



NCR DECISION MATE V

**System Technical
Manual
- Hardware**

CP/M-80 and CP/M-86 are registered trademarks of Digital Research Inc.

MS-DOS is a trademark of Microsoft Inc.

Document Order Numbers

Augsburg Stock No.: 017-0032261B MIRS Library, Dayton: ST-2104-35

Copyright ©1983 by NCR Corporation
Dayton, Ohio
All Rights Reserved
Printed in the Federal Republic of Germany

Second Edition, July 1983

It is the policy of NCR Corporation to improve products as new technology, components, software, and firmware become available. NCR Corporation, therefore, reserves the right to change specifications without prior notice.

All features, functions, and operations described herein may not be marketed by NCR in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult your dealer or NCR office for information that is applicable and current.

FOREWORD

The NCR DECISION MATE V System Technical Manuals are designed to provide both hardware and software information: they are intended for designers, system integrators, programmers, and other interested persons who require detailed information on the construction and operation of the NCR DECISION MATE V.

Problems arising from any changes that you make to the hardware or software of the NCR DECISION MATE V are your responsibility. NCR cannot assist in resolving problems that may arise when making changes to the hardware or software.

The first manual provides general information on the NCR DECISION MATE V and its various options. Information is included on how to identify the various models and kits that are available. The hardware description includes information about the I/O bus, signal levels, power requirements, and plug/pin assignments.

The other manuals provide information on the various operating system software used with the NCR DECISION MATE V. The software descriptions include information for using system routines at machine code level.

The appendices provide schematics, component locations, software listings, and other information that may be helpful to the user of these manuals.

NCR DECISION MATE V SYSTEM TECHNICAL MANUALS

**System Technical Manual
Hardware**

**System Technical Manual
CP/M[®]-80**

**System Technical Manual
MS[™]-DOS**

**System Technical Manual
CP/M[®]-86**

In the NCR DECISION MATE V System Technical Manual series, the chapters are arranged in numeric sequence and the appendices in alphabetic sequence:

Hardware — Chapters 1 and 2, Appendix A

CP/M-80 — Chapter 3, Appendix B

MS-DOS — Chapter 4, Appendix C

CP/M-86 — Chapter 5, Appendix D

NCR DECISION MATE V SYSTEM TECHNICAL MANUAL HARDWARE

CONTENTS

SYSTEM OVERVIEW

General Description	1-1
NCR Decision Net	1-3
Kit Descriptions	1-4
Model and Kit Identification	1-6
Environmental Considerations	1-8
Safety	1-9
NCR 3282 Winchester Disk Drives	1-9

HARDWARE DESCRIPTION

Controller Module	2-1
Controller Board	2-1
Graphics Controller Board	2-18
16-Bit Processor	2-19
Power Supply	2-21
Flexible Disk Drives	2-24
Fixed (Winchester) Disk	2-27
Keyboard	2-31
Keyboard Codes	2-38
Monochrome CRT	2-55
Color CRT	2-57
RS-232C Peripheral Adapters	2-57
Software Strapping	2-62
Hardware Strapping	2-62
Centronics Peripheral Adapter	2-62
Memory Expansion Modules	2-66
Blank Interface Adapter	2-67
Hints on Printer Connection	2-68
Centronics Peripheral Adapter (K210)	2-68
RS-232C Peripheral Adapters (K212, K213)	2-68
Printers not Supporting XOn — XOff	2-69
Printer Fails to Print	2-70
Incorrect Printing	2-70

APPENDIX A

Initialization Firmware (Controller Board)	A-1
Keyboard Firmware (Version 1)	A-67
Keyboard Firmware (Version 2)	A-99
Controller Board	A-133
Controller Board (Multi-Layer)	A-143
Graphics Controller (Monochrome)	A-153
16-Bit Processor	A-159
Power Supply	A-163
AC/DC Wiring	A-165
RS-232C Adapter	A-167
Centronics Adapter	A-169
Fixed Disk Interface	A-171
Keyboard	A-173
Monochrome CRT	A-175
Diagnostic Module	A-177
Memory Extension 64K	A-181
Memory Extension 192K	A-183
Memory Extension 448K	A-185
Matrix Character Set	A-187
Character Generator Table	A-191
Keyboard Output Assignments	A-193
Programmable Communications Interface	A-195
16-Bit Firmware	A-201

SYSTEM OVERVIEW

GENERAL DESCRIPTION

The NCR DECISION MATE V is the first of a new generation of NCR business/professional personal computer systems. It is designed to act as the control center for systems that fulfill a wide range of user requirements. Industry standard interfaces are used to provide the user with a wide selection of peripherals. This system approach provides a powerful tool to use in business applications, scientific calculations, and in various technological fields.

The processor and all supporting logic, CRT screen, disk drives, power supply, and I/O bus are all integrated into a single cabinet: this provides a system that requires the minimum of space and connecting cables. The separate keyboard is connected by a coiled cable to the processor unit.

Most of the electronics are contained on the controller board, where possible LSI components are used providing a high degree of reliability. Included on this board are the Z80 microprocessor, DMA, flexible disk controller, keyboard controller, and 64 K bytes RAM. Also, the I/O bus, complete with seven user accessible sockets, is integrated into the controller board. Two additional sockets are available for the factory installation of some features. In this configuration the NCR DECISION MATE V uses CP/M[®]-80 operating system software and any compatible 8-bit application packages. A 16-bit board, with an Intel 8088 microprocessor, can be connected to the controller board making the computer into a dual 8/16-bit system. In this configuration the system can still use CP/M-80 and 8-bit application software, or it can use either CP/M-80 or MS-DOS[™] operating system software and any compatible 16-bit application packages.

The system may have either a monochrome CRT with a green display on a dark background, or a color CRT capable of displaying eight different colors. The CRTs can display 25 lines of data, with up to 80 characters per line, or in the graphics mode, various figures, charts, and plans.

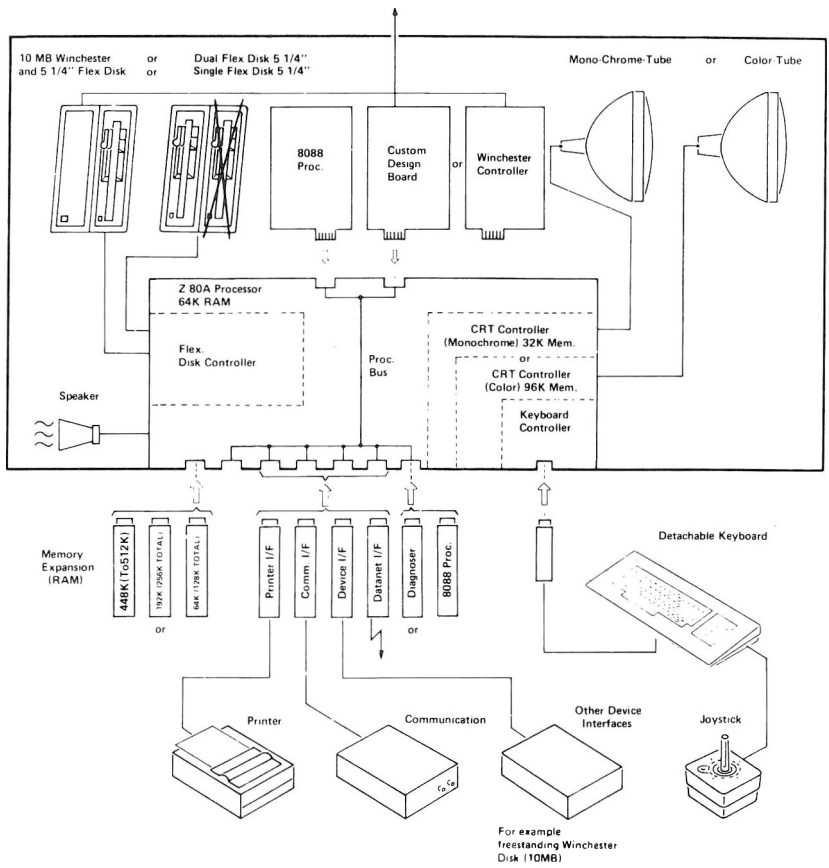


Figure 1.1 System overview

The integrated disk drives are available in the following combinations:

- One 5 1/4-inch flexible disk drive
- Two 5 1/4-inch flexible disk drives
- One 5 1/4-inch flexible disk drive and one 5 1/4-inch fixed Winchester disk drive

The I/O bus is an integral part of the controller board; it includes seven sockets that allow the user to connect the various peripheral adapters and options to the bus. These adapters and options all use a common style of housing, permitting flexibility in configuring the hardware. The cabinet design is such that the adapters/options are fully recessed into the rear of the cabinet when installed. The adapters/options do not require any preparation and can be plugged directly into the bus by the user. The peripheral adapters are used to connect industry standard printers, modems, and terminals to the bus. The bus is also used for the connection of the memory expansion modules and other special purpose modules.

Different versions of a "switcher" type power supply are used depending on the power requirements of the different models of the NCR DECISION MATE V. The power supply also provides logical voltages to the bus for use by the peripheral adapters and options.

NCR DECISION NET

The NCR DECISION NET System is a local area network system, typically consisting of the following components:

- An NCR MODUS File Sharer
- An NCR OMNINET local area network
- A number of NCR DECISION MATE V personal computers (up to 63)

To connect an NCR DECISION MATE V into the NCR DECISION NET System, a special adapter, known as a transporter, is required. This transporter (3273-K600) connects directly into an I/O bus slot at the rear of the NCR DECISION MATE V.

KIT DESCRIPTIONS

Various kits are available to enhance the capabilities or change the hardware configuration of the NCR DECISION MATE V. Many of these kits can be plugged directly into the I/O bus by the user: others require installation by a service engineer.

Integrated Fixed Disk

This kit (K012) provides for the integration of a fixed (Winchester) disk drive. The kit includes the necessary changes for the power supply. When installed, the system has one fixed disk drive (capacity 10M bytes) and one flexible disk drive (capacity 320/360K bytes).

Second Integrated Flexible Disk

These kits upgrade an NCR DECISION MATE V from a system with one flexible disk drive to a system with two flexible disk drives. Kit K018 is for systems produced in the DIN color scheme, kit K019 is for systems in the Corporate Appearance Plan (CAP) color scheme.

Memory Expansion Modules

These kits (K200, 202, and 208) can be installed, by the user, directly into slot 1 of the I/O bus. It is only possible to install one kit into a system. The kits provide the following memory capacities:

- K200 increases memory capacity from 64 K to 128 K bytes
- K202 increases memory capacity from 64 K to 256 K bytes
- K208 increases memory capacity from 64 K to 512 K bytes

Centronics Peripheral Adapter

This kit (K210) can be installed, by the user, directly into the I/O bus (slots 2 to 6). The kit includes a 2 metre (6.6 ft.) cable complete with plug, that allows connection to a Centronics compatible (parallel input) printer.

RS-232-C (Communications) Peripheral Adapter

This kit (K211) can be installed, by the user, directly into the I/O bus (slots 2 to 6). The kit includes a 2 metre (6.6 ft.) cable complete with plug, that allows connection to an RS-232-C compatible (serial) communications device.

RS-232-C (Printer) Peripheral Adapter

This kit (K212) is similar to K211, but is used for the connection of RS-232-C compatible (serial) printers.

RS-232-C (Plotter) Peripheral Adapter

This kit (K213) is used for the connection of a compatible plotter (e.g. NCR 5403) to the NCR DECISION MATE V.

Blank Interface Adapters

These kits (K214, 217, and 218) comprise a housing, an I/O bus connector, and a blank printed circuit board. They provide the necessary hardware to help the system integrator to develop custom interfaces or adapters.

- K214 provides hardware for one adapter
- K217 provides hardware for five adapters
- K218 provides hardware for 25 adapters

Integrated Modem

This kit (K219) can be installed, by the user, directly into the I/O bus (slots 2 to 6). It is designed to support the following modems:

- Bell 103 (300 baud)
- Bell 2035 (1200 baud)

Built into the modem housing are two sockets for the connection of standard telephone type plugs: a cable is included for the connection of a telephone.

Diagnostic Module

This module (K220) can be installed directly into slot 7 of the I/O bus, and is intended for service engineers and other persons interested in diagnosing problems. The module tests the computer and provides information on failures. It also monitors the logical voltages (+5 and +12 Vdc) and indicates whether the voltages are within or outside the specified limits. For a full description, refer to the NCR DECISION MATE V Service Manual.

8/16-Bit Processor

Kit K231 can be installed, by the user, directly into slot 7 of the I/O bus. It converts the NCR DECISION MATE V from an 8-bit system to a dual 8/16-bit system that is capable of using either 8-bit or 16-bit operating system and application software.

Kit 230 functions similarly to K231, but is connected internally to the controller board. This kit should be installed by a field engineer; it is not suitable for installation by the user.

Tilt Mechanism

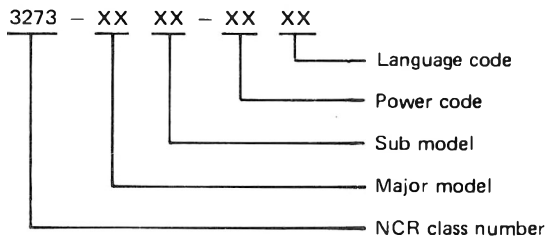
This kit (K240) provides a means of tilting the NCR DECISION MATE V by seven degrees. The user can easily install the device by fixing a base plate to the bottom of the computer, using the existing holes in the base of the cabinet.

NCR OMNINET Transporter

This kit (K600) can be installed, by the user, directly into the I/O bus (slots 2 to 6). The adapter has twice the thickness of other adapters, therefore it requires two slot positions for installation. The adapter provides the interface between the NCR DECISION MATE V and the DECISION NET System.

MODEL AND KIT IDENTIFICATION

The configuration of the hardware is defined by the model number. The standard arrangement of the model number is shown below:



The model number of the NCR DECISION MATE V is shown in Figure 1.2. The optional features are shown in Figure 1.3. An asterisk indicates a planned development not released at the time of printing, but included for reference.

Class	3273	NCR Class Number
Major Model	01 11 12	8-bit Proc./12" CRT Monochrome 8/16-bit Proc./12" CRT Monochrome 8/16-bit Proc./12" CRT Color
Sub-Model	01 02 03	One 5 1/4" Flex Disk Drive Two 5 1/4" Flex Disk Drives One 5 1/4" Flex Disk Drive and one 5 1/4" Winchester Fixed Disk
Power Code	60 65 70 74 75	120 V, 50/60 Hz 220 V, 50/60 Hz 100 V, 50/60 Hz 230 V, 50/60 Hz 240 V, 50/60 Hz
Language Code	00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15* 16* 17* 18* 19* 20 25 42 33 40	US English International English UK English Spanish Portuguese French German Dutch Flemish Swedish Danish Norwegian Italian Greek Yugoslavian Japanese/Katakana Taiwanese Arabic Iranian Hebrew Finnish Latin American/Spanish Canadian South African Swiss
* Not yet released but shown for reference purposes		

Figure 1.2 Model number description

Kit Number	Description
3273-K012 *	Integrated 10 MB Winchester disk (CAP)
3273-K018	Second flexible disk upgrade (DIN)
3273-K019	Second flexible disk upgrade (CAP)
3273-K090	Integrated 10 MB Winchester disk (DIN)
3273-K200	64 to 128KB memory upgrade
3273-K202	64 to 256KB memory upgrade
3273-K208	64 to 512KB memory upgrade
3273-K210	Centronics parallel I/F & cable
3273-K211	RS-232C serial I/F & modem cable
3273-K212	RS-232C serial I/F & printer cable
3273-K213	RS-232-C serial I/F & plotter cable
3273-K214	Blank interface adapter & bus connector
3273-K217	5 blank interface adapters bus connectors
3273-K218	25 blank interface adapters bus connectors
3273-K219	Integrated modem
3273-K220	Diagnostic module
3273-K230	Dual 8/16-bit processor upgrade (internal)
3273-K231	Dual 8/16-bit processor upgrade (external)
3273-K240	Tilt mechanism
3273-K600	NCR OMNINET transporter
* Not yet released but shown for reference purposes	

Figure 1.3 Kits

ENVIRONMENTAL CONSIDERATIONS

The NCR DECISION MATE V is designed to operate in a normal "office-type" environment, without the need for special heating or air conditioning. The environmental requirements are shown in Figure 1.4; operating the computer at or beyond these limits may cause errors or damage to the hardware.

Operating Temperature	10°C to 35°C (50°F - 95°F) at 500 metres (derated 3.3°C per 1000 metres)
Temperature Change	10°C (50°F) per hour
Relative Humidity	20% to 80%
Transit Temperature	-40°C to 60°C (-40°F to 140°F)
Storage Temperature	10°C to 50°C (50°F to 120°F)

Figure 1.4 Environmental requirements

SAFETY STANDARDS

The NCR DECISION MATE V is designed and built to conform to the following standards:

Product Safety

- USA UL 478 UL listing mark used
- Canada CSA 22.2 - 15.4, CSA monogram used
- Europe IEC 380, Inhouse verification
- Germany VDE 0806, GS label granted by trade association

Radio Protection

- USA FCC Docket No. 20780, Class B
- Germany VDE 0871, Class A certified by German Federal Post (FTZ)

Radiation Emission

- USA Public Law 90 - 602, DHEW Publication No. (FDA) 75 - 8003
- Germany X-Ray Emission Regulations

It is the responsibility of those making changes to the system to ensure that these standards are not contravened.

NCR 3282 WINCHESTER DISK DRIVES

These units are freestanding Winchester disk drives for connection to the NCR DECISION MATE V. They are intended for connection to systems with either one or two integrated flexible disk drives, and cannot be connected to systems with an integrated Winchester disk drive. Each unit has a formatted capacity of 10M bytes, and up to three units may be connected to a system, in a daisy-chain arrangement.

Three models of the NCR 3282 Winchester Disk Drive are available:

- NCR 3282-0101 — this model includes an interface for connection to the NCR DECISION MATE V. It is intended for the user who wishes to connect only one freestanding drive to the system. It is not possible to connect additional drives to this model.
- NCR 3282-0102 — this model includes an interface for connection to the NCR DECISION MATE V, also a connector is included to allow the connection of an NCR 3282-0103 in a daisy-chain arrangement.

- NCR 3282-0103 — this model connects to the NCR 3282-0102 and also has a connector permitting the connection of another NCR 3282-0103.

All models are supplied with a cabinet that includes the drive, power supply, fan, power cables, and the appropriate interconnection cables to provide for easy assembly of the units in a daisy-chain configuration. Technical data for the NCR 3282 Winchester Disk Drive is provided in Figure 1.5.

Capacity (formatted) :	10 Mbytes 612 tracks/unit 17 sectors/track (one used as alternative sector) 512 bytes/sector
Track Density:	345 tpi
Transfer Rate:	5Mbits/sec.
Access Time:	track to track: 3 msec. average: 85 msec.
Motor Start Time:	15 sec.
Encoding Method:	MFM
Sectoring:	Soft
Host Interface:	8 bit bi-directional bus

Figure 1.5 NCR 3282 Technical data

HARDWARE DESCRIPTION

CONTROLLER MODULE

The controller module consists of two printed circuit boards, the controller board and the graphics controller board. Two types of graphic controller boards are used: one, with 32Kbytes of RAM for systems with a monochrome CRT, the other, with 96Kbytes of RAM for systems with color CRT.

CONTROLLER BOARD

The controller is a 300mm x 290mm printed circuit board and utilizes a Z80A microprocessor and auxiliary logic to drive and control a family of 8-bit common memory and peripheral modules. Figure 2.1 shows a block diagram of the controller board.

- Z80A microprocessor is operated with a 4 MHz clock.
- The LSI flexible disk controller contains the circuitry and control functions for interfacing one or two flexible disk drives. It is capable of supporting double density format including double sided recording. Hand shaking signals are provided which make DMA operation easy to incorporate with the aid of the DMA controller integrated circuit.
- The RAM controller provides multiplexed addresses and address strobes, as well as refresh access arbitration for the 64Kbytes dynamic RAM contained on the controller board, and also for the memory expansion modules. (Up to a maximum of 256 Kbytes, or 512Kbytes with the multi-layer controller board).
- The loudspeaker is designed to produce tones for keyboard acceptance click, bell, and music demonstration. A separate circuit provides the frequencies from 110Hz to 1174.7Hz. The small audio speaker is housed on the controller board. The loudness is adjusted by a user-accessible potentiometer.
- The keyboard interface connects the ASCII encoded keyboard to the controller board by a four-wire cable carrying logic ground, frame ground, +5V, and data signal. The keyboard is polled by the controller. When a key is pressed, the serial data is returned on the same line. For a detailed description see the keyboard description.

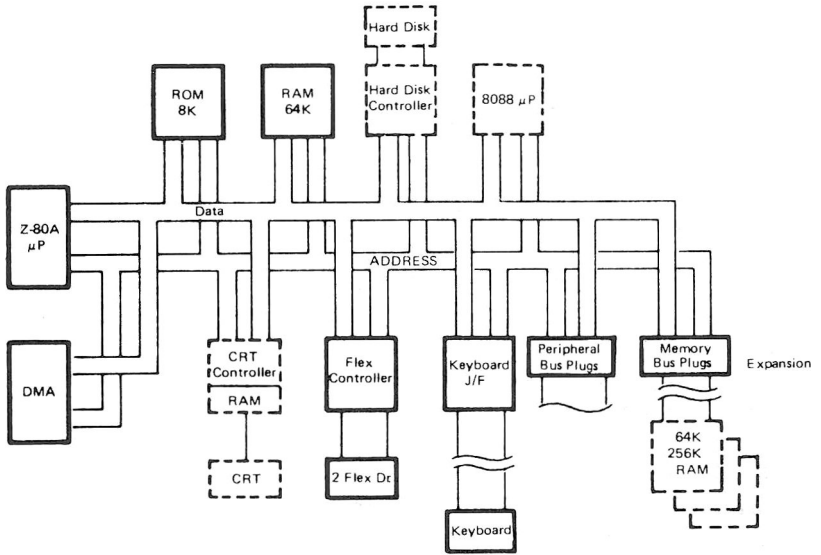


Figure 2.1 Controller block diagram

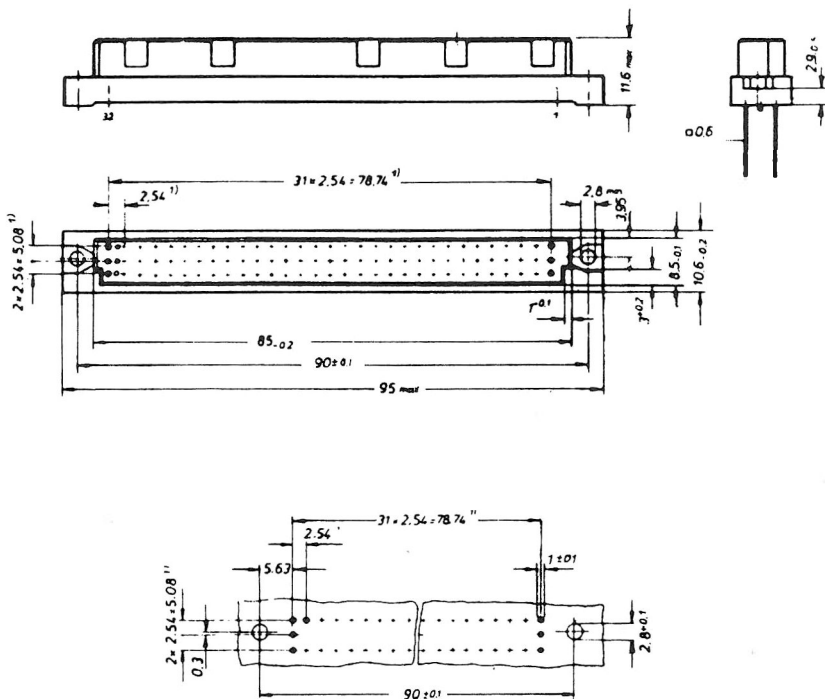
I/O Bus

The lower part of the controller board contains the I/O bus together with seven user-accessible connectors. These connectors are identified on the board as J1 through J7 (reading from left to right), and correspond to the seven slots (numbers 1 through 7) at the rear of the cabinet.

