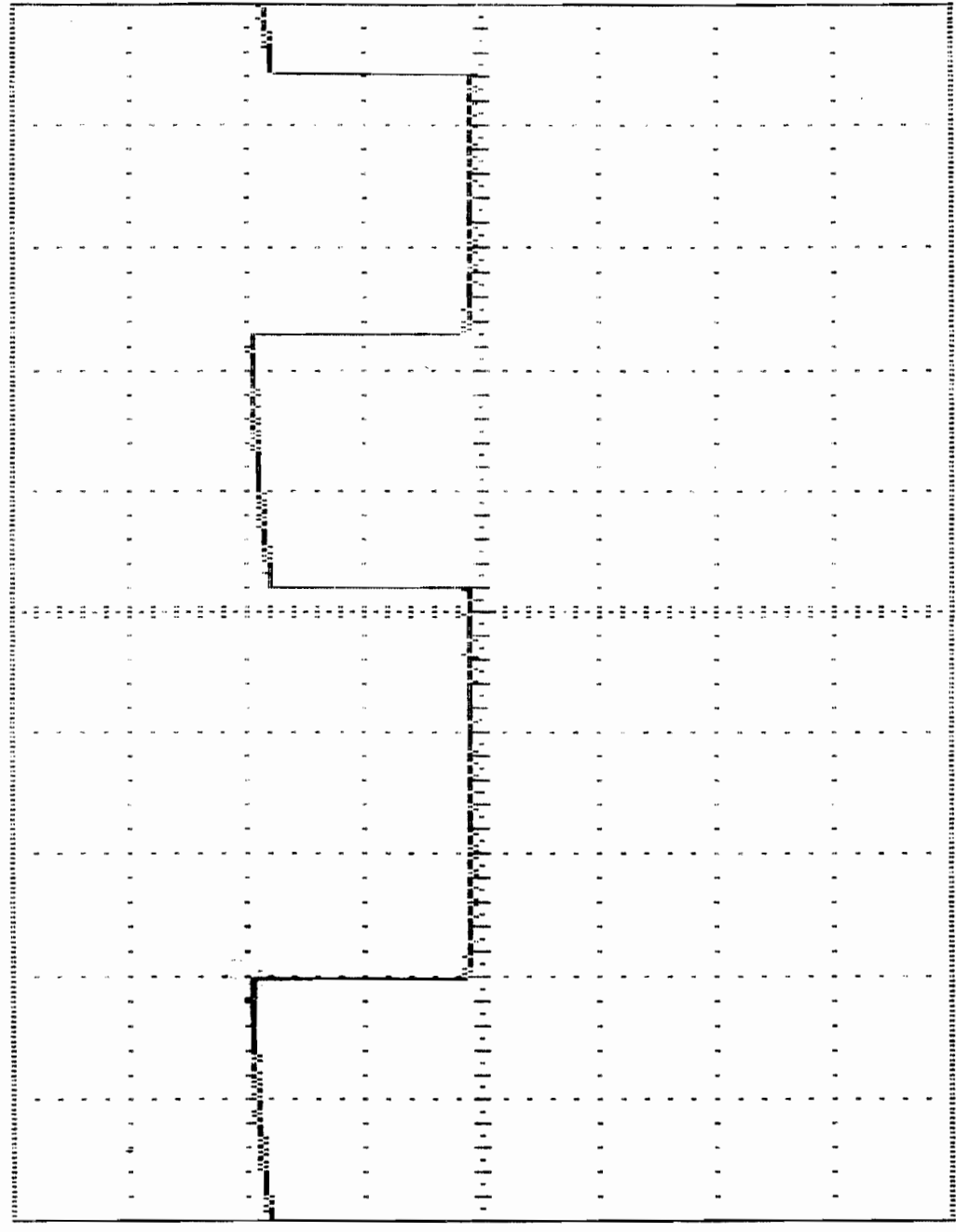
TIMEBASE: $\frac{1}{10}$
1.0 MS/div

TRIG SOURCE: Y1

TRIG SLOPE: (+)

TRIG LEVEL:
+0.08 V

TRIG MODE: single
triggered



F1: ZERO CAL F2: RESET/MANUAL F3: CURSOR M/MC2 F4: DEFINE CURSORS F8: NEXT MENU

20. 07-16 DTRA (PTTW WATCHDOG TIMER LOGIC SIGNAL)