Position J1 is reserved for the connection of any one of the memory expansion modules (K200, K202, K208).

Position J7 is reserved for the connection of either the diagnostic module (K220), or the customer-installable 16-bit processor module (K231).

Positions J2 through J6 are the general purpose slots for the connection of the peripheral adapter and other interfaces to the computer. These five positions are identical, and Figure 2.2 defines the type of connector, while the pin assignments are shown in Figure 2.3.



Controller board bus connector (J2 - J6)

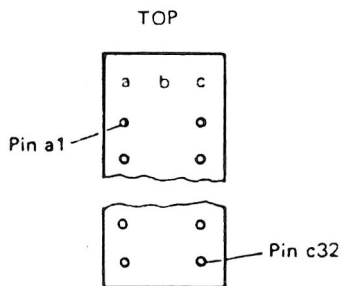


Figure 2.2 Bus connector

a		c
+5 V	1	+5 V
	2	+12 V
RESET/	3	RESET IN/
IOW/	4	IOR/
MEMW/	5	MEMR/
BD1	6	BD0
BD3	7	BD2
BD5	8	BD4
BD7	9	BD6
READY DMA	10	ABTRI/
EOP/	11	
INTACK/	12	IFSEL 4/
AUTO/	13	DIR/
THOLD/	14	HLDA
PCLK/	15	CLK1
LGRD	16	TRAMD/
BA19	17	BA18
BA17	18	BA16
BA15	19	BA14
BA13	20	BA12
BA11	21	BA10
BA9	22	BA8
BA7	23	BA6
BA5	24	BA4
BA3	25	BA2
BA1	26	BA0
IFSEL3/	27	IFSEL2/
IFSEL1/	28	IFSEL0/
DRQ1	29	DRQ0
DACK1/	30	DACK0/
WAIT/	31	INT/
LGRD	32	LGRD

Figure 2.3 Bus (J2-J6) pin assignments

Two additional plug/socket connections are possible on the bus. These are made on the solder side of the controller board and are designated J2A and J7A. These connectors are not considered to be user accessible, rather for factory use, or for use by field engineers and system integrators. Normally, these two connectors are used for:

- J2A — The connection of the fixed disk interface board, or a custom design board.
- J7A — The connection of the 16-bit processor board (factory option and kit K230).

The power requirements for the I/O bus plugs are:

- For each peripheral plug + 5V \pm 5% max. Current 0.5A
or
+12V \pm 5% max. Current 0.1A
- For all plugs (J1, J2, J2A, J3, J4, J5, J6, J7 and J7A)
+ 5V \pm 5% max. 5.0A
+12V \pm 5% max. 0.6A

The DC voltage levels for the I/O bus plugs are:

- Input/Output HIGH-Voltage $V_H = 2.4$ to 5.5 Volts
- Input/Output LOW-Voltage $V_L = -0.5$ to 0.8 Volts

Timing charts are provided in Figures 2.4 to 2.8.

DMA CYCLE (1Byte from FDC → MEMORY)

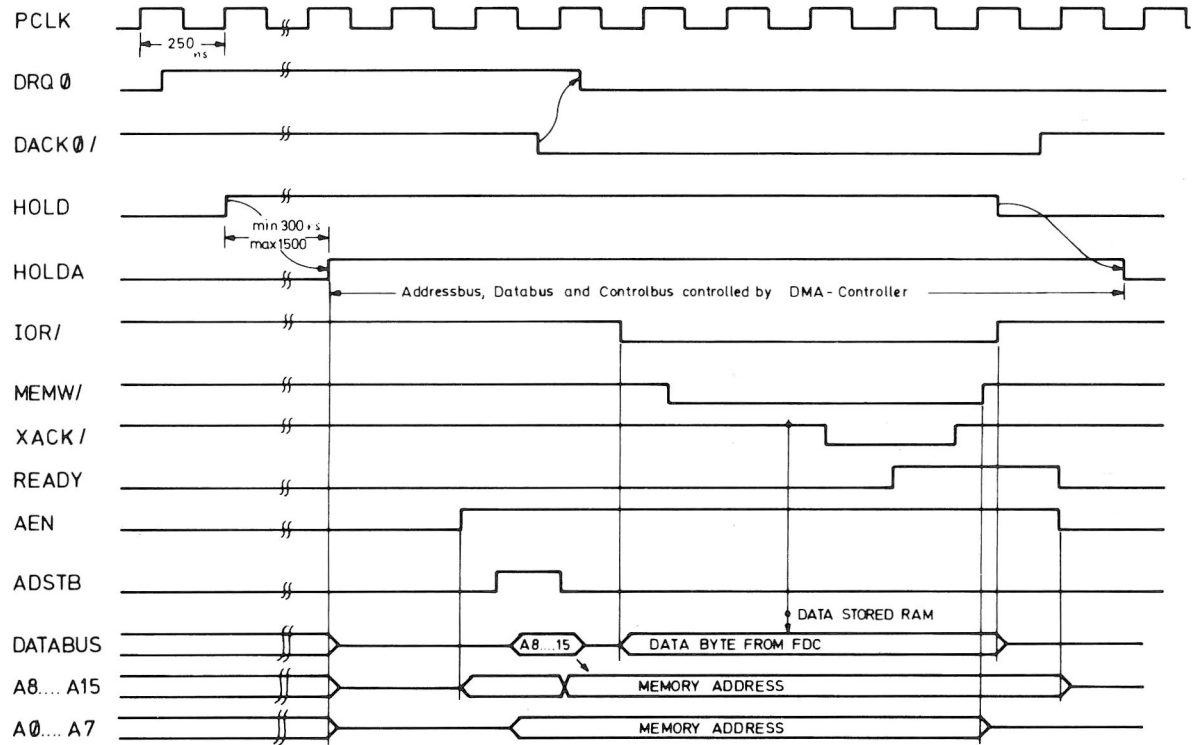
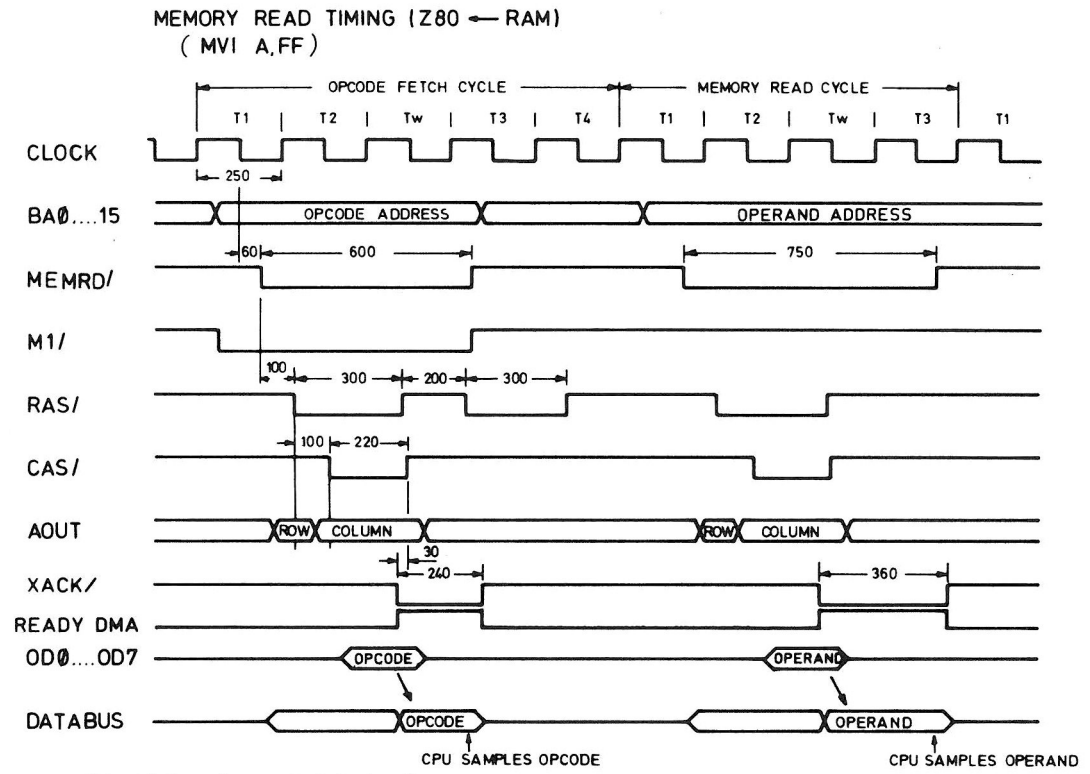


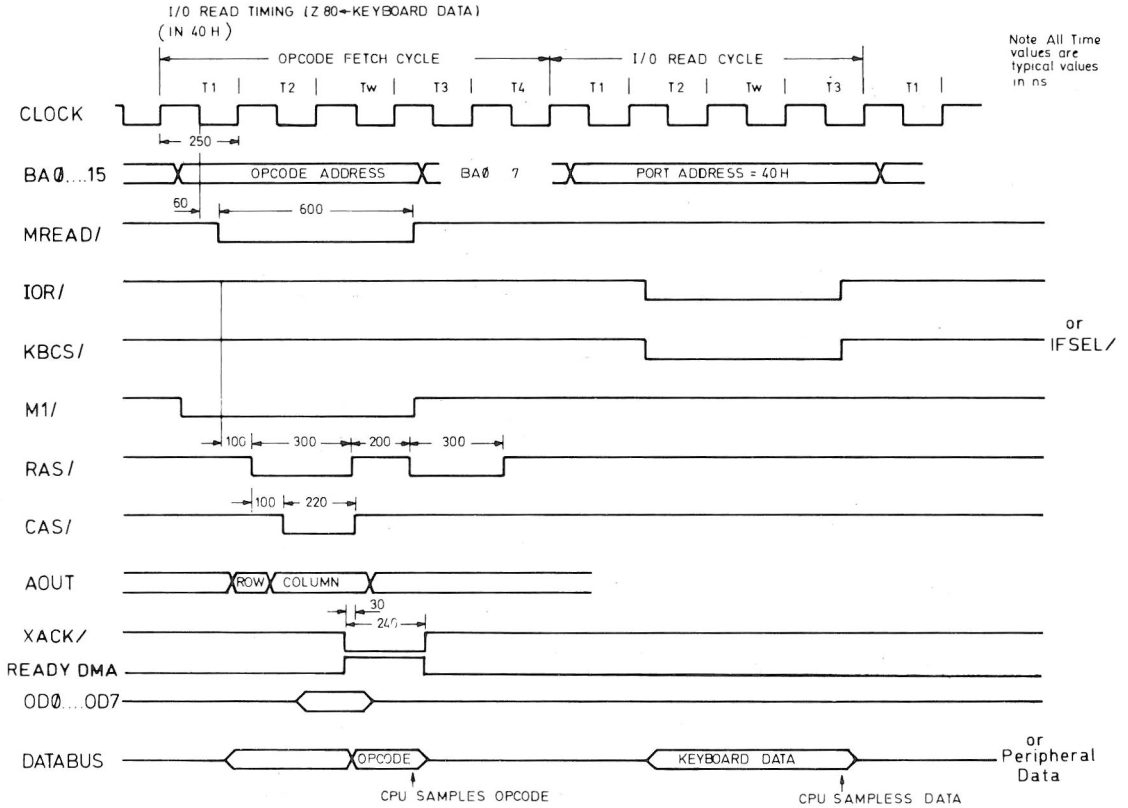
Figure 2.4 DMA cycle timing



Note: All Time values are typical values in ns

Figure 2.15 Memory read timing

Figure 2.6 I/O read timing



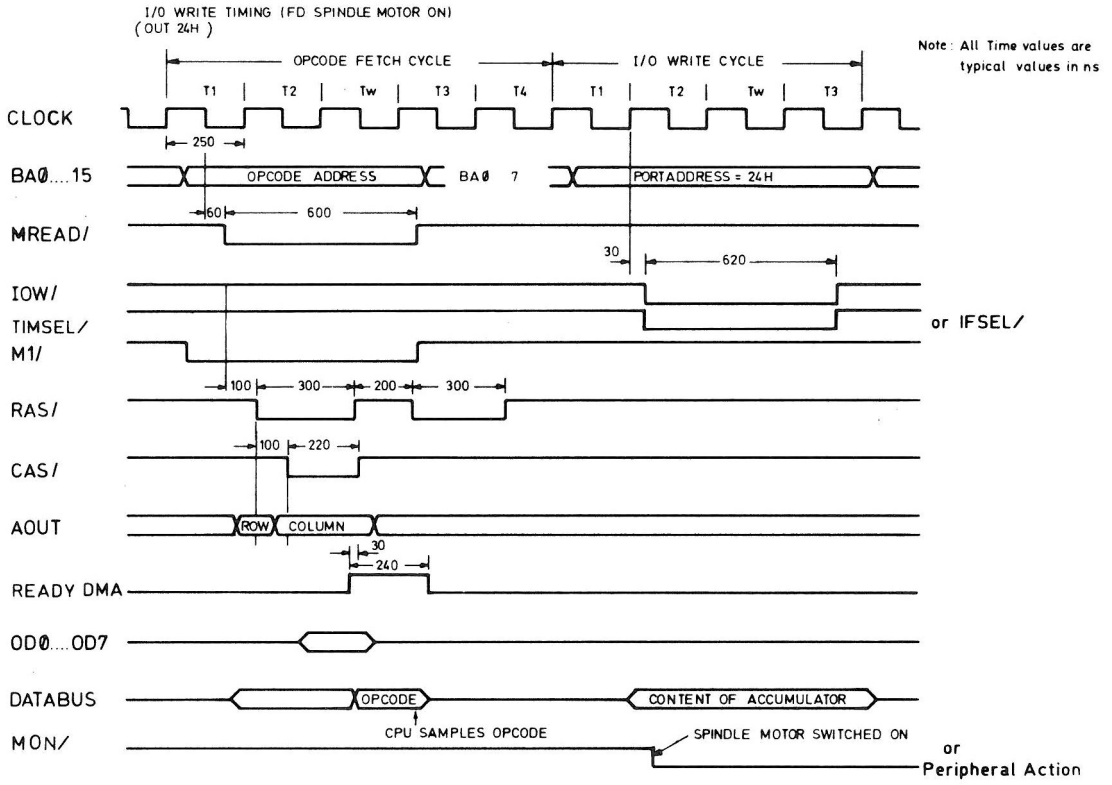


Figure 2.7 I/O write timing

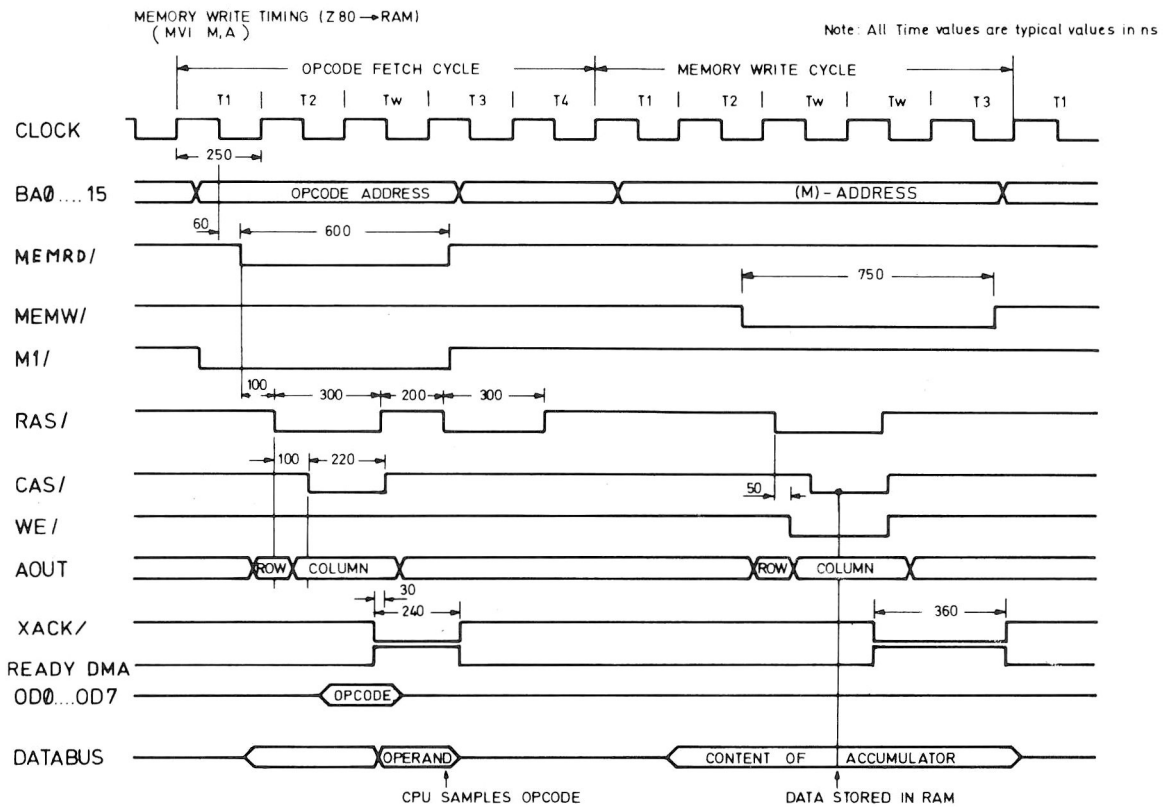


Figure 2.8 Memory write timing

I/O Signals

The signals for the NCR DECISION MATE V I/O bus are defined in Figure 2.9. The distribution of the signals is shown in Figure 2.10.

Pin Name	Pin Number	I/O	Description	DC Characteristics Driver or Receiver Re- quirements of each Peripheral
+5	a1 c1	— —	+5V Supply	I max. = 0.5A
+12	c2	—	+12V Supply	I max. = 0.1A
LGRD	c16 a32 c32	— — —	Logic ground	
BD0	c6	I/O	Data-Bus lines (8-bit) bidirectional, active high	Driver: $I_L > 2\text{mA}$
BD1	a6	I/O		or $I_L > 1.6\text{mA}$
BD2	c7	I/O	Bus-Driver to peripheral BUS (LS245), automatic direction.	by $V_L < 0.45\text{V}$
BD3	a7	I/O		$I_H > -0.2\text{mA}$
BD4	c8	I/O	Switch by IFSEL (I/O-Read). Normal output.	Receiver: $I_L < -0.2\text{mA}$
BD5	a8	I/O		$I_H < 20\mu\text{A}$
BD6	c9	I/O	Direction can be changed by DIR/ signal.	
BD7	a9	I/O	Typical IC's in the interfaces are 74LS245 or Port 8255.	

Figure 2.9 I/O signals (1 of 5)

Pin Name	Pin Number	I/O	Description	DC Characteristics Driver or Receiver Re- quirements of each Peripheral
BA0	c26	0	Buffered 20-bit Address Bus	Receiver: $I_L < -1.2\text{mA}$
		0		$I_H < 60\mu\text{A}$
		0	Range to 1MB Normal output, active high	Spec. Operation
		0		Driver: (Only Tristate)
		0	Tri-State possible with ABTRI/ signal.	$I_L > 24\text{mA}$
BA19	a17	0		$I_H > -10\text{mA}$
IOW/	a4	0	I/O WRITE	Receiver: $I_L < -1.2\text{mA}$
IOR/	c4	0	I/O READ	$I_H < 60\mu\text{A}$
MEMW/	a5	0	MEMORY WRITE	Special Operation
MEMR/	c5	0	MEMORY READ	Driver: $I_L > 24\text{mA}$
			Processor control lines, active low. Tri-state possible	(Only $I_H > -5\text{mA}$ Tristate)
RESET/	a3	0	System Reset Output, active low.	Receiver: $I_L < -0.4\text{mA}$
				$I_H < 20\mu\text{A}$

Figure 2.9 I/O signals (2 of 5)

Pin Name	Pin Number	I/O	Description	DC Characteristics Driver or Receiver Re- quirements of each Peripheral
RESET IN/	c3	I	RESET IN/ is an active low signal, which is used to general system RESET. Internal is a RC circuit and a Schmitt trigger input. A RESET is possible for example by a switch to GRD.	Driver open collector $I_L > 40\text{mA}$ Short protection
ABTRI/	c10	I	Address Bus Tri-state, active low signal.	Driver only open collector $I_L > 8\text{mA}$
READY DMA	a10	O	Ready Signal from the memory controller (XACK), active high.	Receiver: $I_L < -1\text{mA}$ $I_H < 50\mu\text{A}$
DIR/	c13	I	Direction of the databus driver. Low signal change to input.	Driver open collector $I_L > 8\text{mA}$
THOLD/	a14	I	Test Hold. External request to set the Z80 in hold state.	Driver open collector $I_L > 8\text{mA}$
HLDA	c14	O	Hold Acknowledge. A response from Z80 processor. The Z80 processor is in hold state, active high.	Receiver: $I_L < -0.8\text{mA}$ $I_H < 40\mu\text{A}$
PCLK/	a15	O	Processor-Clock: Inverse signal of the processor lock (Z80) with a frequency of 4MHz.	Receiver: $I_L < -0.8\text{mA}$ $I_H < 40\mu\text{A}$

Figure 2.9 I/O signals (3 of 5)

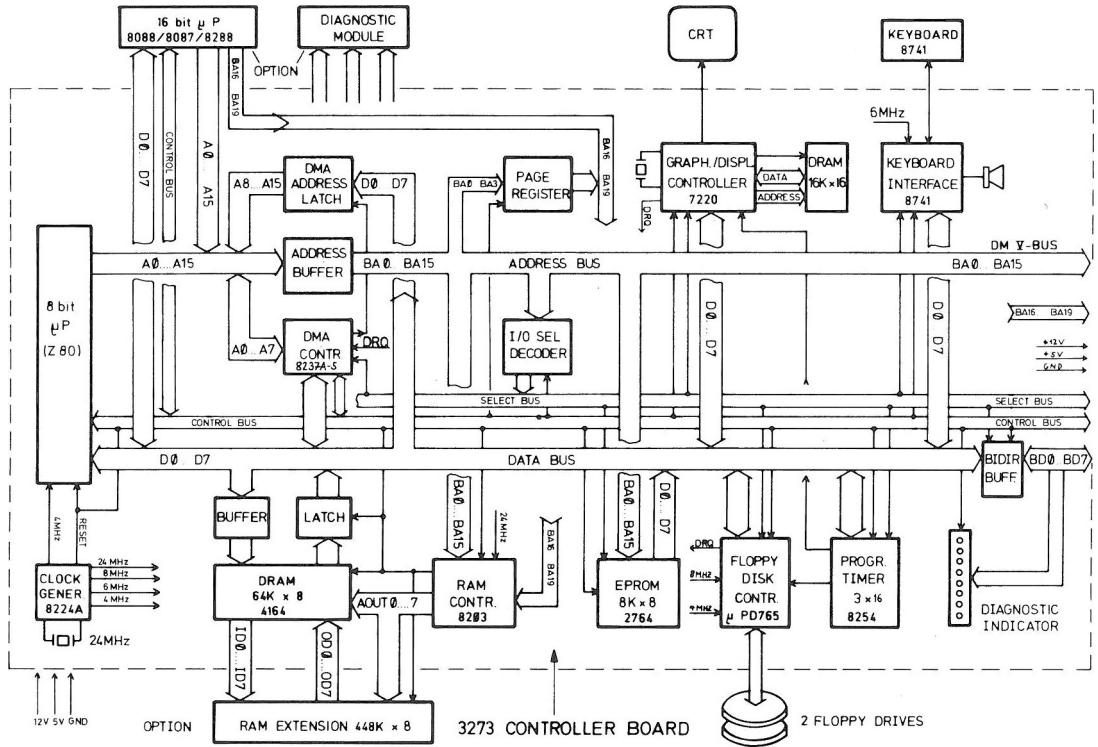
Figure 2.9 I/O signals (4 of 5)

Pin Name	Pin Number	I/O	Description	DC Characteristics Driver or Receiver Re- quirements of each Peripheral
CLK1/	c15	0	Clock output 1MHz.	$I_L < -0.8\text{mA}$ $I_H < 40\mu\text{A}$
TRAMD/	c16	I	Test RAM-Disable. For external ROM or RAM extension. Switching with the system RAM. The RAM output will be disabled. Used by the Diagnostic module. The signal is active low.	Driver only, open collector $I_L > 8\text{mA}$
INT/	c31	I	INTERRUPT. Interrupt for Z80, active low.	Driver only, open collector $I_L > 8\text{mA}$
INTACK/	a12	0	INTERRUPT ACKNOWLEDGE from the Z80 Processor, active low.	Receiver: $I_L < -1.2\text{mA}$ $I_H < 60\mu\text{A}$
WAIT/	a31	I	WAIT/ set the Z80, 8088, and DMA controller in Wait-State (not ready), active low	Driver only, open collector $I_L > 8\text{mA}$
DRQ0	c29	1	DMA-Request for Channel 0, active high	Driver only Tristate
DRQ1	a29	1	DMA Request for Channel 1, active high. Asynchronous channel request inputs are used by peripherals to request DMA service.	$I_H > -2\text{mA}$ $I_L > 2\text{mA}$
DACK0/	c30	0	DMA-Acknowledge Channel 0, active low.	Receiver: $I_L < -0.5\text{mA}$
DACK1/	a30	0	DMA-Acknowledge Channel 1, active low. These lines indicate an active DMA channel.	$I_H < 20\mu\text{A}$

Pin Name	Pin Number	I/O	Description	DC Characteristics Driver or Receiver Requirements of each Peripheral
EOP/	a11	I/O	End of Process-EOP signals that DMA service has been completed. Also input active low. For details 8237A-5 Spec.	Receiver: $I_L < -0.4\text{mA}$ $I_H < 20\mu\text{A}$ only for 3 Peripherals Driver open Collector $I_L > 5\text{mA}$
IFSEL0/ IFSEL1/ IFSEL2/ IFSEL3/ IFSEL4/	c28 a28 c27 a27 c12	0 0 0 0 0	Interface select 0 to 4, active low. The Interface does not need own address decoder. The Select of the I/O ports in the peripherals is made by any IFSEL and BA3 (BA3/). (10 Peripherals). Automatic change of the data bus direction. This change is not possible while a DMA function is being performed.	Receiver: $I_L < -0.8\text{mA}$ $I_H < 40\mu\text{A}$
AUTO/	a13		Change peripheral from board type 1 to board type 2. Not connected on board type 1.	Receiver: $I_L < 1.2\text{mA}$ $I_H < 60\mu\text{A}$ Pull-up resistor 1-10 Kohm to +5V

Figure 2.9 I/O signals (5 of 5)

Figure 2.10 Signal distribution



Interface Configuration

The configuration of the I/O interface and information on the interface select signals is provided in Figure 2.11.

NCR-DECISION MATE V Interface Configuration												
Interface Selects		Family:										
IFSEL/ PIN's on DM V BUS	0A IFSEL 0/ c28	0B	1A IFSEL 1/ a28	1B	2A IFSEL 2/ c27	2B	3A IFSEL 3/ a27	3B	4A IFSEL 4/ c12	4B	SLOT	DMA CHANNEL 0 or 1
PRINTER 1 serial/parallel K212 K210	■	■									2-6	
PLOTTER (See Note 2) K213	■	■									2-6	
COMMUNICATION K211			■	■							2-6	
SWITCHABLE RS-232C K801	■	■	■	■	(See Note 1)						2-6	
PLOTTER (See Note 2) K801	■				(WITH CP/M or MS-DOS)						2-6	
PLOTTER (See Note 2) K801					(WITH p-SYSTEM)					■	2-6	
BUFFERED SYNC/ ASYNC (See Note 3) K215			■								2-6	
REAL-TIME-CLOCK (See Note 3) K803										■	2-6	
IEEE 488 (See Note 3) K804						■					2-6	
MOUSE INTERFACE (See Note 3) K806					■						2-6	
DECISION NET K600									■		2-6	1
HARD DISK ext. (NCR 3282)									■		2-6	
HARD DISK int.									■		2-A	
PORT ADDRESS	60H 67H	68H 6FH	70H 77H	78H 7FH	30H 37H	38H 3FH	80H 87H	88H BFH	C0H C7H	C8H CFH		
No IFSEL for 16-Bit - Ext., Diagonser, Memory 64K, 192K, 448K.												
■ = Software and Hardware, ■ = Hardware.												
Note 1: Must be strappable by hardware and software												
Note 2: Use either K213 or K801												
Note 3: Any software must be able to select all 10 port addresses												
Note 4: Any port address cannot be used more than once in a system												

Figure 2.11 Interface configuration

GRAPHICS CONTROLLER BOARD

With the 32Kbytes of graphics memory, the system can create a 640 x 400 resolution image. Upper and lower case characters are displayed in a 6 x 9 dot pattern on a 8 x 16 dot matrix which allows for descended lower case characters. The graphic controller operates in mixed mode. It can draw the following figures: horizontal and vertical lines, vectors, rectangles, arcs, circles, graphic character symbols, line pattern or area pattern, and slanted figures (+n x 45). Also, the CRT controller supports scrolling, zooming set mask, reset mask, replace complement and clear screen. Reverse and normal video, blink attribute and half intensity are also available. (See NEC 7220 data sheets)

The graphics controller board is connected directly onto the controller board with two connectors, J/P 113, and J/P 114. The pin assignments for these connectors are shown in Figure 2.12.

a		c	a		c
+5 V	1	+5 V	+5V	1	+5V
LGRD	2	LGRD		2	
LPEN	3	LPENSW	D1	3	D0
+12 V	4	+12 V	D3	4	D2
+12 V	5		D5	5	D4
	6		D7	6	D6
	7			7	
	8		HSYNCX	8	HSYN
	9		VSYNX	9	LGRD
	10		BA1	10	BA0
	11			11	
	12		GDCIOW/ DACK2/	12	GDCIOR/ GDRQ2
	13			13	
	14			14	
	15	LGRD		15	WCLK
LGRD	16	LGRD	LGRD	16	LGRD

Pin assignments P114

Pin assignments P113

Figure 2.12 Graphic controller connections

The graphics controller board for color systems has a memory capacity of 96Kbytes, and can display eight colors — black, white, red, green, blue, yellow, magenta, and cyan.

In the text mode the characteristics are:

- 25 lines, 80 characters per line
- Characters in any of the eight colors
- Background in any of the eight colors
- Blinking attribute

In the graphics mode the characteristics are:

- Each dot position in any of the eight colors
- No attributes

16-BIT PROCESSOR

The 16-bit processor can be installed in an NCR DECISION MATE V, allowing the computer to function as either an 8- or a 16-bit processor.

The 16-bit processor is available in two versions; both versions function identically. The two versions are:

- A factory-installed feature (F230), or an engineer-installed kit (K230); the board is connected to the solder side of the controller board at P/J7A. This version is not installable by the user.
- A self-contained module (K231) which the user can plug directly into the controller bus at slot 7.

The configuration of the 16-bit processor board is:

- 8088 processor with 5 MHz clock
- Bus controller
- 4 K x 8 PROM
- Spare socket for a co-processor (e.g. 8087)
- Switch logic for dual processor concept

Because the Z80A and the 8088 processor use the same bus, only one processor will work at a time. The other processor is in HOLD state. Both processors have equal rights and will perform all memory and I/O operations.

Only the operating processor can perform switching to the other processor.

The 16-bit processor operates in maximum mode.

Contents of the PROM:

Address range FF000-FFFFF

Start address after Reset FFFF0

Jump to start point of the operating system

Jump to start point of the operating system Level 0 diagnostic

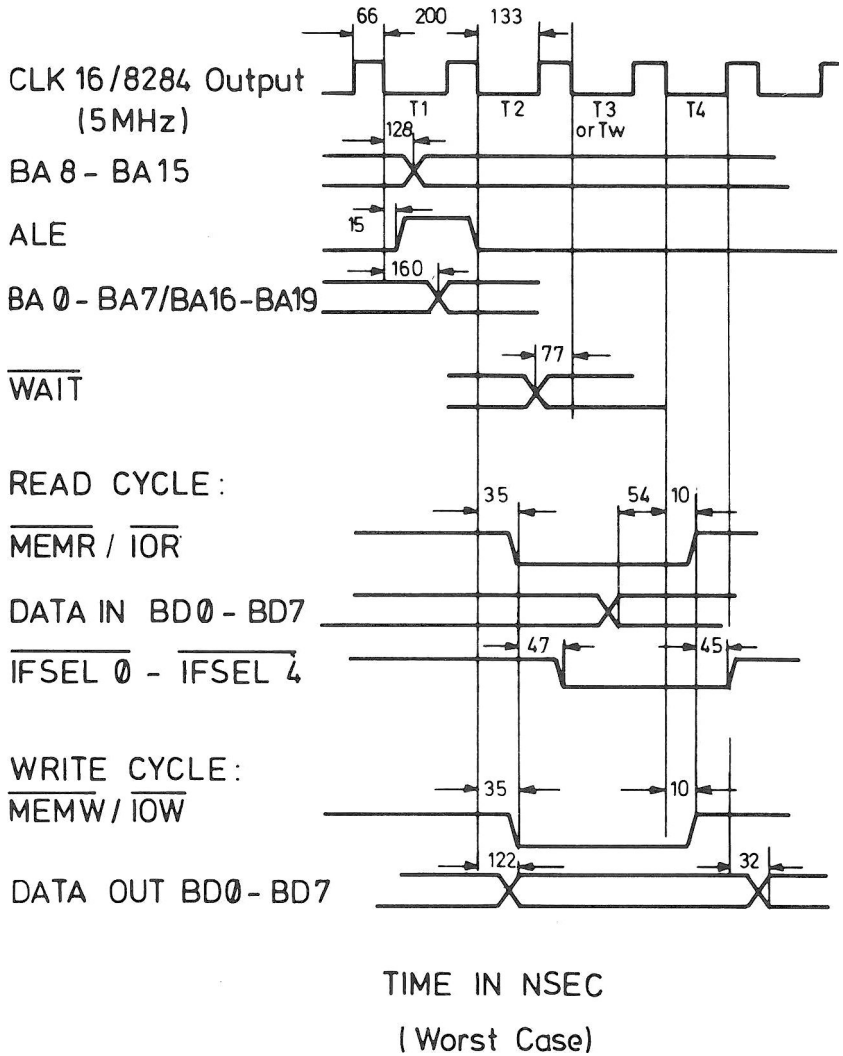


Figure 2.13 16-bit processor timing

The processor timing of the 16-bit processor is shown in Figure 2.13, and the pin assignments in Figure 2.14.

	a	b	c
1	+5 V	+5 V	+5 V
2	OPT 2		+12 V
3	RESET/		RESETIN/
4	IOW/		IOR/
5	MEMW/		MEMR/
6	D1		D0
7	D3		D2
8	D5		D4
9	D7		D6
10	READYDMA	READYP	ABTRI/
11	EOP/	HOLD	
12	INTACK/	SWITCH 16/	IFSEL4/
13	AUTO/	HLDA 16	DIR/
14	THOLD/	16 BITAV/	HLDA
15	PCLK/	STDMARQ/	CLK1
16	LGRD	LGRD	TRAMD/
17	BA19	16 BITSET /	BA18
18	BA17		BA16
19	A15		A14
20	A13	MEMRQ/	A12
21	A11		A10
22	A9		A8
23	A7		A6
24	A5		A4
25	A3		A2
26	A1		A0
27	IFSEL3/		IFSEL2/
28	IFSEL1/		IFSEL0/
29	DRQ1		DRQ0
30	DACK1/		DACK0/
31	WAIT/		INT/
32	LGRD	LGRD	LGRD

Figure 2.14 16-bit processor pin assignments

POWER SUPPLY

The power supply comprises the following modules:

- Power connector
- Primary fuse
- Transformer
- Rectifier
- Switcher (+5V, +12V)
- 1 Secondary Fuse for unregulated output voltage

The power supply is capable of operating with the following ac power sources.

Frequency	Nominal Voltage	Range
49 to 61 Hz	100	90 – 107
	120	104 – 127
	220	198 – 235
	230	207 – 246
	240	216 – 257

The average ac input wattage is 70W (Basic Model w/o peripherals). The outputs of the standard power supply are:

Supply	Regulated	Max. Current	Ripple
+18V to +35V	+ 5.1V± 3%	10.0 A	< 50mVpp
+18V to +35V	+12.2V± 3%	3.2 A	< 120mVpp

Models with hard/flexible disk combination require a larger transformer and switcher assembly. The outputs of these are:

Supply	Regulated	Max. Current	Ripple
+18V to +35V	+ 5.1V± 3%	13.0 A	< 50mVpp
+18V to +35V	+12.2V± 3%	4.1 A	< 120mVpp

The color CRT also requires a larger transformer, but the standard switcher and rectifier can be used for controller board, flexible disks, keyboard, and I/Fs. The rectifier and switcher for the color CRT are placed on the CRT board.

Output power requirements for the color CRT are:

Regulated	Max. Current	Ripple
107V± 2V	0.5 A	< 400mVpp

Current limiting and over voltage protection are provided. The electrical power is enough for both the computer and its interfaces and options, but printers and free-standing disk drives require their own ac power source. The following power line transients should not affect the specified performance:

Input Voltage	Duration	Frequency of Occurrence
50% of nominal	1/2 cycle	Once in 10 seconds

The power supply assembly is available in three models:

- For a power source with a nominal voltage of 120 Vac
- For a power source with a nominal voltage of 220 Vac
- A model that can be strapped to suit power sources with nominal voltages of 100, 230, or 240 Vac

These different models of the power supply are identified by the "Power Code," which is a part of the model number.

Power Code	Nominal Voltage
60	120 Vac
65	220 Vac
70*	100 Vac
74*	230 Vac
75*	240 Vac

* Selectable by transformer strapping

The strapping of the transformer for power supplies with power codes 70, 74, and 75 is shown in Figure 2.15.

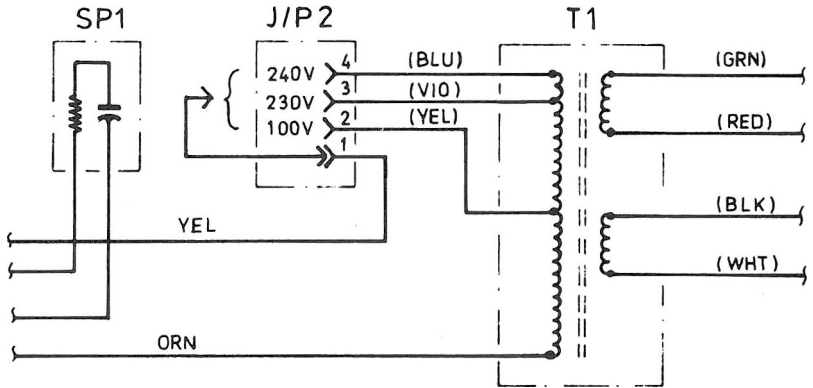


Figure 2.15 Transformer connections

FLEXIBLE DISK DRIVES

Integrated in the computer are one or two 5 1/4-inch flexible disk drives to provide mass storage of programs and data for the computer. The drives contain read/write electronic controller electronics, driver mechanics, read/write heads, and headpositioning mechanism. The disk drive on the left is called A, the second drive on the right side is called B. For maximum motor life, if no commands have been issued to the drives within 15 seconds after completion of a previous command, the drive motor is turned off. The head load mechanism is coupled with motor on signal.

Systems with an integrated fixed disk have only one flexible disk drive, this is mounted on the rightmost side of the computer and is called drive A.

Power requirements for the flexible disk drive are given in Figure 2.16, and technical data in Figure 2.17.

Voltage	Current	Ripple
+ 5V +5%	Typical 0.5 A max. 0.8 A	<100 mVpp
+12V +5%	Typical 0.25 A max. 0.9 A	<200 mVpp
Max. power dissipation: less than 10W.		

Figure 2.16 Power requirements

Capacity (double-sided, double density) unformatted	500 Kbytes
Formatted (CP/M)	320 Kbytes
Formatted (MS-DOS)	360 Kbytes
Tracks per inch	48
Tracks per disk side	40
Sectors per track (CP/M)	8
Sectors per track (MS-DOS)	9
Bytes per sector	512
Motor start time	<400msec
Head movement (track to track)	<6msec
Data transfer time	250 bits/sec

Figure 2.17 Technical data

The flexible disk drives use the standard pin assignments as shown in Figure 2.18.

READY/	34	33	GRD
SIDSEL/	32	31	GRD
RDDATA/	30	29	GRD
WRPROT/	28	27	GRD
TRACK0/	26	25	GRD
WREN/	24	23	GRD
WRDATA/	22	21	GRD
STEP/	20	19	GRD
DIR/	18	17	GRD
MOTORON/	16	15	GRD
	14	13	GRD
SEL1/	12	11	GRD
SEL0/	10	9	GRD
INDEX/	8	7	GRD
	6	5	GRD
	4	3	GRD
HDLOAD/	2	1	GRD

Figure 2.18 Pin assignments

The input and output signals to the disk drive are standard, industry-compatible signals. These signals are:

Drive Select (0.1.2.3)

Active low input signal to select one of four FDD. Drive Select active turns the motor on, loads the heads, and turns on the front panel LED of the selected drive.

Index

Active low output signal which is at an active level each time the index hole is sensed. This signal is active for one pulse each disk revolution to indicate the beginning of a track. Index should be held active when no flexible disk is inserted in the system.

Motor On

Active low input signal to turn the motor on. Time is allowed by the system before reading or writing to allow the motor to start. The line is deactivated after 2 seconds (nominal) following the last command. This turn off signal may be varied by the host system depending on application.

Direction Select

Input signal to define the direction the heads move when the Step line is pulsed. Active low causes the head to move toward the center of the disk. Active high causes the head to move toward the outside of the disk.

Step

Active low input signal to move the head in the direction specified by Direction Select. Minimum pulse width for step is $0.8\mu\text{s}$. Direction Select is not changed within 1ms on each side of the trailing edge of Step. Each step pulse is delayed by 6ms from the preceding step.

Write Data

Input signal to provide the data to be written on the flexible disk. Each transition from high to low causes the current through the read/write heads to reverse, causing a data bit to be written. This line is enabled by Write Gate active. Write Data is not active during a read operation. Write pulse width for both data bit and clock pulse shall range more than $0.25\mu\text{s}$.

Write Gate

Active low input signal to enable data to be written on the flexible disk. Write Gate not active enables the stepper logic and read data logic.

Track 00

Active low output signal to indicate when the read/write heads are positioned at track zero. Track 00 is not active when the read/write heads are not at track zero.

Write Protect

Active low output signal to indicate when a write protected disk is installed in the drive. The drive inhibits writing and provides the write protect signal.

Read Data

Output signal containing the "Raw Data" (clock and data together) as detected by the drive electronics. The Read Data signals are composite clock and data pulses of 1 microsecond, ± 250 nanoseconds.

In Use

Active low input signal indicates that all of the daisy changed FDDs are under the control of the system, and the indicator on the front panel is on. (not used)

Ready

Active low output signal indicates that the FDD is in ready state; it is satisfied when the FDD is powered on, the disk is installed, the disk rotates at nominal speed, and two Index pulses have been counted after disk starts rotation. Required time for this signal to become true after the Motor On signal becomes true is less than 800 ms.

FIXED (WINCHESTER) DISK

The Winchester disk drive uses two non-removable 5 1/4-inch disks as storage media. The total formatted capacity of the four surfaces is 10 Mbytes. The two disks are called disk B and disk C. The Western Digital Winchester disk controller interfaces the disk drive to the host processor. All necessary buffers and receivers/drivers are included on the Winchester disk controller board to allow direct connection to the drive. Between the disk controller board and main controller board, a small interface board for bus adaption is included.

Power requirements for the fixed disk drive are given in Figure 2.19, and technical data in Figure 2.20.

Voltage	Current	Ripple
+12V $\pm 5\%$	1.6 A typical, 3.5 A max.	>50mVpp
+ 5V $\pm 5\%$	1.1 A typical, 1.7 A max.	>50mVpp

Figure 2.19 Power requirements

Capacity (formatted)	10 Mbytes
Tracks per inch Tracks per unit	345 612
Sectors per track (1 as alternative sector)	17
Bytes per sector	512
Motor start time Head movement (track to track) (average)	15 sec 3msec 85msec
Date transfer time	5 Mbits/sec

Figure 2.20 Technical data

The technical details of the Western Digital disk controller are shown in Figure 2.21.

Encoding method	MFM
Sectoring	Soft
Host Interface	8-bit bi-directional bus
Data Transfer Rate	5.000 Mbits/sec.
Power Requirements	5V±5%, 3A max.

Figure 2.21 Controller technical data

The input and output signals for the drive are:

Write Gate

The active state of this signal, or logical one level, enables data to be written on the disk. The inactive state of this signal, or logical zero level, enables data to be transferred from the drive. During power fail or power up, all data recorded on the drive shall be retained reliably, regardless of the frequency of occurrence of dc power transitions at the drive itself, and regardless of the sequence and rate of dc voltage decay or rise; the user must provide the WRITE GATE input with an open circuit (inactive, high impedance driver) during these conditions.

Head Select 2⁰ and 2¹

These two lines provide for the selection of each individual read/write head in a binary coded sequence. Head Select 2⁰ is the least significant line. Heads are numbered 0 through 3. When both Head Select lines are false, head 0 will be selected.

Direction In

This signal defines direction of motion of the R/W head when the Step line is pulsed. A logical zero defines the direction as "out" and if a pulse is applied to the Step line, the R/W heads will move away from the center of the disk. If this line is a logical one, the direction of motion is defined as "in" and the R/W heads will move toward the center of the disk. Direction must not change during step time.

Step

This interface line is a control signal which causes the R/W head to move with the direction of motion defined by the Direction In line. Any change in the Direction In line must be made at least 100ns before the leading edge of the step pulse. The buffered seek mode is automatically selected any time step pulses are issued at a rate between 25 and 500 μ sec. The minimum time between successive steps is 3.0mS. In the slow step mode, the R/W heads will move at the rate of the incoming step pulses.

Drive Select 1-4

Drive Select, when a logical one connects the drive interface to the control lines and activates the LED on the front panel of the drive. Shunts are provided on the drive which can be cut in a specified pattern so as to determine which unique select line (DS1-4) on the interface will activate that particular drive.

MFM Write Data

This is a differential pair that defines the transitions to be written on the track. The transition of +MFM Write Data line going more positive than the -MFM Write Data will cause a flux reversal on the track, provided Write Gate is active. This signal must be driven to an inactive state (+MFM Write Data more negative than -MFM Write Data) by the host system when in a read mode.

To ensure data integrity at the error rate specified, the write data presented by the host must be pre-compensated on tracks 128 through 305. Data patterns which cause a large amount of bit shift will have appropriate data bits shifted early or late with respect to the nominal bit cell position. Bit shift compensation, whether early or late with respect to the nominal bit cell position, will be 12ns.

Seek Complete

This line will go from a logic zero to a one when the R/W heads have settled on the final track at the end of a seek. Reading or writing should not be attempted when Seek Complete is a logic zero.

Seek Complete must go inactive for any of the following cases:

1. A recalibration sequence is initiated (by drive logic) at power on or after power failure because R/W heads are not over track zero.
2. 12 μ sec (typical) after the leading edge of a step pulse or series of step pulses.
3. The dc voltage fault portion of the write fault circuit is activated.

Track 0

This interface signal indicates a logic one only when the drive's R/W heads are positioned at track zero (the outermost data track). When the Track zero signal goes 'true,' it shall always be 'true' prior to Seek Complete 'true.'

Write Fault

This signal indicates that a condition exists at the drive that will cause improper writing on the disk. When this line is a logic one, further writing shall be inhibited at the drive until the condition is corrected. It cannot be reset via the interface.

There are four conditions which will cause Write Fault to be activated:

- Write current malfunction:

Write current in a head without Write Gate active

or

Write Gate and Drive Select active with no write current in a head.

- A drive malfunction causing more than, or less than, one head to be selected, or a malfunction causing a head to be selected for read during write or for write during read.
- dc voltages are grossly out of tolerance.
- Write Gate and an inactive condition on Seek Complete.

Index

This interface signal is provided by the drive once each revolution (16.67mS nom.) to indicate the beginning of the track. This signal is normally logic zero (normally high) and makes the transition to logical one to indicate Index. Only the transition from high to low is valid.

Ready

This interface signal when a logic one together with Seek Complete, indicates that the drive is ready to read, write, or seek, and that the I/O signals are valid. When this line is a logic zero, all writing, reading, and seeking are inhibited. Ready shall be inactive only as a result of and during recovery from a power off condition or whenever the dc voltage monitor portion of the Write Fault circuit is activated.

MFM Read Data

The data recovered by reading a pre-recorded track is transmitted to the host system via a differential pair of MFM Read Data lines. The transition of the +MFM Read Data line going more positive than the -MFM Read Data line represents a flux reversal on the track of the selected head.

KEYBOARD

The low profile ASCII encoded keyboard is designed according to NCR Corporate Appearance Plan (CAP) styling and newest ergonomic requirements. It is connected to the system logic by a four-wire twisted cable, non-extended length 0.5m.

Key arrangement consists of standard alpha keys, the 10-key numeric keypad, cursor control keys, 20 programmable function keys with escutcheon for function key description, and following features:

- Automatic repeat entry on all data keys
- Rollover key feature for fast operation
- Double zero key
- Word processing mode (shift inversion lock for alpha keys only)
- 8-byte FIFO buffer (plus 20 bytes on controller)
- Simple decimal point/comma exchange

The keyboard is currently available in two versions to provide the languages shown in Figure 2.22.

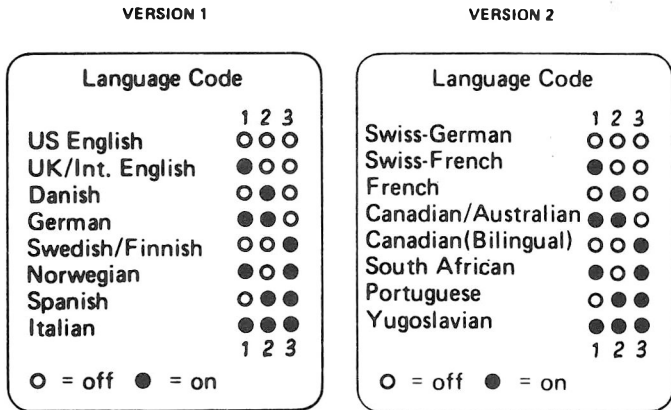


Figure 2.22 Keyboard languages

The selection of the required language is made by the switch assembly mounted under the base of the keyboard, see Figure 2.23.

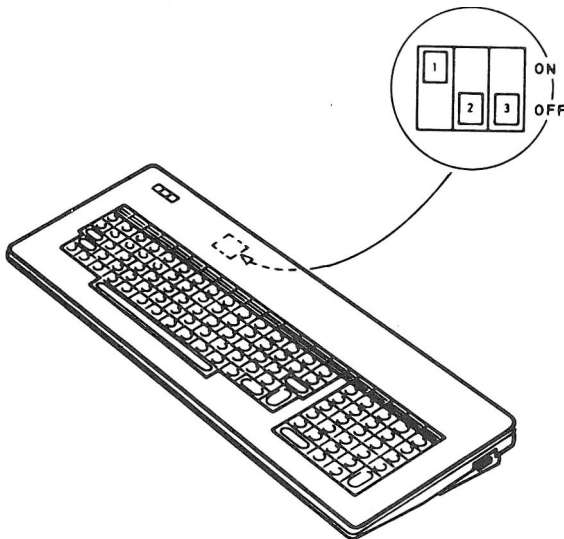


Figure 2.23 Language code selection

Keytip sets are provided so that the keyboard layout matches the standard national layouts; see Figure 2.24 for version 1 keyboard, and Figure 2.25 for version 2.

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	!	@	#	\$	%	^	&	*	()	_	=	←	TAB
CONTROL	Q	W	E	R	T	Y	U	I	O	P	{	}	CONTROL	
	↓	A	S	D	F	G	H	J	K	L	;	;	~	
	↑		Z	X	C	V	B	N	M	<	>	?	↑	↓

F16	F17	F18	F19	F20
↘	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	.		↓

US ENGLISH

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	!	..	£	\$	%	&	'	()	_	=	~	←	TAB
CONTROL	Q	W	E	R	T	Y	U	I	O	P	@	{	CONTROL	
	↓	A	S	D	F	G	H	J	K	L	+	*	}	
	↑		Z	X	C	V	B	N	M	<	>	?	↑	↓

F16	F17	F18	F19	F20
↘	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	.		↓

UK/INTERNATIONAL ENGLISH

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	!	..	£	\$	%	&	/	()	=	?	\	←	TAB
CONTROL	Q	W	E	R	T	Y	U	I	O	P	Å	¨	CONTROL	
	↓	A	S	D	F	G	H	J	K	L	Æ	Ø	*@	
	↑	>	Z	X	C	V	B	N	M	;	:	=	↑	↓

F16	F17	F18	F19	F20
↘	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	,		↓

DANISH

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	!	..	§	\$	%	&	/	()	=	?-ß	\	←	TAB
CONTROL	Q	W	E	R	T	Z	U	I	O	P	Ü	*+	CONTROL	
	↓	A	S	D	F	G	H	J	K	L	Ö	Ä	#+	
	↑	>	Y	X	C	V	B	N	M	;	:	=	↑	↓

F16	F17	F18	F19	F20
↘	←	↓	↑	→
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	,		↓

GERMAN

Figure 2.24 Keyboard layouts (1 of 2)

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	
ESC	!	"	#	§	%	&	/	()	=	?	~	←	→	TAB	↖	←	↓	↑	→
CONTROL	Q	W	E	R	T	Y	U	I	O	P	^	~	CONTROL			CLR	7	8	9	/
	↓	A	S	D	F	G	H	J	K	L	O	^	*	Ⓜ		-	4	5	6	*
	↑	>	Z	X	C	V	B	N	M	;	:	=	↑	↓		+	1	2	3	↓
	<															0	00	,		↓

SWEDISH, FINNISH

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	
ESC	!	"	£	\$	%	&	/	()	=	?	~	←	→	TAB	↖	←	↓	↑	→
CONTROL	Q	W	E	R	I	Y	U	I	O	P	^	~	CONTROL			CLR	7	8	9	/
	↓	A	S	D	F	G	H	J	K	L	0	Æ	*	Ⓜ		-	4	5	6	*
	↑	>	Z	X	C	V	B	N	M	;	:	=	↑	↓		+	1	2	3	↓
	<															0	00	,		↓

NORWEGIAN

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	
FSC	!	"	£	\$	%	&	/	()	=	?	¿	←	→	TAB	↖	←	↓	↑	→
CONTROL	Q	W	E	R	T	Y	U	I	O	P	~	^	CONTROL			CLR	7	8	9	/
	↓	A	S	D	F	G	H	J	K	L	Ñ	~	*	Ⓜ		-	4	5	6	*
	↑	>	Z	X	C	V	B	N	M	;	:	=	↑	↓		+	1	2	3	↓
	<															0	00	,		↓

SPANISH

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	
ESC	!	"	£	\$	%	&	/	()	=	?	^	←	→	TAB	↖	←	↓	↑	→
CONTROL	Q	Z	E	.	T	Y	U	I	O	P	è	é	*	Ⓜ	CONTROL	CLR	7	8	9	/
	↓	A	S	D	=	G	H	J	K	L	ç	ò	*	Ⓜ		-	4	5	6	*
	↑	>	W	X	C	V	B	N	M	;	:	=	↑	↓		+	1	2	3	↓
	<															0	00	,		↓

ITALIAN

Figure 2.24 Keyboard layouts (2 of 2)

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	+ 1	" 2	* 3	§ 4	% 5	& 6	/ 7	(8) 9	= 0	? !	\ ^	←	TAB
CONTROL	Q	W	E	R	T	Z	U	I	O	P	ë ü	./	CONTROL	
↓	A	S	D	F	G	H	J	K	L	é ö	à ä	£ \$		
↑	≥	Y	X	C	V	B	N	M	;	:	=	↑		

F16	F17	F18	F19	F20
↘	→	↓	↑	←
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	,		↵

SWISS-GERMAN

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	+ 1	" 2	* 3	§ 4	% 5	& 6	/ 7	(8) 9	= 0	? !	\ ^	←	TAB
CONTROL	Q	W	E	R	T	Z	U	I	O	P	ü ë	./	CONTROL	
↓	A	S	D	F	G	H	J	K	L	ö é	ä à	£ \$		
↑	≥	Y	X	C	V	B	N	M	;	:	=	↑		

F16	F17	F18	F19	F20
↘	→	↓	↑	←
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	,		↵

SWISS FRENCH

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	1 &	2 é	3 "	4 ' /	5 (§	6)	7 à	8 ç	9 ù	0 ä	1 °	-	←	TAB
CONTROL	A	Z	E	R	T	Y	U	I	O	P	¨	* \$	CONTROL	
↓	Q	S	D	F	G	H	J	K	L	M	% ü	£ \		
↑	≥	W	X	C	V	B	N	;	:	/	=	↑		

F16	F17	F18	F19	F20
↘	→	↓	↑	←
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	,		↵

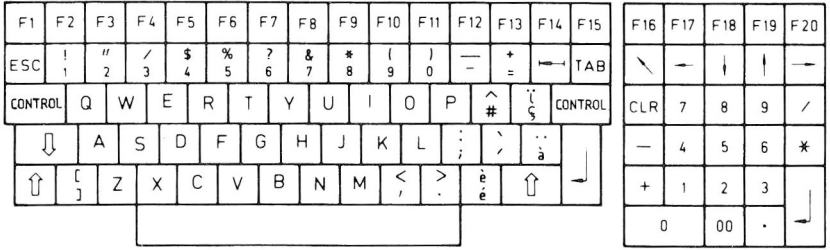
FRENCH

F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15
ESC	! 1	@ 2	# 3	\$ 4	% 5	¢ 6	& 7	* 8	(9) 0	-	+ =	←	TAB
CONTROL	Q	W	E	R	T	Y	U	I	O	P	[]	¨	CONTROL	
↓	A	S	D	F	G	H	J	K	L	;	;	"/	£ \	
↑	↑	Z	X	C	V	B	N	M	<	>	?	/	↑	

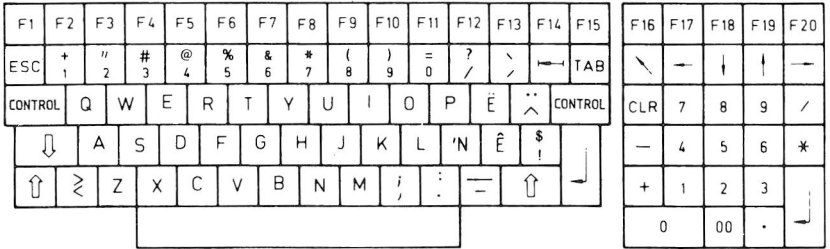
F16	F17	F18	F19	F20
↘	→	↓	↑	←
CLR	7	8	9	/
-	4	5	6	*
+	1	2	3	
0	00	.		↵

AUSTRALIAN (PRELIMINARY)

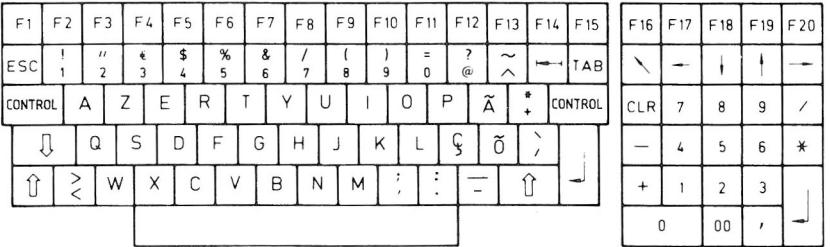
Figure 2.25 Keyboard layouts (1 of 2)



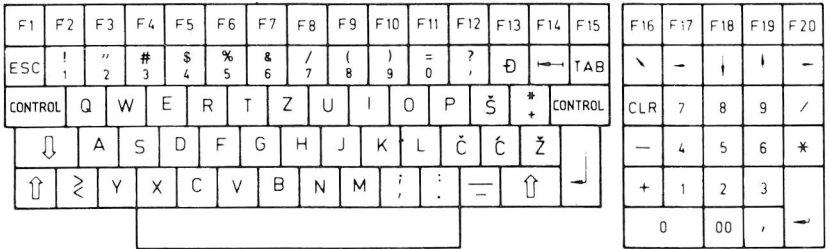
CANADIAN (BILINGUAL – PRELIMINARY)



SOUTH AFRICAN



PORTUGUESE



YUGOSLAVIAN

Figure 2.25 Keyboard layouts (2 of 2)

Technical details of the keyboard are provided in Figures 2.26 to 2.28.

Width	429 mm (16.9 in.)
Height	38 mm (1.5 in.)
Depth	216 mm (8.5 in.)
Weight	2 kg (4.4 lb.)

Figure 2.26 Keyboard dimensions and weight

Supply voltage	+5 Vdc
Current	100 mA (max.)
Logic level '1'	2.4 to 5.3 Vdc at max. 3mA
Logic level '0'	0 to 0.5 Vdc at max. 30mA

Figure 2.27 Keyboard voltage requirements

J109 Pin	Function	Color	Keyboard PCB (TB1)
1	Logic Gnd	black	4
2	Frame Gnd	shield	5
3	+5V	white	2
4	SDPOLL/ poll/data line	yellow	3
5	Reserved	red	1

Figure 2.28 Keyboard cable pin assignments

The joy stick is an option for program games. Functionally, the joy stick is connected parallel to the cursor keys, with a relationship between the arrows on the keyboard and the direction in which the joy stick is moved. The pin assignments for the joy stick connector are shown in Figure 2.29.

Pin	Assignment
1	Col 10
2	Col 9
3	Col 11
4	Col 12
5	Log. Gnd
6	Col 4
7	+5V
8	Row 4
9	Reserved

Figure 2.29 Joy stick pin assignments

KEYBOARD CODES

The layout of the keyboard is shown in Figure 2.31 and gives the key-switch position number for each of the 100 keys in the keyboard. Each key-switch can generate up to three different codes depending upon whether it is pressed alone, together with the shift key, or together with the Control key.

The codes generated by the alphabetic keys are changed from lower to uppercase or upper to lowercase when the Capital Mode key is pressed, for example, pressing the key in position C5 on the US English version of the keyboard produces the keycodes and characters shown in Figure 2.30.

Key(s)	Keycode	Character
C5	67	g
C5 with Shift	47	G
C5 with Capital Mode	47	G
C5 with Capital Mode and Shift	67	g
C5 with Control	17	

Figure 2.30 Keycode example

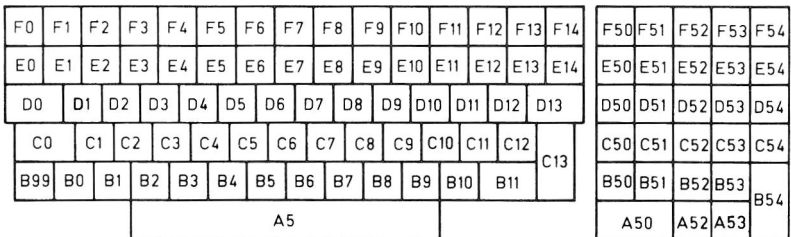


Figure 2.31 Keyboard switch positions

The keycodes for version 1 keyboards are shown in Figure 2.32, and for version 2 keyboards in Figure 2.33.

	US ENGLISH		UK/INT ENGLISH		DANISH		GERMAN					
	SHIFTED		CONTROL		UNSHIFTED		SHIFTED		CONTROL		UNSHIFTED	
POSITION	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED
A5	20	20	20	20	20	20	20	20	20	20	20	20
A50	30	30	30	30	30	30	30	30	30	30	30	30
A52	30	30	30	30	30	30	30	30	30	30	30	30
A53	2E	2E	-	2E	2E	-	2E	2E	-	2E	2E	-
B99	-	-	-	-	-	-	-	-	-	-	-	-
B0	5C	7C	1C	5C	7C	1C	3C	3E	-	3C	3E	-
B1	7A	5A	1A	7A	5A	1A	7A	5A	1A	79	59	19
B2	78	58	18	78	58	18	78	58	18	78	58	18
B3	63	43	03	63	43	03	63	43	03	63	43	03
B4	76	56	16	76	56	16	76	56	16	76	56	16
B5	62	42	02	62	42	02	62	42	02	62	42	02
B6	6E	4E	0E	6E	4E	0E	6E	4E	0E	6E	4E	0E
B7	6D	4D	0D	6D	4D	0D	6D	4D	0D	6D	4D	0D
B8	2C	3C	-	2C	3C	-	2C	3B	-	2C	3B	-
B9	2E	3E	-	2E	3E	-	2E	3A	-	2E	3A	-
B10	2F	3F	-	2F	3F	-	2D	5F	1F	20	5F	1F
B11	-	-	-	-	-	-	-	-	-	-	-	-
B50	2B	DB	BB	2B	DB	BB	2B	DB	BB	2B	DB	BB
B51	31	DC	BC	31	DC	BC	31	DC	BC	31	DC	BC
B52	32	DD	BD	32	DD	BD	32	DD	BD	32	DD	BD
B53	33	DE	BE	33	DE	BE	33	DE	BE	33	DE	BE
B54	88	88	98	88	88	98	88	88	98	88	88	98
C0	-	-	-	-	-	-	-	-	-	-	-	-
C1	61	41	01	61	41	01	61	41	01	61	41	01
C2	73	53	11	73	53	13	73	53	13	73	53	13

Figure 2.32 Version 1 key-switch codes (1 of 8)

POSITION	US ENGLISH			UK/INT ENGLISH			DANISH			GERMAN		
	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL
C3	64	44	04	64	44	04	64	44	04	64	44	04
C4	66	46	06	66	46	06	66	46	06	66	46	06
C5	67	47	07	67	47	07	67	47	07	67	47	07
C6	68	48	08	68	48	08	68	48	08	68	48	08
C7	6A	4A	0A	6A	4A	0A	6A	4A	0A	6A	4A	0A
C8	6B	4B	0B	6B	4B	0B	6B	4B	0B	6B	4B	0B
C9	6C	4C	0C	6C	4C	0C	6C	4C	0C	6C	4C	0C
C10	3B	3A	-	3B	2B	-	7B	5B	1B	7C	5C	1C
C11	27	22	00	3A	2A	-	7C	5C	1C	7B	5B	1B
C12	60	7E	1E	5D	7D	1D	40	2A	-	23	5E	1E
C13	88	88	98	88	88	98	88	88	98	88	88	98
C50	2D	F4	F5	2D	F4	F5	2D	F4	F5	2D	F4	F5
C51	34	D4	B4	34	D4	B4	34	D4	B4	34	D4	B4
C52	35	D5	B5	35	D5	B5	35	D5	B5	35	D5	B5
C53	36	D6	B6	36	D6	B6	36	D6	B6	36	D6	B6
C54	2A	DA	BA	27	DA	BA	2A	DA	BA	2A	DA	BA
D0	-	-	-	-	-	-	-	-	-	-	-	-
D1	71	51	11	71	51	11	71	51	11	71	51	11
D2	77	57	17	77	57	17	77	57	17	77	57	17
D3	65	45	05	65	45	05	65	45	05	65	45	05
D4	72	52	12	72	52	12	72	52	12	72	52	12
D5	74	54	14	74	54	14	74	54	14	74	54	14
D6	79	59	19	79	59	19	79	59	19	7A	5A	1A
D7	75	55	15	75	55	15	75	55	15	75	55	15
D8	69	49	09	69	49	09	69	49	09	6A	4A	0A

Figure 2.32 Version 1 key-switch codes (2 of 8)

	US ENGLISH			UK/INT ENGLISH			DANISH			GERMAN		
POSITION	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL
D9	6F	4F	0F	6F	4F	0F	6F	4F	0F	6F	4F	0F
D10	70	50	10	70	50	10	70	50	10	70	50	10
D11	5B	7B	1B	40	60	00	7D	5D	1D	7D	5D	1D
D12	5D	7D	1D	5B	7B	1B	5E	7E	1E	2B	2A	-
D13	-	-	-	-	-	-	-	-	-	-	-	-
D50	86	7F	96	86	7F	96	86	7F	96	86	7F	96
D51	37	D7	B7	37	D7	B7	37	D7	B7	37	D7	B7
D52	38	D8	B8	38	D8	B8	38	D8	B8	38	D8	B8
D53	39	D9	B9	39	D9	B9	39	D9	B9	39	D9	B9
D54	2F	DF	BF	2F	DF	BF	2F	DF	BF	2F	DF	BF
E0	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B
E1	31	21	-	31	21	-	31	21	-	31	21	-
E2	32	40	-	32	22	-	32	22	-	32	22	-
E3	33	23	-	33	23	-	33	23	-	33	40	-
E4	34	24	-	34	24	-	34	24	-	34	24	-
E5	35	25	-	35	25	-	35	25	-	35	25	-
E6	36	5E	-	36	26	-	36	26	-	36	26	-
E7	37	26	-	37	27	-	37	2F	-	37	2F	-
E8	38	2A	-	38	28	-	38	28	-	38	28	-
E9	39	28	-	39	29	-	39	29	-	39	29	-
E10	30	29	-	30	5F	1F	30	3D	-	30	3D	-
E11	2D	5F	1F	2D	3D	-	2B	3F	-	7E	3F	-
E12	3D	2B	-	5E	7E	1E	27	60	00	27	60	00
E13	8B	8B	9B	8B	8B	9B	8B	8B	9B	8B	8B	9B
E14	09	09	09	09	09	09	09	09	09	09	09	09

Figure 2.32 Version 1 key-switch codes (3 of 8)

	US ENGLISH		UK/INT ENGLISH		DANISH		GERMAN					
	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED
POSITION	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL
E50	81	81	91	81	81	81	91	81	81	91	81	81
E51	82	82	92	82	82	82	92	82	82	92	82	82
E52	83	83	93	83	83	83	93	83	83	93	83	83
E53	84	84	94	84	84	84	94	84	84	94	84	84
E54	85	85	95	85	85	85	95	85	85	95	85	85
F0	E0	C0	A0	E0	C0	A0	E0	C0	A0	E0	C0	A0
F1	E1	C1	A1	E1	C1	A1	E1	C1	A1	E1	C1	A1
F2	E2	C2	A2	E2	C2	A2	E2	C2	A2	E2	C2	A2
F3	E3	C3	A3	E3	C3	A3	E3	C3	A3	E3	C3	A3
F4	E4	C4	A4	E4	C4	A4	E4	C4	A4	E4	C4	A4
F5	E5	C5	A5	E5	C5	A5	E5	C5	A5	E5	C5	A5
F6	E6	C6	A6	E6	C6	A6	E6	C6	A6	E6	C6	A6
F7	E7	C7	A7	E7	C7	A7	E7	C7	A7	E7	C7	A7
F8	E8	C8	A8	E8	C8	A8	E8	C8	A8	E8	C8	A8
F9	E9	C9	A9	E9	C9	A9	E9	C9	A9	E9	C9	A9
F10	EA	CA	AA	EA	CA	AA	EA	CA	AA	EA	CA	AA
F11	EB	CB	AB	EB	CB	AB	EB	CB	AB	EB	CB	AB
F12	EC	CC	AC	EC	CC	AC	EC	CC	AC	EC	CC	AC
F13	ED	CD	AD	ED	CD	AD	ED	CD	AD	ED	CD	AD
F14	EE	CE	AE	EE	CE	AE	EE	CE	AE	EE	CE	AE
F50	EF	CF	AF	EF	CF	AF	EF	CF	AF	EF	CF	AF
F51	F0	D0	B0	F0	D0	B0	F0	D0	B0	F0	D0	B0
F52	F1	D1	B1	F1	D1	B1	F1	D1	B1	F1	D1	B1
F53	F2	D2	B2	F2	D2	B2	F2	D2	B2	F2	D2	B2
F54	F3	D3	B3	F3	D3	B3	F3	D3	B3	F3	D3	B3

Figure 2.32 Version 1 key-switches (4 of 8)

POSITION	SWEDISH /FINNISH		NOR- WEGIAN		SPANISH		ITALIAN		CONTROL		CONTROL	
	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED
A5	20	20	20	20	20	20	20	20	20	20	20	20
A50	30	30	30	30	30	30	30	30	30	30	30	30
A52	30	30	30	30	30	30	30	30	30	30	30	30
A53	2E	2E	-	2E	2E	-	2E	2E	-	2E	2E	-
B99	-	-	-	-	-	-	-	-	-	-	-	-
B0	3C	3E	-	2C	3E	-	3C	3E	-	3C	3E	-
B1	7A	5A	1A	7A	5A	1A	7A	5A	1A	77	57	17
B2	78	58	18	78	58	18	78	58	18	78	58	18
B3	63	43	03	63	43	03	63	43	03	63	43	03
B4	76	56	16	76	56	16	76	56	16	76	56	16
B5	62	42	02	62	42	02	62	42	02	62	42	02
B6	6E	4E	0E	6E	4E	0E	6E	4E	0E	6E	4E	0E
B7	6D	4D	0D	6D	4D	0D	6D	4D	0D	6D	4D	0D
B8	2C	3B	-	2C	3B	-	2C	3B	-	2C	3B	-
B9	2E	3A	-	2E	3A	-	2E	3A	-	2E	3A	-
B10	2D	5F	1F	2D	5F	1F	2D	5F	1F	2D	5F	1F
B11	-	-	-	-	-	-	-	-	-	-	-	-
B50	2B	DB	BB	2B	DB	BB	2B	DB	BB	2B	DB	BB
B51	31	DC	BC	31	DC	BC	31	DC	BC	31	DC	BC
B52	32	DD	BD	32	DD	BD	32	DD	BD	32	DD	BD
B53	33	DE	BE	33	DE	BE	33	DE	BE	33	DE	BE
B54	88	88	98	88	88	98	88	88	98	88	88	98
C0	-	-	-	-	-	-	-	-	-	-	-	-
C1	61	41	01	61	41	01	61	41	01	61	41	01
C2	73	53	13	73	53	13	73	53	13	73	53	13

Figure 2.32 Version 1 key-switch codes (5 of 8)

POSITION	SWEDISH /FINNISH		NOR- WEGIAN		SPANISH		ITALIAN		CONTROL		CONTROL	
	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED
C3	64	44	04	64	44	04	64	44	04	64	44	04
C4	66	46	06	66	46	06	66	46	06	66	46	06
C5	67	47	07	67	47	07	67	47	07	67	47	07
C6	68	48	08	68	48	08	68	48	08	68	48	08
C7	6A	4A	0A	6A	4A	0A	6A	4A	0A	6A	4A	0A
C8	6B	4B	0B	6B	4B	0B	6B	4B	0B	6B	4B	0B
C9	6C	4C	0C	6C	4C	0C	6C	4C	0C	6C	4C	0C
C10	7C	5C	1C	7C	5C	1C	7C	5C	1C	7B	5C	1C
C11	7B	5B	1B	7B	5B	1B	27	60	00	60	5B	1B
C12	40	2A	—	40	2A	—	7B	7D	1D	7C	40	00
C13	88	88	98	88	88	98	88	88	98	88	88	98
C50	2D	F4	F5	2D	F4	F5	2D	F4	F5	2D	F4	F5
C51	34	D4	B4	34	D4	B4	34	D4	B4	34	D4	B4
C52	35	D5	B5	35	D5	B5	35	D5	B5	35	D5	B5
C53	36	D6	B6	36	D6	B6	36	D6	B6	36	D6	B6
C54	2A	DA	BA	2A	DA	BA	2A	DA	BA	2A	DA	BA
D0	—	—	—	—	—	—	—	—	—	—	—	—
D1	71	51	11	71	51	11	71	51	11	71	51	11
D2	77	57	17	77	57	17	77	57	17	7A	5A	1A
D3	65	45	05	65	45	05	65	45	05	65	45	05
D4	72	52	12	72	52	12	72	52	12	72	52	12
D5	74	54	14	74	54	14	74	54	14	74	54	14
D6	79	59	19	79	59	19	79	59	19	79	59	19
D7	75	55	15	75	55	15	75	55	15	75	55	15
D8	69	49	09	69	49	09	69	49	09	69	49	09

Figure 2.32 Version 1 key-switch codes (6 of 8)

	SWEDISH /FINNISH			NOR-WEGIAN			SPANISH			ITALIAN		
	UNSHIFTED	CONTROL	UNSHIFTED	UNSHIFTED	CONTROL	UNSHIFTED	UNSHIFTED	CONTROL	UNSHIFTED	UNSHIFTED	CONTROL	UNSHIFTED
POSITION												
D9	6F	4F	0F	6F	4F	0F	6F	4F	0F	6F	4F	0F
D10	70	50	10	70	50	10	70	50	10	70	50	10
D11	7D	5D	1D	7D	5D	1D	5E	7E	1E	5D	7D	1D
D12	5E	7E	1E	5E	7E	1E	2B	2A	-	2B	2A	-
D13	-	-	-	-	-	-	-	-	-	-	-	-
D50	86	7F	96	86	7F	96	86	7F	96	86	7F	96
D51	37	D7	B7	37	D7	B7	37	D7	B7	37	D7	B7
D52	38	D8	B8	38	D8	B8	38	D8	B8	38	D8	B8
D53	39	D9	B9	39	D9	B9	39	D9	B9	39	D9	B9
D54	2F	DF	BF	2F	DF	BF	2F	DF	BF	2F	DF	BF
E0	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B
E1	31	21	-	31	21	-	31	21	-	31	21	-
E2	32	22	-	32	22	-	32	22	-	32	22	-
E3	33	23	-	33	23	-	33	23	-	33	23	-
E4	34	24	-	34	24	-	34	24	-	34	24	-
E5	35	25	-	35	25	-	35	25	-	35	25	-
E6	36	26	-	36	26	-	36	26	-	36	26	-
E7	37	2F	-	37	2F	-	37	2F	-	37	2F	-
E8	38	28	-	38	28	-	38	28	-	38	28	-
E9	39	29	-	39	29	-	39	29	-	39	29	-
E10	30	3D	-	30	3D	-	30	3D	-	30	3D	-
E11	2B	3F	-	2B	3F	-	40	3F	-	27	3F	-
E12	27	60	00	27	60	00	5B	5D	1B	7E	5E	1E
E13	8B	8B	9B	8B	8B	9B	8B	8B	9B	8B	8B	9B
E14	09	09	09	09	09	09	09	09	09	09	09	09

Figure 2.32 Version 1 key-switch codes (7 of 8)

SWEDISH /FINNISH	NOR- WEGIAN		SPANISH		ITALIAN							
	UNSHIFTED POSITION	SHIFTED CONTROL	UNSHIFTED	SHIFTED1	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED
E50	81	81	91	81	81	91	81	81	91	81	81	91
E51	82	82	92	82	82	92	82	82	92	82	82	92
E52	83	83	93	83	83	93	83	83	93	83	83	93
E53	84	84	94	84	84	94	84	84	94	84	84	94
E54	85	85	95	85	85	95	85	85	95	85	85	95
F0	E0	C0	A0	E0	C0	A0	E0	C0	A0	E0	C0	A0
F1	E1	C1	A1	E1	C1	A1	E1	C1	A1	E1	C1	A1
F2	E2	C2	A2	E2	C2	A2	E2	C2	A2	E2	C2	A2
F3	E3	C3	A3	E3	C3	A3	E3	C3	A3	E3	C3	A3
F4	E4	C4	A4	E4	C4	A4	E4	C4	A4	E4	C4	A4
F5	E5	C5	A5	E5	C5	A5	E5	C5	A5	E5	C5	A5
F6	E6	C6	A6	E6	C6	A6	E6	C6	A6	E6	C6	A6
F7	E7	C7	A7	E7	C7	A7	E7	C7	A7	E7	C7	A7
F8	E8	C8	A8	E8	C8	A8	E8	C8	A8	E8	C8	A8
F9	E9	C9	A9	E9	C9	A9	E9	C9	A9	E9	C9	A9
F10	EA	CA	AA	EA	CA	AA	EA	CA	AA	EA	CA	AA
F11	EB	CB	AB	EB	CB	AB	EB	CB	AB	EB	CB	AB
F12	EC	CC	AC	EC	CC	AC	EC	CC	AC	EC	CC	AC
F13	ED	CD	AD	ED	CD	AD	ED	CD	AD	ED	CD	AD
F14	EE	CE	AE	EE	CE	AE	EE	CE	AE	EE	CE	AE
F50	EF	CF	AF	EF	CF	AF	EF	CF	AF	EF	CF	AF
F51	F0	D0	B0	F0	D0	B0	F0	D0	B0	F0	D0	B0
F52	F1	D1	B1	F1	D1	B1	F1	D1	B1	F1	D1	B1
F53	F2	D2	B2	F2	D2	B2	F2	D2	B2	F2	D2	B2
F54	F3	D3	B3	F3	D3	B3	F3	D3	B3	F3	D3	B3

Figure 2.32 Version 1 key-switch codes (8 of 8)

	SWISS-GERMAN		SWISS-FRENCH		FRENCH		CANAD./AUSTRAL.					
	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED
POSITION	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL	CONTROL
A5	20	20	20	20	20	20	20	20	20	20	20	20
A50	30	30	30	30	30	30	30	30	30	30	30	30
A52	30	30	30	30	30	30	30	30	30	30	30	30
A53	2E	2E	-	2E	2E	-	2E	2E	-	2E	2E	-
B99	-	-	-	-	-	-	-	-	-	-	-	-
B0	3C	3E	-	3C	3E	-	3C	3E	-	5C	7C	1C
B1	79	59	19	79	59	19	77	57	17	7A	5A	1A
B2	78	58	18	78	58	18	78	58	18	78	58	18
B3	63	43	03	63	43	03	63	43	03	63	43	03
B4	76	56	16	76	56	16	76	56	16	76	56	16
B5	62	42	02	62	42	02	62	42	02	62	42	02
B6	6E	4E	0E	6E	4E	0E	6E	4E	0E	6E	4E	0E
B7	6D	4D	0D	6D	4D	0D	2C	3F	-	6D	4D	0D
B8	2C	3B	-	2C	3B	-	3B	2E	-	2C	3C	-
B9	2E	3A	-	2E	3A	-	3A	2F	-	2E	3E	-
B10	2D	5F	1F	2D	5F	1F	3D	2B	-	2F	3F	-
B11	-	-	-	-	-	-	-	-	-	-	-	-
B50	2B	DB	BB	2B	DB	BB	2B	DB	BB	2B	DB	BB
B51	31	DC	BC	31	DC	BC	31	DC	BC	31	DC	BC
B52	32	DD	BD	32	DD	BD	32	DD	BD	32	DD	BD
B53	33	DE	BE	33	DE	BE	33	DE	BE	33	DE	BE
B54	88	88	98	88	88	98	88	88	98	88	88	98
C0	-	-	-	-	-	-	-	-	-	-	-	-
C1	61	41	01	61	41	01	71	51	11	61	41	01
C2	73	53	13	73	53	13	73	53	13	73	53	13

Figure 2.33 Version 2 key-switch codes (1 of 8)

	SWISS-GERMAN			SWISS-FRENCH			FRENCH			CANAD./AUSTRAL.		
	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL
POSITION												
C3	64	44	04	64	44	04	64	44	04	64	44	04
C4	66	46	06	66	46	06	66	46	06	66	46	06
C5	67	47	07	67	47	07	67	47	07	67	47	07
C6	68	48	08	68	48	08	68	48	08	68	48	08
C7	6A	4A	0A	6A	4A	0A	6A	4A	0A	6A	4A	0A
C8	6B	4B	0B	6B	4B	0B	6B	4B	0B	6B	4B	0B
C9	6C	4C	0C	6C	4C	0C	6C	4C	0C	6C	4C	0C
C10	7C	5C	1C	5C	7C	1C	6D	4D	0D	3B	3A	-
C11	7B	5B	1B	5B	7B	1B	7C	25	1C	27	22	20
C12	24	23	-	24	23	-	60	23	1D	60	7B	1B
C13	88	88	98	88	88	98	88	88	98	88	88	98
C50	2D	F4	F5	2D	F4	F5	2D	F4	F5	2D	F4	F5
C51	34	D4	B4	34	D4	B4	34	D4	B4	34	D4	B4
C52	35	D5	B5	35	D5	B5	35	D5	B5	35	D5	B5
C53	36	D6	B6	36	D6	B6	36	D6	B6	36	D6	B6
C54	2A	DA	BA	2A	DA	BA	2A	DA	BA	2A	DA	BA
D0	-	-	-	-	-	-	-	-	-	-	-	-
D1	71	51	11	71	51	11	61	41	01	71	51	11
D2	77	57	17	77	57	17	7A	5A	1A	77	57	17
D3	65	45	05	65	45	05	65	45	05	65	45	05
D4	72	52	12	72	52	12	72	52	12	72	52	12
D5	74	54	14	74	54	14	74	54	14	74	54	14
D6	7A	5A	1A	7A	5A	1A	79	59	19	79	59	19
D7	75	55	15	75	55	15	75	55	15	75	55	15
D8	69	49	09	69	49	09	69	49	09	69	49	09

Figure 2.33 Version 2 key-switch codes (2 of 8)

	SWISS-GERMAN			SWISS-FRENCH			FRENCH			CANAD./AUSTRAL.		
	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL
POSITION												
D9	6F	4F	0F	6F	4F	0F	6F	4F	0F	6F	4F	0F
D10	70	50	40	70	50	10	70	50	10	70	50	10
D11	7D	5D	1D	5D	7D	1D	5E	7E	1E	5D	5B	1D
D12	7E	27	1E	7E	27	1E	24	2A	-	5E	7E	1E
D13	-	-	-	-	-	-	-	-	-	-	-	-
D50	86	7F	96	86	7F	96	86	7F	96	86	7F	96
D51	37	D7	B7	37	D7	B7	37	D7	B7	37	D7	B7
D52	38	D8	B8	38	D8	B8	38	D8	B8	38	D8	B8
D53	39	D9	B9	39	D9	B9	39	D9	B9	39	D9	B9
D54	2F	DF	BF	2F	DF	BF	2F	DF	BF	2F	DF	BF
E0	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B
E1	31	2B	-	31	2B	-	26	31	-	31	21	-
E2	32	22	-	32	22	-	7B	32	-	32	40	-
E3	33	2A	-	33	2A	-	22	33	-	33	23	-
E4	34	7D	-	34	7D	-	27	34	-	34	24	-
E5	35	25	-	35	25	-	28	35	-	35	25	-
E6	36	26	-	36	26	-	50	36	-	36	7D	-
E7	37	2F	-	37	2F	-	7D	37	-	37	26	-
E8	38	28	-	38	28	-	21	38	-	38	2A	-
E9	39	29	-	39	29	-	5C	39	-	39	28	-
E10	30	3D	-	30	3D	-	40	30	00	30	29	-
E11	21	3F	-	21	3F	-	29	5B	1B	2D	5F	1F
E12	5E	60	00	5E	60	00	2D	5F	1F	3D	2B	-
E13	8B	8B	9B	8B	8B	9B	8B	8B	9B	8B	8B	9B
E14	09	09	09	09	09	09	09	09	09	09	09	09

Figure 2.33 Version 2 key-switch codes (3 of 8)

POSITION	SWISS-GERMAN		SWISS-FRENCH		FRENCH		CANAD./AUSTRAL.		CONTROL		CONTROL	
	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED
E50	81	81	91	81	81	91	81	81	91	81	81	91
E51	82	82	92	82	82	92	82	82	92	82	82	92
E52	83	83	93	83	83	93	83	83	93	83	83	93
E53	84	84	94	84	84	94	84	84	94	84	84	94
E54	85	85	95	85	85	95	85	85	95	85	85	95
F0	E0	C0	A0	E0	C0	A0	E0	C0	A0	E0	C0	A0
F1	E1	C1	A1	E1	C1	A1	E1	C1	A1	E1	C1	A1
F2	E2	C2	A2	E2	C2	A2	E2	C2	A2	E2	C2	A2
F3	E3	C3	A3	E3	C3	A3	E3	C3	A3	E3	C3	A3
F4	E4	C4	A4	E4	C4	A4	E4	C4	A4	E4	C4	A4
F5	E5	C5	A5	E5	C5	A5	E5	C5	A5	E5	C5	A5
F6	E6	C6	A6	E6	C6	A6	E6	C6	A6	E6	C6	A6
F7	E7	C7	A7	E7	C7	A7	E7	C7	A7	E7	C7	A7
F8	E8	C8	A8	E8	C8	A8	E8	C8	A8	E8	C8	A8
F9	E9	C9	A9	E9	C9	A9	E9	C9	A9	E9	C9	A9
F10	EA	CA	AA	EA	CA	AA	EA	CA	AA	EA	CA	AA
F11	EB	CB	AB	EB	CB	AB	EB	CB	AB	EB	CB	AB
F12	EC	CC	AC	EC	CC	AC	EC	CC	AC	EC	CC	AC
F13	ED	CD	AD	ED	CD	AD	ED	CD	AD	ED	CD	AD
F14	EE	CE	AE	EE	CE	AE	EE	CE	AE	EE	CE	AE
F50	EF	CF	AF	EF	CF	AF	EF	CF	AF	EF	CF	AF
F51	F0	D0	B0	F0	D0	B0	F0	D0	B0	F0	D0	B0
F52	F1	D1	B1	F1	D1	B1	F1	D1	B1	F1	D1	B1
F53	F2	D2	B2	F2	D2	B2	F2	D2	B2	F2	D2	B2
F54	F3	D3	B3	F3	D3	B3	F3	D3	B3	F3	D3	B3

Figure 2.33 Version 2 key-switch codes (4 of 8)

CANAD. BILING.	SOUTH AFRICAN		PORTU- GUESE		JUGO- SLAVIAN		UNSHIFTED		SHIFTED		CONTROL	
	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED	UNSHIFTED	SHIFTED
POSITION												
A5	20	20	20	20	20	20	20	20	20	20	20	20
A50	30	30	30	30	30	30	30	30	30	30	30	30
A52	30	30	30	30	30	30	30	30	30	30	30	30
A53	2E	2E	-	2E	2E	-	2E	2E	-	2E	2E	-
B99	-	-	-	-	-	-	-	-	-	-	-	-
B0	5C	7C	1C	2C	3E	-	3C	3E	-	3C	3E	-
B1	7A	5A	1A	7A	5A	1A	77	57	17	79	59	19
B2	78	58	18	78	58	18	78	58	18	78	58	18
B3	63	43	03	63	43	03	63	43	03	63	43	03
B4	76	56	16	76	56	16	76	56	16	76	56	16
B5	62	42	02	62	42	02	62	42	02	62	42	02
B6	6E	4E	0E	6E	4E	0E	6E	4E	0E	6E	4E	0E
B7	6D	4D	0D	6D	4D	0D	6D	4D	0D	6D	4D	0D
B8	2C	3C	-	2C	3B	-	2C	3B	-	2C	3B	-
B9	2E	3E	-	2E	3A	-	2E	3A	-	2E	3A	-
B10	2F	3F	-	2D	5F	1F	2D	5F	1F	20	5F	1F
B11	-	-	-	-	-	-	-	-	-	-	-	-
B50	2B	DB	BB	2B	DB	BB	2B	DB	BB	2B	DB	BB
B51	31	DC	BC	31	DC	BC	31	DC	BC	31	DC	BC
B52	32	DD	BD	32	DD	BD	32	DD	BD	32	DD	BD
B53	33	DE	BE	33	DE	BE	33	DE	BE	33	DE	BE
B54	88	88	98	88	88	98	88	88	98	88	88	98
C0	-	-	-	-	-	-	-	-	-	-	-	-
C1	61	41	01	61	41	01	71	51	11	61	41	01
C2	73	53	13	73	53	13	73	53	13	73	53	13

Figure 2.33 Version 2 key-switch codes (5 of 8)

CANAD. BILING.	SOUTH AFRICAN			PORTU- GUESE			JUGO- SLAVIAN			CONTROL		
	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL
POSITION												
C3	64	44	04	64	44	04	64	44	04	64	44	04
C4	66	46	06	66	46	06	66	46	06	66	46	06
C5	67	47	07	67	47	07	67	47	07	67	47	07
C6	68	48	08	68	48	08	68	48	08	68	48	08
C7	6A	4A	0A	6A	4A	0A	6A	4A	0A	6A	4A	0A
C8	6B	4B	0B	6B	4B	0B	6B	4B	0B	6B	4B	0B
C9	6C	4C	0C	6C	4C	0C	6C	4C	0C	6C	4C	0C
C10	3B	3A	-	7C	5C	1C	7D	5D	1D	7C	5C	1C
C11	27	22	00	7B	5B	1B	7C	5C	1C	7B	5B	1B
C12	60	7B	1B	21	41	-	27	60	00	7E	5E	1E
C13	88	88	98	88	88	98	88	88	98	88	88	98
C50	2D	F4	F5	2D	F4	F5	2D	F4	F5	2D	F4	F5
C51	34	D4	B4	34	D4	B4	34	D4	B4	34	D4	B4
C52	35	D5	B5	35	D5	B5	35	D5	B5	35	D5	B5
C53	36	D6	B6	36	D6	B6	36	D6	B6	36	D6	B6
C54	2A	DA	BA	2A	DA	BA	2A	DA	BA	2A	DA	BA
D0	-	-	-	-	-	-	-	-	-	-	-	-
D1	71	51	11	71	51	11	61	41	01	71	51	11
D2	77	57	17	77	57	17	7A	5A	1A	77	57	17
D3	65	45	05	65	45	05	65	45	05	65	45	05
D4	72	52	12	72	52	12	72	52	12	72	52	02
D5	74	54	14	74	54	14	74	54	14	74	54	14
D6	79	59	19	79	59	19	79	59	19	7A	5A	1A
D7	75	55	15	75	55	15	75	55	15	75	55	15
D8	69	49	09	69	49	09	69	49	09	69	49	09

Figure 2.33 Version 2 key-switch codes (6 of 8)

POSITION	CANAD. BILING.			SOUTH AFRICAN			PORTU-GUESE			JUGO-SLAVIAN		
	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL
D9	6F	4F	0F	6F	4F	0F	6F	4F	0F	6F	4F	0F
D10	70	50	10	70	50	10	70	50	10	70	50	10
D11	5D	5B	1D	7D	5D	1D	7B	5B	1B	7D	5D	1D
D12	5E	7E	1E	5E	7E	1E	2B	2A	-	2B	27	-
D13	-	-	-	-	-	-	-	-	-	-	-	-
D50	86	7F	96	86	7F	96	86	7F	96	86	7F	96
D51	37	D7	B7	37	D7	B7	37	D7	B7	37	D7	B7
D52	38	D8	B8	38	D8	B8	38	D8	B8	38	D8	B8
D53	39	D9	B9	39	D9	B9	39	D9	B9	39	D9	B9
D54	2F	DF	BF	2F	DF	BF	2F	DF	BF	2F	DF	BF
E0	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B	1B
E1	31	21	-	31	2B	-	31	21	-	31	21	-
E2	32	40	-	32	22	-	32	22	-	32	22	-
E3	33	23	-	33	23	-	33	23	-	33	23	-
E4	34	24	-	34	40	-	34	24	-	34	24	-
E5	35	25	-	35	25	-	35	25	-	35	25	-
E6	36	7D	-	36	26	-	36	26	-	36	26	-
E7	37	26	-	37	2A	-	37	2F	-	37	2F	-
E8	38	2A	-	38	28	-	38	28	-	38	28	-
E9	39	28	-	39	29	-	39	29	-	39	29	-
E10	30	29	-	30	3D	-	30	3D	-	30	3D	-
E11	2D	5F	1F	2F	3F	-	40	3F	-	27	3F	-
E12	3D	2B	-	27	60	00	5E	7E	1E	40	60	00
E13	8B	8B	9B	8B	8B	9B	8B	8B	9B	8B	8B	9B
E14	09	09	09	09	09	09	09	09	09	09	09	09

Figure 2.33 Version 2 key-switch codes (7 of 8)

CANAD. BILING.	SOUTH AFRICAN			PORTU- GUESE			JUGO- SLAVIAN			CONTROL		
	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL	UNSHIFTED	SHIFTED	CONTROL
POSITION												
E50	81	81	91	81	81	91	81	81	91	81	81	91
E51	82	82	92	82	82	92	82	82	92	82	82	92
E52	83	83	93	83	83	93	83	83	93	83	83	93
E53	84	84	94	84	84	94	84	84	94	84	84	94
E54	85	85	95	85	85	95	85	85	95	85	85	95
F0	E0	C0	A0	E0	C0	A0	E0	C0	A0	E0	C0	A0
F1	E1	C1	A1	E1	C1	A1	E1	C1	A1	E1	C1	A1
F2	E2	C2	A2	E2	C2	A2	E2	C2	A2	E2	C2	A2
F3	E3	C3	A3	E3	C3	A3	E3	C3	A3	E3	C3	A3
F4	E4	C4	A4	E4	C4	A4	E4	C4	A4	E4	C4	A4
F5	E5	C5	A5	E5	C5	A5	E5	C5	A5	E5	C5	A5
F6	E6	C6	A6	E6	C6	A6	E6	C6	A6	E6	C6	A6
F7	E7	C7	A7	E7	C7	A7	E7	C7	A7	E7	C7	A7
F8	E8	C8	A8	E8	C8	A8	E8	C8	A8	E8	C8	A8
F9	E9	C9	A9	E9	C9	A9	E9	C9	A9	E9	C9	A9
F10	EA	CA	AA	EA	CA	AA	EA	CA	AA	EA	CA	AA
F11	EB	CB	AB	EB	CB	AB	EB	CB	AB	EB	CB	AB
F12	EC	CC	AC	EC	CC	AC	EC	CC	AC	EC	CC	AC
F13	ED	CD	AD	ED	CD	AD	ED	CD	AD	ED	CD	AD
F14	EE	CE	AE	EE	CE	AE	EE	CE	AE	EE	CE	AE
F50	EF	CF	AF	EF	CF	AF	EF	CF	AF	EF	CF	AF
F51	F0	D0	B0	F0	D0	B0	F0	D0	B0	F0	D0	B0
F52	F1	D1	B1	F1	D1	B1	F1	D1	B1	F1	D1	B1
F53	F2	D2	B2	F2	D2	B2	F2	D2	B2	F2	D2	B2
F54	F3	D3	B3	F3	D3	B3	F3	D3	B3	F3	D3	B3

Figure 2.33 Version 2 key-switch codes (8 of 8)

MONOCHROME CRT

With the monochrome 12-inch CRT, text with 25 horizontal lines across the screen can be displayed. Each line has 80 character positions. It also has graphics capability to draw complex pictures. This graphics capability makes the screen all points addressable (APA). (640 points horizontally and 400 points vertically.) The tube uses a green-colored phosphor for a green display on a dark background. An anti-reflection surface is provided. The screen is refreshed at 56.2Hz, the horizontal frequency is 23.8k Hz. The CRT requires a supply voltage of $+12\text{Vdc} \pm 5\%$ and a maximum current of 1.5A. Brightness and contrast controls are conveniently located on the front of the unit, allowing easy adjustment for different user's preference and room lighting.

The dimensions of the character field on the CRT screen are shown in Figure 2.34.

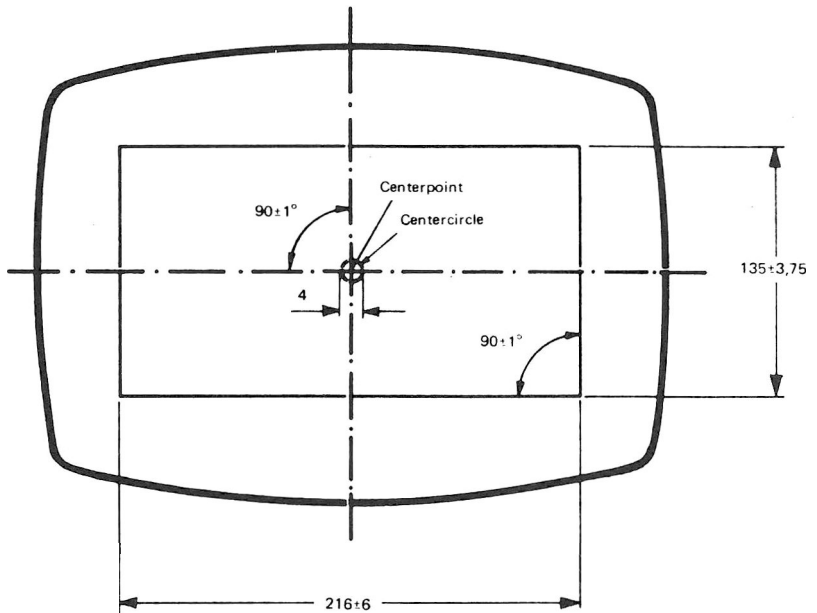


Figure 2.34 CRT character field

The signals and pin assignments between the controller board and the CRT are shown in Figure 2.35, and a timing diagram in Figure 2.36.

Controller Board (TB 101)	Signal	CRT Board (J/P 1)
Pin 1	HSYNC	Pin 5
2	+12V	4
3	VIDEO	3
4	VSYNC	2
5	LGRD	1

Figure 2.35 CRT signals

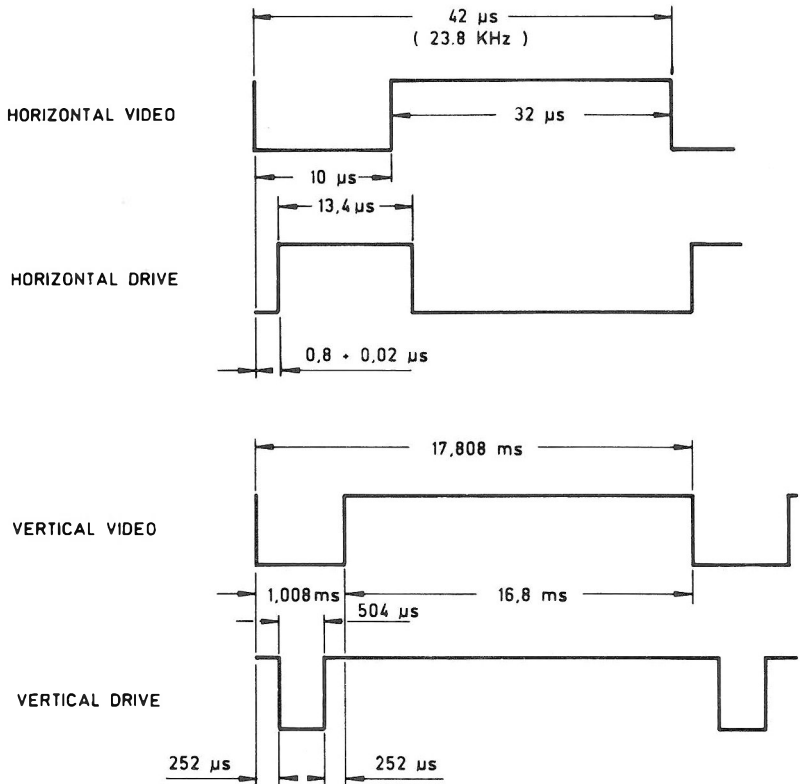


Figure 2.36 CRT (monochrome) timing

COLOR CRT

The 12-inch color CRT displays 25 horizontal lines with 80 characters per line. In the graphics mode, the CRT is all points addressable with 640 pixels horizontally and 400 pixels vertically. Characters are in a 6 x 9 matrix on an 8 x 16 box. The CRT can display eight colors: black, white, red, green, blue, yellow, magenta, and cyan. The selection of color is controlled by the software. Screen refresh is at 56.06 Hz, the horizontal frequency is 24.27 KHz.

The CRT requires a supply voltage of 107Vac \pm 2V with a maximum current of 0.45A. The CRT monitor accepts separate direct drive inputs at TTL level to video blue, video green, video red, horizontal drive, and vertical drive circuits.

Technical details of the color CRT are shown in Figure 2.37.

Deflection angle	90 Degrees
CRT surface	Non-glare, anti-reflection, high-contrast
Phospor	P22 (red, green, blue)
Dot trio pitch	0.31 mm
Number of colors	8

Figure 2.37 CRT Technical details

RS-232-C PERIPHERAL ADAPTERS

The adapter consists of a small printed circuit board, containing a 2651 programmable communications interface and support logic, a connecting cable and plug, and a housing that allows direct connection to the I/O bus of the NCR DECISION MATE V.

Three versions of the adapter are available, K211, K212, and K213:

- K211 for the connection of communications devices such as modems and line couplers
- K212 for the connection of serial printers and terminals
- K213 for the connection of serial plotters

Functionally, the adapters are similar: the differences being in the connecting cables and plugs, and the straps on the printed circuit board. The 2651 programmable communications integrated circuit supports many serial data communication disciplines in full or half-duplex mode. The interface is capable of transmission rates from 50 to 19200 baud.

The pin assignments for the I/O bus connector are shown in Figure 2.38. These signals match those of the I/O bus and are defined in the I/O bus description in this manual. The differences in the cables are shown in Figure 2.39.

a		c
+5 V	1	+5 V
IRQ	2	+12 V
RESET/ IOW/	3	
	4	IOR/
	5	
BD1	6	BD0
BD3	7	BD2
BD5	8	BD4
BD7	9	BD6
	10	
	11	
	12	
	13	
	14	
	15	
LGRD	16	
	17	
	18	
	19	
	20	
	21	
	22	
	23	
	24	
BA1	25	BA2
	26	BA0
	27	
IFSEL1/	28	IFSEL0/
	29	
	30	
	31	
LGRD	32	LGRD

Figure 2.38 RS-232C pin assignments

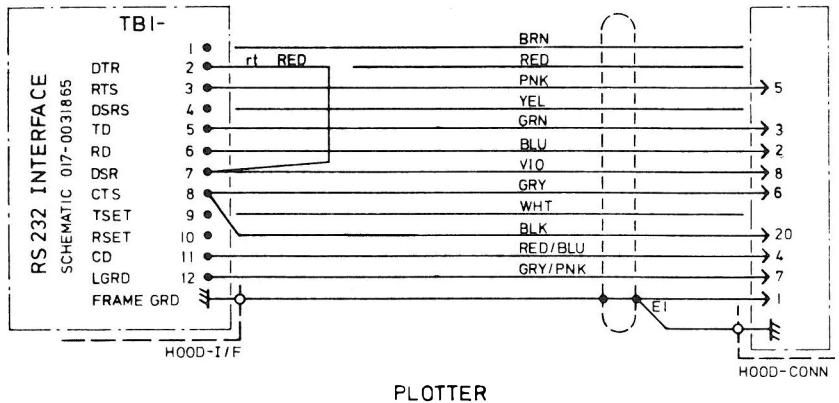
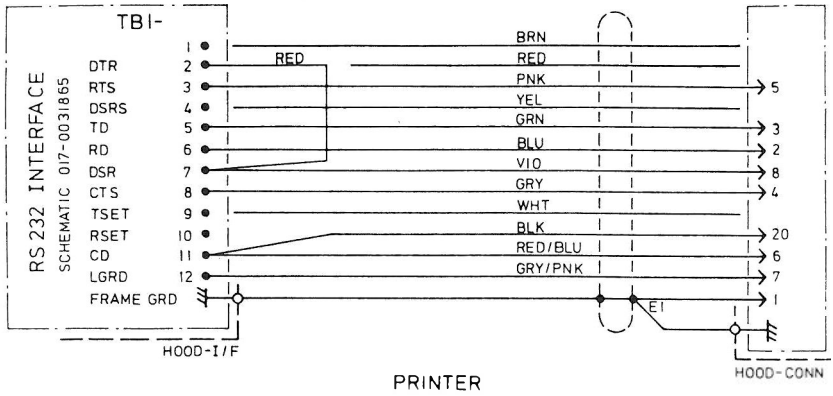
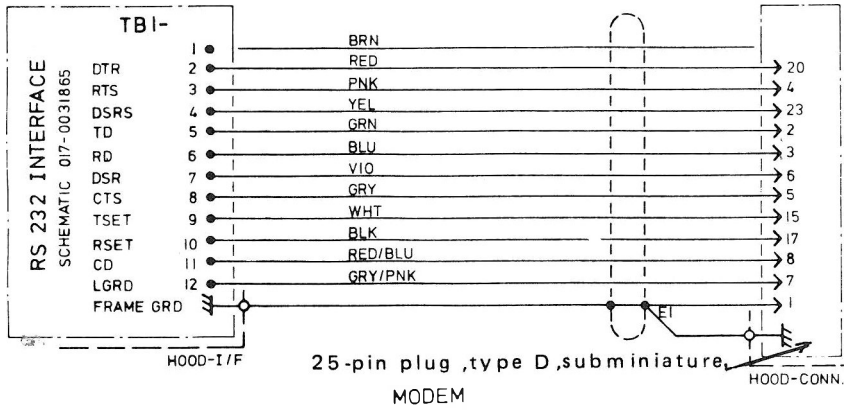


Figure 2.39 RS-232C cables (1 of 2)

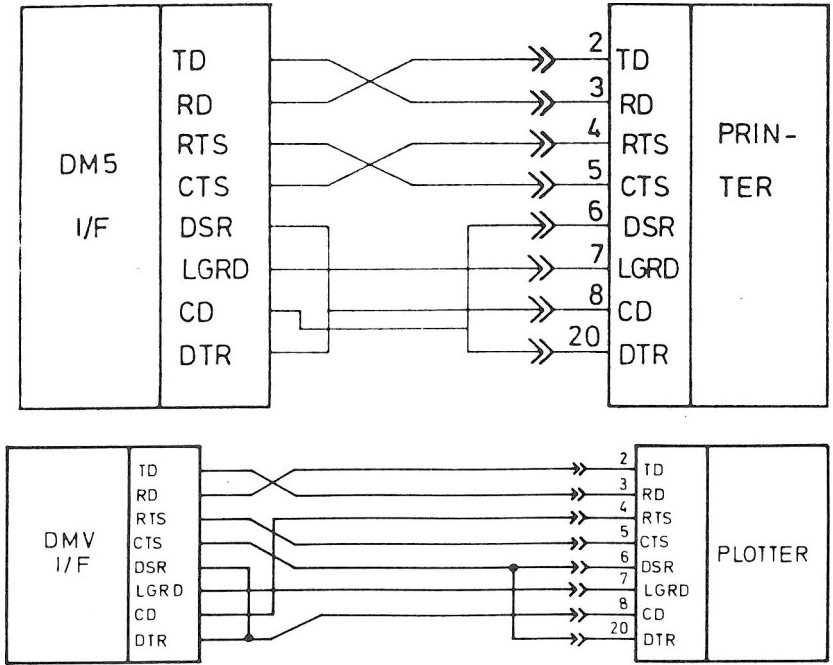


Figure 2.39 RS-232C cables (2 of 2)

The signals to and from the adapter are standard RS-232-C signals; Figure 2.40 defines the required voltage levels for correct operation; Figure 2.41 defines the standard RS-232-C interface signals.

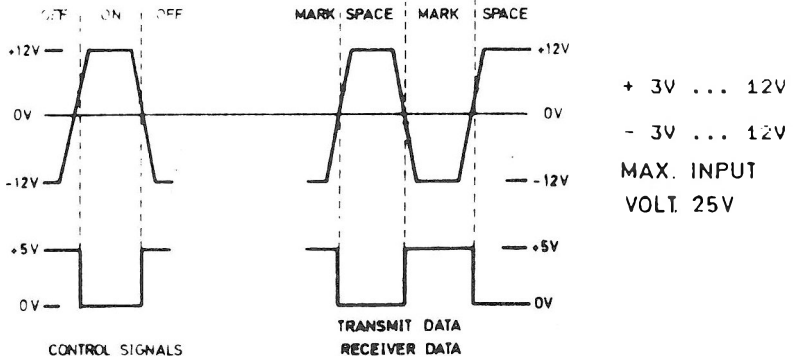


Figure 2.40 RS-232C voltage levels

TRANSMIT DATA (TD)	Serial (5-8 Bit) data transmitted by the computer. Logic '1' is low, logic '0' is high. Between character transmissions this signal will be low.
RECEIVE DATA (RD)	Serial (5-8 Bit) data received by the terminal. Logic '1' is low, logic '0' is high. Between characters this signal must be low.
CLEAR TO SEND (CTS)	Input to the computer, must be high for data transmission.
REQUEST TO SEND (RTS)	This output indicates that the terminal wishes to transmit. In the no modem mode this signal is always high. In the modem and half duplex modes, this signal is low, when there is no data for the terminal to transmit, and high when the terminal wishes to transmit.
DATA SET READY (DSR)	Input to the computer, must be high for data transmission in the modem mode. This signal is ignored in the no modem mode.
DATA TERMINAL READY (DTR)	Output indicating that the computer is ready to receive data.
DATA SIGNAL RATE SELECT (DSRS)	Controls a feature of some modems which allows them to transmit at two different signalling rates. On = high signal rate. This signal (ON state = 12V) can also be used for jumpers in modemless operations.
TRANSMITTER SIGNAL ELEMENT TIMING (TSET)	External clock for transmitter
RECEIVER SIGNAL ELEMENT TIMING (RSET)	External clock for receiver
CARRIER DETECT (CD)	Input from the modem. This signal indicates that the (CD) modem is correctly receiving the data carrier signal from the remote modem. This line must be high for receiving data.

Figure 2.41 RS-232-C signals

SOFTWARE STRAPPING

The following asynchronous communications features are selected by the operating system software:

- 5, 6, 7, or 8-bit data characters
 - 1, 1 1/2, or 2 stop bits
 - Odd, even, or no parity
 - Baud rates at —

50	1800
75	2000
110	2400
134.5	3600
150	4800
300	7200
600	9600
1200	19200
-
- Line break signal generation and detection
 - False start bit detection
 - Parity overrun, and framing error detection
 - Internal diagnostics

HARDWARE STRAPPING

Hardware straps are used to indicate K211, K212, or K213, see Figure 2.42.

	J1	J2
K211 (Modem)	Closed	Open
K212 (Printer)	Open	Closed
K213 (Plotter)	Open	Closed

Figure 2.42 RS-232-C Hardware strapping

CENTRONICS PERIPHERAL ADAPTER

The adapter (K210) provides an interface between the computer and Centronics-compatible (parallel) printers.

The adapter consists of a small printed circuit board, containing an 8255 programmable communications interface and support logic, a connecting cable and plug, and a housing that allows direct connection to the I/O bus of the NCR DECISION MATE V.

The wiring of the connecting cable and plug is shown in Figure 2.43, and the pin assignments for the I/O bus connector are shown in Figure 2.44. The signals match those of the I/O bus and are defined in the I/O bus description in this manual.

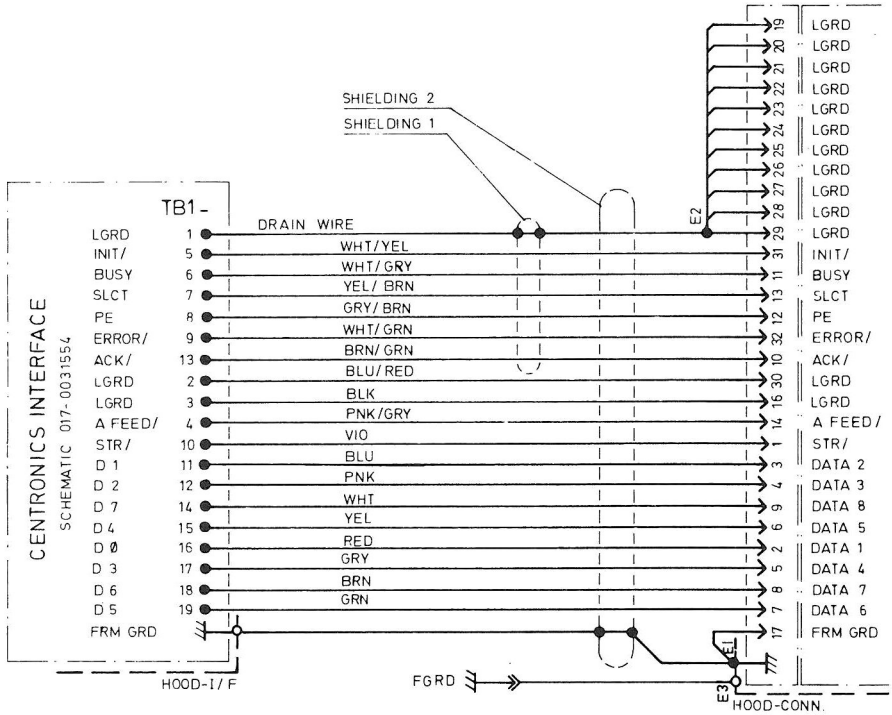


Figure 2.43 Centronics cable

a		c
+5 V	1	+5 V
IRQ	2	
RESET/	3	
IOW/	4	IOR/
	5	
BD1	6	BD0
BD3	7	BD2
BD5	8	BD4
BD7	9	BD6
	10	
	11	
	12	
	13	
	14	
	15	CLK1
LGRD	16	
	17	
	18	
	19	
	20	
	21	
	22	
	23	
	24	
	25	
BA1	26	BA0
	27	
	28	IFSEL0/
	29	
	30	
	31	
LGRD	32	LGRD

Figure 2.44 Centronics pin assignments

The signals to and from the adapter are standard TTL level. The rise and fall time of the signals must be less than $0.2\mu\text{sec}$. The interface timing is shown in Figure 2.45.

The Centronics interface signals are defined in Figure 2.46.

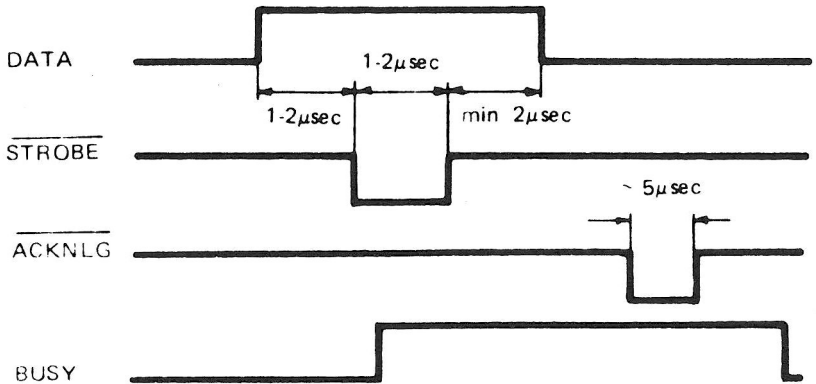


Figure 2.45 Centronics interface timing

STROBE/	Pulse is used to clock read data in. The pulse width must be more than 1 μsec at receiving printer. Read in of data is performed at the "low" level of this signal.
ACKNLG/	Indicates that data has been received by the printer. New data cannot be received before ACKNLG is reset. Pulse width 2.5 μsec to 10 μsec.
BUSY	When "HIGH" the printer indicates that it cannot receive data. The signal becomes "HIGH" in the following cases: <ul style="list-style-type: none"> - During data entry (depending on printer) - During printing operation - In off-line state - During printer error status - During line feed
DATA 1-8	Signals represent information of parallel data "HIGH" = log 1; "LOW" = log 0.
PE	"HIGH" signal indicates that the printer is out of paper.
SLCT	(Input) Printer is in selected mode (option).
AUTO FEED/	(Output) Auto line feed (option).
INIT/	(Output) Printer reset (option).
ERROR/	(Input) Indicates printer error (option) for example at paper end, off-line, safety switch open, failure status.
TEST 1-4	Optional loop test inputs.

Figure 2.46 Centronics signals

MEMORY EXPANSION MODULES

The memory expansion modules (K200, K202, and K208) are contained in small housings that fit directly into the I/O bus of the NCR DECISION MATE V. Three different modules are available; however, only one at a time may be installed; and this must be fitted to slot 1.

The modules contain semiconductor memory ICs (dynamic RAM) and a bus interface connector. They are available in the following sizes:

- 64 K bytes — to provide a system capacity of 128 K bytes
- 192 K bytes — to provide a system capacity of 256 K bytes
- 448 K bytes — to provide a system capacity of 512 K bytes

a		c
+5V	1	+5 V
	2	
AOUT7	3	
AOUT6	4	AOUT5
AOUT4	5	AOUT3
	6	
	7	
	8	
	9	
	10	AOUT2
	11	AOUT1
OD0	12	AOUT0
OD1	13	OD2
OD3	14	
	15	
LGRD	16	LGRD
	17	
	18	
ID0	19	
ID1	20	
ID2	21	
ID3	22	
ID4	23	CAS7/
ID5	24	CAS6/
ID6	25	CAS5/
ID7	26	CAS4/
CAS2/	27	CAS1/
CAS3/	28	
OD4	29	OD5
RAS/	30	WE/
OD6	31	OD7
LGRD	32	LGRD

Figure 2.47 Memory expansion pin assignment

BLANK INTERFACE ADAPTER

This adapter provides the system integrator with the basic hardware to allow him to develop and install custom-built interfaces or options for special requirements.

The adapter consists of a small printed circuit board (blank) complete with an I/O bus connector, contained in a small housing that permits direct connection to the I/O bus of the NCR DECISION MATE V. The dimensions of the blank board are given in Figure 2.48. The developer must consider the standard bus I/O arrangement and the power limitations as specified in the I/O bus description of this manual.

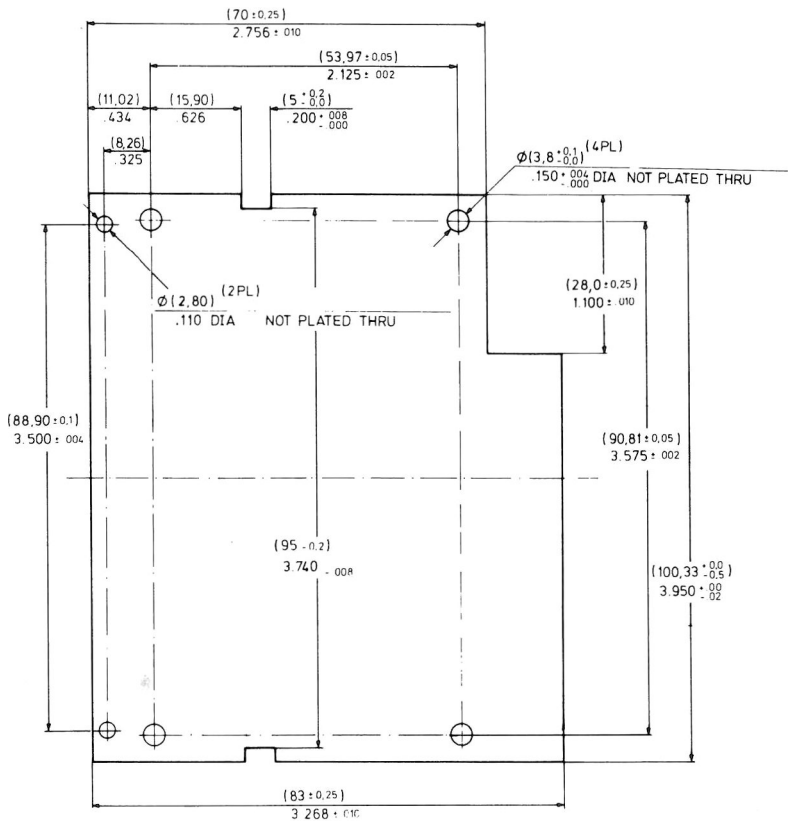


Figure 2.48 Blank interface board dimension

HINTS ON PRINTER CONNECTION

Printers (and other devices) recommended by NCR are fully compatible with the NCR DECISION MATE V; with the appropriate adapter the device can be connected to the computer, without any modifications, and it will function correctly. Many other devices, that have not been tested by NCR, are also fully compatible and can be connected without any modifications being carried out. Other RS-232C and Centronics-compatible devices may require some modifications to make them fully compatible with the NCR DECISION MATE V.

This section provides some hints on how to connect non-recommended devices.

- Study the descriptions and schematics in the hardware part of this manual
- Read the section "Interfacing Printers"
- Study the documentation provided with the device that is being connected to the computer
- Select the adapter kit that matches your requirements as near as possible
- Only one printer or plotter may be connected to the computer at a time

CENTRONICS PERIPHERAL ADAPTER (K210)

Generally, Centronics-compatible printers can be connected without difficulty. This type of interface offers the following advantages:

- No strapping required
- High transmission speed
- Status signals (e.g. out-of-paper) easily controlled by application software

NOTE: Some printers (e.g. Epson) are Centronics and RS-232C compatible and it may be necessary to make a printer hardware change; check the printer documentation. Many printers will not function until the operating system software has been loaded.

RS-232C PERIPHERAL ADAPTERS (K212, K213)

Using RS-232C compatible printers generally requires more careful selection of printer and interface. Some form of strapping is often required in three different areas:

- The printer
- The peripheral adapter
- The NCR DECISION MATE V software

Even when these three parts of the system appear to be compatible, it may be that the system does not function correctly. The following information should help in resolving some of the problems that may be encountered.

The operating system software used by the NCR DECISION MATE V (CP/M and MS-DOS) supports the X On — X Off protocol. The CONFIG utility in these operating systems allows for:

- Number of bits
- Parity selection
- Number of start/stop bits
- Baud rate

These must be set to match the settings in the printer exactly. It is possible that many printers do not meet these requirements except at certain baud rates and when they are in the “full-duplex” mode.

The NCR versions of CP/M and MS-DOS do not support the ETX/ACK protocol.

The kit K212 is designed to match the X On — X Off protocol and expects the printer to provide ON signals (+12V, space) at:

- Pin 20, Data Terminal Ready (DTR)
- Pin 4, Request to Send (RTS)

PRINTERS NOT SUPPORTING X ON— X OFF

If a printer does not send control data back to the computer, then control is achieved by a control line controlling the printer buffer. This line does not have a standard pin assignment, and may be connected to many different pins (e.g. 2, 4, 11, 13, 19, 20). Check the printer documentation to determine which pin is used.

Kit K212 is prepared for control by pin 4, Request To Send (RTS). See Figure 2.49.

Status Pin 4	NCR DECISION MATE V	Printer
On (+12V)	Transmit data	Printer ready
Off (-12V)	Stop data transmission	Printer busy

Figure 2.49 Printer control

In the RS-232C interface the signal RTS corresponds to Clear To Send (CTS), and this signal is used to control the Programmable Communication Interface 2651.

If the printer uses pin 20, Data Terminal Ready (DTR) for control, then the kit K213 may be used without further modification.

Printers that use other pins for control (2, 11, 13, 19) require either a change to the cabling, or a special intermediate connecting cable.

PRINTER FAILS TO PRINT

1. Check software strapping (CONFIG)
2. Check strapping of J1, J2 on RS-232C interface board
3. Check printer strapping
4. Check pin 5 (RTS) is on, +12V
5. Monitor interface with an oscilloscope or a digital analyser

INCORRECT PRINTING

In the X On — X Off mode, check:

1. Pin 20 (DTR) is on (+12V)
2. All strappings
3. Connect a bridge between TB1-4 and TB1-8 on the RS-232C interface board
4. Monitor interface with an oscilloscope or digital analyser

In other modes of operation check if TB1-8 (CTS) on the RS-232C interface board changes state when the printer buffer is full. If TB1-8 does not change state, connect this pin to the line that provides the printer buffer status.

APPENDIX A

INITIALIZATION FIRMWARE (CONTROLLER BOARD)

Rz 04
05
06
07
08
09
10

EPROM LOCATED FROM ADDR 0000H TO 0FFFH IS USED FOR
:
: - RAM-TEST (1000H- 0FFFFH)
: - DRIVER FOR KEYBOARD
: - DRIVER FOR FLEX CONTR.
: - DRIVER FOR CRT
: - LEVEL0-DIAGNOSTIC
: - SYSTEM-LOADER

```

74
75
76
77
78
79
80 F000
81 0000
82
83
84
85
86
87
88
89 0000
90 0010
91 0011
92 0020
93 0050
94 0060
95 0012
96 0013
97 0014
98 0000
99 0000
100 00B1
101 00B2
102 00B3
103
104
105
106
107
LED 5
108
*****
109
8
110 00FC
---- -X-
111 00FA
---- -X-
112 00F0
---- -X-
113 000E
-X- ----X
114 00BE
X- ----X
115 00EE
--X ----X
116 0000
xxx xxxX
117 007E
---- -X-
118 00FF
-----
119
120
121
122 1000
123 F800
124 F80E
125 F820
126
127
128
129
130
131 2006
132 2000
133 F700
134
;
;
;
; *** SYSTEM EQUATES ***
; *****
;
;
; STK EQU 0F000H ;STACK-POINTER
RAMEND EQU 00H ;RAM ENDADR +1 (MSB)
;
;
;
; *** "OUT" EQUATES ***
; *****
;
;
; ERRDIS EQU 00H ;LED SELECT
RANSEL EQU 10H ;LOWER RAM SELECT
ROMSEL EQU 11H ;ROM SELECT
DMASEL EQU 20H ;DMA CONTR BASE ADR
FDCSEL EQU 50H ;FLEX CONTR BASE ADR
FRISEL EQU 60H ;PRINTER
SETTC EQU 12H
SYSSTA EQU 13H ;SYSTEM-STATUS-SELECT
MON EQU 14H ;DISK MOTOR ON
PROCCH EQU 0D0H ;PROCESSOR CHANGE SWITCH
TIM0 EQU 80H ;COUNTER 0
TIM1 EQU 81H ;COUNTER 1
TIM2 EQU 82H ;COUNTER 2
TIMM0 EQU 83H ;CONTERS MODE
;
;
; *** LED-ERROR EQUATES ***
; *****
;
;
; NEMERR EQU 0F0H ;MEMORY-ERROR
DMAERR EQU 0F4H ;DMA-ERROR
KEYERR EQU 0F6H ;KEYBOARD-ERROR
FDCERR EQU 0DEH ;FLEX-ERROR
GDCERR EQU 0BEH ;GDC-ERROR
P16EFR EQU 0EEH ;16 BIT PROCESSOR ERROR
PRDERR EQU 00H ;PROZESSOR-ERROR
SUMERR EQU 7EH ;SUMCHECK-ERROR
DIACOM EQU 0FFH ;LEVELO COMPLETE
;
;
;
; VV200 EQU 01000H
VV205 EQU 0FB00H
VV206 EQU 0FB0EH
VV207 EQU 0FB20H
;
;
;
; DIAGNOSTIC BOX EQUATES
; *****
;
;
; ROMADR EQU 2006H
ANFDIA EQU 2000H
DIASLK EQU 0F700H
;

```

```

136
137
138
*****
139
140
141
142
*****
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169 0000
170 0001
171 0003
172 0004
173 0005
174 0006
175 0007
176 0008
177 0009
178
179
:
:
: *****
:
:
: FUNCTION OF FLEX DRIVES 5 1/4"
:
: *****
:
:
: ENTRY CONDITION: VIA CALL
: <HL> = ADR. OF PARAMETER-BLOCK
: <DE> = ADR. OF RESULT-BLOCK
:
:
: PARAMETER-BLOCK: XX FUNCTION
: XX DON'T CARE
: XX DON'T CARE
: XX UNIT NR
: XX TRACK NR
: XX SECTOR NR
: XX HEAD NR
: XX DENSITY
: XX DMA ADR LOW
: XX DMA ADR HIGH
:
:
: EXIT CONDITION: VIA RET
:
: RESULT BLOCK: 1-7 STATUS BYTES FROM 8272
:
:
: *** PARAMETER BLOCK OFFSET ***
: *****
:
:
:
: FFARAM EQU 0000H ;FUNCTION
: LFARAM EQU 0001H ;NOT USED
: UPARAM EQU 0003H ;UNIT NR
: TPARAM EQU 0004H ;TRACK NR
: SPARAM EQU 0005H ;SECTOR NR
: HPARAM EQU 0006H ;HEAD
: DPARAM EQU 0007H ;DENSITY
: ALFARA EQU 0008H ;DMA ADR LOW
: AHFARA EQU 0009H ;DMA ADR HIGH
:
:
:

```

```

181      ;
182      ;
183      ;
184      ;
185      ;
186      ;
187      0050      RDMST      EQU 50H
188      0050      FDCST      EQU 50H      ;MAIN-STATUS-REGISTER
189      0051      FDCRA      EQU 51H      ;READ DATA
190      0051      FDCWR      EQU 51H      ;WRITE DATA
191      0008      FDCIN      EQU 08H      ;FDC INT BIT IN SYSTEM STATUS
192      ;
193      ;
194      ;
195      ;
196      ;
197      0042      FRTRK      EQU 42H      ;READ A TRACK
198      ;
199      0003      FDCSPE      EQU 03H      ;SPECIFY
200      0004      FDCSEN      EQU 04H      ;SENSE DRIVE STATUS
201      0045      FDCWD      EQU 45H      ;WRITE DATA
202      0046      FDCRD      EQU 46H      ;READ DATA
203      0007      FDCRE      EQU 07H      ;RECALIBRATE
204      0008      FDCSIS      EQU 08H      ;SENSE INTERRUPT STATUS
205      004A      FDCRID      EQU 4AH      ;READ ID
206      004D      FDCFT      EQU 4DH      ;FORMAT A TRACK
207      000F      FDCSEE      EQU 0FH      ;SEEK
208      ;
209      ;
210      ;
211      ;
212      ;
213      ;
214      0008      FDCEOT      EQU 8H      ;FINAL SECTOR NR OF A TRACK
215      ;
216      ;
217      ;
218      ;
219      ;

```



```

221                                     ;
222                                     ;
223                                     ;
CRT ***
224                                     ; *****
****
225                                     ;
226                                     ;
227                                     ;
228 0000 3E 00                          LD      A,GDCRES
229 0002 D3 A1                          OUT     (GDCCOM),A
230 0004 21 D0 04                       LD      HL,INTPAR          ;RESET PARAMETER
LIST
231 0007 06 08                          LD      B,0BH
232 0009 0E A0                          LD      C,GDCPAR
233 000B ED B3                          OTIR                               ;SEND PARAMETER
234                                     ;
235 000D 3E 6F                          LD      A,VSYNM           ;MASTERMODE
236 000F D3 A1                          OUT     (GDCCOM),A
237 0011 3E 52                          LD      A,01010010B      ;COUNTER 1 ONE-SH
OT (MODE 1)
238 0013 D3 B3                          OUT     (TIMM0),A
239 0015 3E 20                          LD      A,32
240 0017 D3 B1                          OUT     (TIM1),A         ;MONO = 12,8 US
241                                     ;
242 0019 3E 01                          LD      A,,HIGH.BEGIN
243 001B ED 47                          LD      L,A
244 001D ED 56                          IM      I                 ;INTERRUPT-MODE1
245                                     ;
246 001F 06 05                          LD      B,5
247 0021 AF                          XOR     A                 ;CLEAR DIAG.INDIC
AT(ON-RAM(2006H-200AH)
248 0022 21 06 20                       LD      HL,ROMADR
249 0025 77                          DGT1:  LD      (HL),A
250 0026 23                          INC     HL
251 0027 10 FC                          DJNZ   DGT1
252                                     ;
253 0029 1B 17                          JR      DGTTEST
254                                     ;
255                                     ;
256                                     ; *****
*****
257                                     ;
258 003B                                ORG     3BH                ;INTERRUPT
259                                     ;
260 003B E5                          PUSH   HL
261 0039 F5                          PUSH   AF
262 003A ED 57                          LD     A,I
263 003C 67                          LD     H,A
264 003D 2E 12                          LD     L,1B
265 003F F1                          POP    AF
266 0040 E3                          EX     (SP),HL
267 0041 C9                          RET
268                                     ;
269                                     ; *****
*****
270                                     ;
271                                     ;
272 0042 3E B2                       DGTTEST: LD     A,10000010B          ;INIT DIA
G-PORT
273 0044 D3 0F                       OUT     (0FH),A             ;PDTRAC - OUTPUT
/MDD0
274                                     ;
- INPUT /MDD0
275 0046 3E 03                       LD     A,3
276 0048 D3 0E                       OUT     (0EH),A           ;DIAGBOX ROM/RAM
ENABLE
277                                     ;
278 004A 31 00 F7                     LD     SP,DIASTK         ;STACK FROM DIAG
279                                     ;
280 004D 21 06 20                     LD     HL,ROMADR
281 0050 7E                          LD     A,(HL)
282 0051 23                          INC    HL
283 0052 86                          ADD    A,(HL)
284 0053 23                          INC    HL

```

285 0054 86
286 0055 FE FE
287 0057 CC 00 20

G-BOX

288

289 005A AF

290 005B D3 0E

DISABLE

291 005D C3 FD 02

292

ADD A, (HL)
CF OFEH
CALL Z, ANFDIA

:CALL DIA

;

XOR A
OUT (0EH), A

:DIAG-BOX ROM/RAM

JF LEVEL0

:*****

358

```

294
295 0100
296
297 0100 C3 FD 02
GRAM
298 0103 C3 13 01
ARD
299 0106 C3 AB 07
ISC
300 0109 C3 24 01
FER/CURSORPOSITION
301 010C C3 70 06
STEM
302
303 0112
304
305 0112 C9
307
308
309
*****
310
311
312
313
*****
314
315
316
317
318
319 0040
320 0041
321 0041
322
323
324
325
326
327 0113 DB 41
328 0115 1F
329 0116 30 FB
330 0118 DB 40
ARACTER
331 011A C9
332
333 0118 DB 41
334 011D 1F
335 011E 3E 00
336 0120 D0
337 0121 DB 40
338 0123 C9
339

```

```

;
; ORG 100H
;
; BEGIN: JP LEVEL0 ;START OF ANY PRO
;
; JP KEYINP ;DRIVER FOR KEYBO
;
; JP FDCDR1 ;DRIVER FOR FLEXD
;
; JP MODCRT ;MODIFIES CRT-BUF
;
; JP STARTS ;START OF LOAD SY
;
;
; ORG 112H
;
; RET
;
;
; *****
;
; DRIVER FOR KEYBOARD-CONTROLLER 8741
;
; *****
;
;
; *** "IN/OUT" EQUATES ***
; *****
;
; KEYDAT EQU 40H ;READ KEYBOARD DATA
; KEYSTA EQU 41H ;READ " STATUS
; KEYCOM EQU 41H ;WRITE " COMMAND
;
;
; EXIT COND: <A>=ASCII-CHARACTER
;
;
;
; KEYINP: IN A,(KEYSTA)
; RRA
; JR NC,KEYINP
; IN A,(KEYDAT) ;READ KEYBOARD-CH
;
; RET
;
; KYLOOK: IN A,(KEYSTA)
; RRA
; LD A,0
; RET NC
; IN A,(KEYDAT)
; RET
;
;
;

```

102

RET

◀

✓

```

341
342
*****
343
344
345
346
*****
347
348
349
350
351
352
353
354 FB00
355 FB01
356 FB30
357 0050
358 001B
359 F900
360 F901
361 F902
362 F903
363 F905
364 F906
365 F907
366 F909
367 F90A
368 F90B
369 0002
370 FA00
371
372
373
374
375
376
377 008B
378 0001
379 000C
380 0000
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402 0001
403 0002
404 0004

```

```

;
;*****
;
; CRT-ROUTINES
;
;*****
;
;
;
; *** CRT-BUFFER EQUATES ***
; *****
;
;
; CURSX EQU 0F800H ;CURSOR - X - POSITION
; CURSY EQU 0F801H ;CURSOR - Y - POSITION
; BUFSTA EQU 0F830H ;CRT-BUFFER START ADR.
; CHARA EQU 80 ;80 CHARACTER/LINE
; ROWS EQU 24 ;24 LINES/SCREEN
; CEADL EQU 0F900H ;LOW BYTE OF CURSOR
; CEADH EQU 0F901H ;HIGH BYTE OF CURSOR
; CDAD EQU 0F902H ;POS OF PIXEL
; SP1 EQU 0F903H ;START OF PAGE 1
; LP11 EQU 0F905H ;LENGTH OF PAGE 1 LOW
; LP12 EQU 0F906H ; " " " 1 HIGH
; SP2 EQU 0F907H ;START OF PAGE 2
; LP21 EQU 0F909H ;LENGTH OF PAGE 2 LOW
; LP22 EQU 0F90AH ; " " " 2 HIGH
; INVFLG EQU 0F90BH
; WDAT1 EQU 02H
; CHRRAM EQU 0FA00H
;
;
; *** CONTROL CHARACTER EQUATES ***
; *****
;
;
; NEWL EQU 8BH ;NEW LINE
; HOME EQU 01H ;HOME POSITION
; SCRER EQU 0CH ;SCREEN ERASE
; NOMOV EQU 00H ;NO CURSOR MOVE
;
;
; ***ATTRIBUTES ***
; *****
;
; IF FARBE=1
;
; MONO
;
; ====
;
; INVER EQU 1 ;SCREEN INVERSE
; BLINK EQU 2 ;SCREEN BLINKING
; HLFIN EQU 4 ;HALF INDENSITY
;
; ENDIF
;
; IF FARBE
;
; COLOR
;
; =====
;
; NC EQU 1 ;NOT USED
; BLINK EQU 2
; FR EQU 4 ;FRONT RED

```

DECISION-MATE-FIRMWARE AMF-DMCOCH0104-00 12

```

405 0008
406 0010
407 0020
408 0040
409 0080
410
411
412

```

```

;
;
; FG EQU 8 ; " GREEN
; FR EQU 10H ; " BLUE
; BR EQU 20H ;BACK RED/
; BG EQU 40H ; " GREEN/
; BB EQU 80H ; " BLUE/
;
;
; ENDIF
;

```

414	:			
415	:			
416	:			
*****	:			
417	:			
*****	:			
418	:			
419	:		GDC-EQUATES	
420	:			
421	:			
422	:			
*****	:			
423	:			
*****	:			
424	:			
425	:			
426	:		*** GDC PORT EQUATES ***	
427	:		*****	
428	:			
429 00A2	GDCPAZ EQU	0A2H	:	WRITE ZOOM-PARAMETER + Z
OOM-REG.				
430 00A1	GDCCOM EQU	0A1H	:	WRITE COMMAND
431 00A0	GDCPAR EQU	0A0H	:	WRITE PARAMETER
432 00A0	GDCSTA EQU	0A0H	:	READ STATUS
433 00A1	FIFO EQU	0A1H	:	READ DATA
434	:			
435	:		*** GDC STATUS EQUATES ***	
436	:		*****	
437	:			
438 0001	DATRDY EQU	01H	:	A BYTE IS AVAILABLE FOR
READ				
439 0002	FIFULL EQU	02H	:	FIFO IS FULL
440 0004	FIFEMP EQU	04H	:	FIFO IS EMPTY
441 000B	DRWINP EQU	0BH	:	DRAWING IN PROCESS
442 0010	DMAEXC EQU	10H	:	DMA DATA TRANSFER IN PRO
CESS				
443 0020	VERETR EQU	20H	:	VERTICAL RETRACE IN PROC
ESS				
444 0040	HORETR EQU	40H	:	HORIZONTAL RETRACE IN PR
OCESS				
445 00B0	LIPDET EQU	80H	:	LIGHT PEN DEECTED (ADDR
ESS VALID)				
446	:			
447	:			
448	:		*** GDC COMMAND EQUATES ***	
449	:		*****	
450	:			
451 0000	GDCRES EQU	00H	:	RESET-BLANK DISPLAY, IDLE
-MODE, INIT				
452 006E	VSYNCS EQU	6EH	:	SLAVE-MODE
453 006F	VSYNCM EQU	6FH	:	MASTER-MODE
454 004B	CCHAR EQU	4BH	:	CURSOR/CHARACTER CHARACT
ERISTICS				
455 006B	START EQU	6BH	:	START DISPLAY/END IDLE-M
ODE				
456 0046	ZOOM EQU	46H	:	SPECIFY ZOOM-FACRDR
457 0049	CURS EQU	49H	:	SPECIFY CURSOR-POSITION
458 0047	PITCH EQU	47H	:	PITCH SPECIFY
459 004A	MASK EQU	4AH	:	LOAD MASK-REGISTER
460 004C	FIGS EQU	4CH	:	SPECIFY FIGURE DRAWING-P
ARAMETER				
461 006C	FIGD EQU	6CH	:	START FIGURE DRAW
462 006B	GCHRD EQU	6BH	:	START GRAPHICS CHARACTER
DRAW				
463 00E0	CURD EQU	0E0H	:	READ CURSOR-ADDRESS
464 00C0	LPRD EQU	0C0H	:	READ LIGHT-PEN ADDRESS
465 0070	PRAM EQU	70H	:	LOAD PARAMETER-RAM
466 0000	PRAMSA EQU	00H	:	LOWER 4 BITS ARE STARTIN
G ADR. IN RAM				
467 0020	WDAT EQU	20H	:	WRITE DATA INTO DISPLAY
MEMORY				
468				(COMMAND+TYPE+MODE)
469 0000	TYWORD EQU	00H	:	DATA TRANSFER BYTES
470 0010	TYLOBY EQU	10H	:	LOW BYTE OF WORD
471 001B	TYHIBY EQU	1BH	:	HIGH BYTE " "
472				MODE OF RMW MEMORY CYCLE
473 0006	MOREPL EQU	00H	:	REPLACE WITH PATTERN
474 0001	MOCOMP EQU	01H	:	COMPLEMENT
475 0002	MORES EQU	02H	:	RESET TO 0
476 0003	MOSET EQU	03H	:	SET TO 1
477 00A0	RDAT EQU	0A0H	:	READ DATA FROM DISPLAY M

EM.

```

478 00A4          DMAR   EQU    0A4H   ;DMA-READ-REQUEST
479 0024          DMAW   EQU    24H   ;DMA WRITE-REQUEST
480 0050          NRWAPL EQU    80    ;# OF WORD ADDRESSES/LINE
481 000A          LFC    EQU    10    ;LINES PER CHARACTER
482 000D          SCRON  EQU    0DH   ;SCREEN ON
483 000C          SCROFF EQU    0CH   ;SCREEN OFF
484 000F          SYNCON EQU    0FH
485 000E          SYNCOF EQU    0EH
486              ;
487              ;
488              ;      *** GDC RESET PARAMETER ***
489              ;      *****
490              ;
491 0000          RESMOP EQU    00H   ;MODE OF OPERATION SELECT
BITS
492              ; (RESMOP+DISPLAY+FRAME+Dy
NRAM+WINDOW)
493              ; DISPLAY-MODE
494 0000          MIXGAC EQU    00H   ;MIXED-MODE
495 0002          GRAMOD EQU    02H   ;GRAPHIC-MODE
496 0020          CHAMOD EQU    20H   ;CHARACTER-MODE
497 0000          NOINTL EQU    00H   ;NON INTERLACED
498 0008          INLRFF EQU    08H   ;
499 0007          INTLAC EQU    07H   ;INTERLACED
500 0000          STATRM EQU    00H
501 0004          DYNRAM EQU    04H   ;REFRESH
502 0000          DRWALL EQU    00
503 0010          DRWRET EQU    10H   ;DRAWING ONLY DURING RETR
ACE BLANKING
504              ;

```

```

506
507
508 0124
509 0124 E5
510 0125 C5
511 0126 D5
512 0127 F5
513
514
515
516
517
518 0128 FE 00
519 012A CA 9C 01
520 012D FE 88
521 012F CA 3F 01
522 0132 FE 01
523 0134 CA 58 01
524 0137 FE 0C
525 0139 CA 62 01
526
527 013C C3 68 01
UPPER DEPENDING
528 013F AF
529 0140 32 00 F8
530 0143 3A 01 F8
531 0146 3C
532 0147 FE 19
533 0149 C2 52 01
534 014C F5
535 014D CD 30 02
536 0150 F1
537 0151 3D
538 0152 32 01 F8
539 0155 C3 9C 01
540
541
542
543
544
545 0158 AF
546 0159 32 00 F8
547 015C 32 01 F8
548 015F C3 9C 01
549
550
551
552
553
554
555 0162 CD 13 02
556 0165 C3 58 01
557
558
559
560
561
562
563 0168 3A 00 F8
564 016B FE 50
565 016D C2 78 01
566 0170 3A 01 F8
567 0173 3C
568 0174 32 01 F8
569 0177 AF
;
;
MODCRT:  PUSH    HL
          PUSH    BC
          PUSH    DE
          PUSH    AF      ;SAVE ASCII-CHARACTER
;
; *** CHECK IF ANY CONTROL CHARACTER ***
; *****
;
;
          CP      NOMOV
          JP      Z,GETRE      ;NO CURSOR MOVE
          CP      NEWL
          JP      Z,CARRET    ;CARRIDGE RETURN
          CP      HOME
          JP      Z,CHP      ;HOME POSITION
          CP      SCRER
          JP      Z,SCERA    ;SCREEN ERASE
;
          JP      SETCHA    ;SET CHARACTER INTO CRT-B
CARRET:  XOR      A
          LD      (CURSX),A
          LD      A,(CURSY)
          INC     A
          CP      25      ;SCROLL IF = 25
          JP      NZ,CAR1
          PUSH   AF
          CALL   SCRRLM
          POP    AF
          DEC    A
CAR1:   LD      (CURSY),A
          JP      GETRE
;
; *** CURSOR HOME POSITION ***
; *****
;
;
CHP:    XOR      A
          LD      (CURSX),A
          LD      (CURSY),A
          JP      GETRE
;
;
; *** SREEN ERASE ***
; *****
;
;
SCERA:  CALL    MCLSCR
          JP      CHP
;
;
; *** SET CHARACTER ***
; *****
;
;
SETCHA: LD      A,(CURSX)
          CP      80
          JP      NZ,CHA1
          LD      A,(CURSY)
          INC     A
          LD      (CURSY),A
          XOR     A

```

```

570 0178 32 00 FB
571 0178 4F
572 017C 3A 01 FB
573 017F FE 19
574 0181 F5
575 0182 C5

576 0183 CC 30 02
577 0186 C1
578 0187 F1
579 0188 C2 8C 01
580 0188 3D
581 018C 47
582 018D 32 01 FB
583 0190 3A 00 FB
584 0193 3C
585 0194 32 00 FB
586 0197 F1
587 0198 F5
588 0199 CD A1 01
ACTER
589
x-POS
590 019C F1
591 019D D1
592 019E C1
593 019F E1
594 01A0 C9
595
596

```

```

CHA1: LD (CURSX),A
LD C,A
LD A,(CURSY)
CP 25
PUSH AF ;<A> = Y POSITION
PUSH BC ;<C> = X POSITION

CALL Z,SCROLLM
POP BC
POP AF
JP NZ,CHA2
DEC A

CHA2: LD B,A ;<B> = Y POSITION
LD (CURSY),A
LD A,(CURSX)
INC A
LD (CURSX),A
POP AF
PUSH AF
CALL WRMCHR ;<A> = ASCII CHAR

; <B>=Y-POS, <C> =

GETRE: POP AF
POP DE
POP BC
POP HL
RET
:
:

```



```

598
599
600
*****
601
602
603
604
605
*****
606
607
608
609
610
611
612
613
614
615
616
617 01A1 E5
618 01A2 D5
619 01A3 C5
620 01A4 F5
621
622 01A5 CD E5 01
TER-FOS.
623 01A8 CD C2 02
624 01AB CD C1 01
625 01AE 3E 20
626 01B0 CD A0 04
627 01B3 F1
TER
628 01B4 CD A6 04
629 01B7 3A 0B F9
630 01BA CD A6 04
631 01BD C1
632 01BE D1
633 01BF E1
634 01C0 C9
635
636
637 01C1 CD A4 02
RITE CHARACTER
638 01C4 3E 4C
639 01C6 CD A0 04
640 01C9 3E 02
641 01CB CD A6 04
642 01CE AF
643 01CF CD A6 04
644 01D2 C9
645
646 01D3 CD A4 02
EAD CHARACTER
647 01D6 3E 4C
648 01D8 CD A0 04
649 01DB 3E 02
650 01DD CD A6 04
651 01E0 3D
652 01E1 CD A6 04
653 01E4 C9
654
655 01E5 2A 03 F9
656 01E8 7D
657 01E9 B4
658 01EA C2 FB 01
659 01ED 21 50 00
L>
660 01F0 78
661 01F1 CD 68 0A
;
;
; *****
;
; G D C - CHARACTER - MODE-DRIVER
; (IN MIXED-MODE)
;
; *****
;
; ENTRY: <A> = .ASCII-CHARACTER
; <B> = Y-POSITION
; <C> = X-POSITION
;
;
; **** WRITE CHARACTER INTO CRT-RAM ****
; *****
;
; WRMCHR: PUSH HL
; PUSH DE
; PUSH BC
; PUSH AF
;
; CALL CALCHR ;CALCULATE CHARAC
;
; CALL SETCU
; CALL SETCCA ;SEND CHARACTER
; LD A,NDAT+TYWORD+MOREFL
; CALL OUTCMD
; POP AF ;<A>=ASCII-CHARAC
;
; CALL OUTPAR
; LD A,(INVFLG) ;FETCH ATTRIBUT
; CALL OUTPAR
; POP BC
; POP DE
; POP HL
; RET
;
; SETCCA: CALL SETMSK ;SET LENGTH FOR W
;
; LD A,FIGS
; CALL OUTCMD
; LD A,2
; CALL OUTPAR
; XOR A
; CALL OUTPAR
; RET
;
; REACCA: CALL SETMSK ;SET LENGTH FOR R
;
; LD A,FIGS
; CALL OUTCMD
; LD A,2
; CALL OUTPAR
; DEC A
; CALL OUTPAR
; RET
;
; CALCHR: LD HL,(SP1)
; LD A,L
; OR H
; JP NZ,CALCH1
; LD HL,80 ;Y*80 + X = <CEAD
;
; LD A,B ;<B>= Y-POS
; CALL MULT

```



```

726 025F 22 03 F9          LD      (SP1),HL
727                          ;
728 0262 3A 06 F9          LD      A,(LP12)
729 0265 3D                DEC     A                ;LENGTH1 - 1
730 0266 CA EB 02          JP      Z,INI10
731 0269 32 06 F9          LD      (LP12),A
732                          ;
733 026C 3A 0A F9          LD      A,(LP22)
734 026F 3C                INC     A                ;LENGTH2 + 1
735 0270 32 0A F9          LD      (LP22),A
736 0273 C3 DA 02          JP      SCROL1
737 0276
738                          ;
739                          ;
740                          ;
741                          ;
742 0274 21 00 00          INITIA: LD      HL,0
743 0279 22 03 F9          LD      (SP1),HL
744 027C 22 07 F9          LD      (SP2),HL
745 027F AF                XOR     A
746 0280 32 0A F9          LD      (LP22),A
747 0283 3E 19            LD      A,25
748 0285 32 06 F9          LD      (LP12),A
749 0288 C9                RET
750                          ;
751 0289 CD B2 02          SETCR: CALL   CUHOM1    ;WHOLE GRAPHIC HO
ME
752 028C CD A4 02          CALL   SETMSK
753 028F 3E 4C            LD      A,FIGS        ;SET DIRECTION AN
D NUMBER AND WORDS
754 0291 CD A0 04          CALL   OUTCMD
755 0294 3E 02            LD      A,2           ;DIR = 2
756 0296 CD A6 04          CALL   OUTPAR
757 0299 3E FF            LD      A,OFFH        ;LENGTH 16K WORD
758 029B CD A6 04          CALL   OUTPAR
759 029E 3E 3F            LD      A,3FH         ;HIGH BYTE
760 02A0 CD A6 04          CALL   OUTPAR
761 02A3 C9                RET
762                          ;
763 02A4 3E 4A            SETMSK: LD      A,MASK ;SET MASK TO STOR
ED VALUES
764 02A6 CD A0 04          CALL   OUTCMD
765 02A9 3E FF            LD      A,OFFH        ;LOAD LOW BYTE
766 02AB CD A6 04          CALL   OUTPAR
767 02AE CD A6 04          CALL   OUTPAR
768 02B1 C9                RET
769                          ;
770 02B2 3E 49            CUHOM1: LD      A,CURS ;CURSOR HOME
771 02B4 CD A0 04          CALL   OUTCMD
772 02B7 AF                XOR     A
773 02B8 CD A6 04          CALL   OUTPAR
774 02BB CD A6 04          CALL   OUTPAR
775 02BE CD A6 04          CALL   OUTPAR
776 02C1 C9                RET
777                          ;
778 02C2 3E 49            SETCU:  LD      A,CURS ;SET CURSOR
779 02C4 CD A0 04          CALL   OUTCMD
780 02C7 3A 00 F9          LD      A,(CEADL)    ;LOW BYTE OF CURS
OR
781 02CA CD A6 04          CALL   OUTPAR
782 02CD 3A 01 F9          LD      A,(CEADH)    ;HIGH BYTE OF CUR
SOR
783 02D0 CD A6 04          CALL   OUTPAR
784 02D3 3A 02 F9          LD      A,(CDAD)     ;POSITON OF PIXEL
785 02D6 CD A6 04          CALL   OUTPAR
786 02D9 C9                RET
787                          ;
788 02DA 3E 70            SCROL1: LD      A,PRAM+0
789 02DC CD A0 04          CALL   OUTCMD        ;SET START AND LE
NGTH OF

```

2	790 02DF 06 08	LD	B,8	;PAGE 1 AND PAGE
	791 02E1 21 03 F9	LD	HL,SP1	
	792 02E4 CD C6 04	CALL	SENPAR	
	793 02E7 C9	RET		
	794	:		
	795 02E8 21 00 00	INI10:	LD HL,0	
	796 02EB 22 03 F9		LD (SP1),HL	
	797 02EE 22 07 F9		LD (SP2),HL	;START OF PAGE 1
= 0	798 02F1 AF	XOR	A	
	799 02F2 32 0A F9	LD	(LP22),A	;LENGTH OF PAGE 2
= 0				
	800 02F5 3E 19	LD	A,25	
	801 02F7 32 06 F9	LD	(LP12),A	;LENGTH OF PAGE 1
= 25				
	802 02FA C3 DA 02	JP	SCROL1	
	803	:		
	804	:		
	805	:		

```

B12 ;*****
*****
B13 ;
B14 ; POWER UP POUTINE
B15 ;
B16 ;*****
*****
B17 ;
B18 ;
B19 02FD 3E 00 358 LEVEL0: LD A,00
B20 02FF D3 00 OUT (ERRDIS),A ; SWITCH 0
N DIAGNOSTIC-LED
B21 0301 3E 00 LD A,GDCRES
B22 0303 D3 A1 OUT (GDCCOM),A ; RESET GD
C
B23 ;
B24 0305 3E 00 PROCHK: LD A,PROERR
B25 0307 D3 00 OUT (ERRDIS),A ; SWITCH 0
N PROZESSOR-ERROR-LED
B26 ;
B27 ;
B28 ; *** SUMCHECK ***
B29 ; *****
B30 ;
B31 ;
B32 0309 3E 7E SUMCHK: LD A,SUMERR
B33 030B D3 00 OUT (ERRDIS),A ; SUMCHECK
-ERROR
B34 030D AF XOR A
B35 030E 21 00 00 LD HL,0
B36 0311 11 00 20 LD DE,2000H ; ROM-TEST
OVER BK
B37 0314 86 SUM1: ADD A,(HL)
B38 0315 23 INC HL
B39 0316 1D DEC E
B40 0317 C2 14 03 JP NZ,SUM1
B41 031A 15 DEC D
B42 031B C2 14 03 JP NZ,SUM1
B43 031E C6 00 ADD A,0
B44 0320 C2 14 06 JF NZ,ENDLO ; JUMP IF
SUMCHECK-ERROR
B45 ;
B46 ;
B47 ;
B48 ; *** RAM-TEST ***
B49 ; *****
B50 ;
B51 0323 3E FC RAMTST: LD A,MEMERR
B52 0325 D3 00 OUT (ERRDIS),A
B53 0327 21 00 20 LD HL,2000H
B54 032A 06 00 LD B,RAMEND
B55 ;
B56 ;
B57 032C 36 55 LL124 LD (HL),055H ; TEST BYTE 55H
B58 032E 23 INC HL
B59 032F 36 AA LD (HL),0AAH
B60 0331 23 INC HL
B61 0332 7C LD A,H
B62 0333 B8 CF B
B63 0334 C2 2C 03 JP NZ,LL124 ; MEMORY WRITE COM
PLETE ?
B64 0337 06 1F LD B,01FH
B65 ;
B66 0339 2B LL125 DEC HL
B67 033A 7C LD A,H
B68 033B B8 CF B ; MEMORY READ COMPLETE ?
B69 033C CA 4F 03 JF Z,LL126 ; YES - NEXT TEST
B70 033F 7E LD A,(HL)

```

871 0340 FE AA		CP	OAAH
872 0342 C2 14 06		JP	NZ,ENDLO
873 0345 2B		DEC	HL
874 0346 7E		LD	A,(HL)
875 0347 FE 55		CP	055H
876 0349 CA 39 03		JP	Z,LL125 ;JUMP IF OK
877 034D C3 14 06		JP	ENDLO ;JUMP IF ERROR
878			
879 034F 23	:	LL126 INC	HL ;<HL>= RAM START ADR.
880 0350 06 00		LD	B,RAMEND ;RAM ENDADR. + 1
881	:		
882 0352 36 AA	LL127	LD	(HL),OAAH
883 0354 23		INC	HL
884 0355 36 55		LD	(HL),055H
885 0357 23		INC	HL
886 0358 7C		LD	A,H
887 0359 B8		CP	B
888 035A C2 52 03		JP	NZ,LL127 ;MEMORY WRITE COM
DELETE ?			
889 035D 06 1F		LD	B,01FH ;YES !
890	:		
891 035F 2B	LL128	DEC	HL
892 0360 7C		LD	A,H
893 0361 B8		CP	B
894 0362 CA 75 03		JP	Z,NEXT1 ;JUMP IF TEST COMPLETE
895 0365 7E		LD	A,(HL)
896 0366 FE 55		CP	055H
897 0368 C2 14 06		JP	NZ,ENDLO ;JUMP IF MEMORY-E
RROR			
898 036B 2B		DEC	HL
899 036C 7E		LD	A,(HL)
900 036D FE AA		CP	OAAH
901 036F CA 5F 03		JP	Z,LL128
902 0372 C3 14 06		JP	ENDLO
903	:		
904	:		

387

CAA

Next 2

3Dφ

JMP 41A NEXT2

```

908 ; *** KEYBOARD-CONTROLLER-TEST ***
909 ; *****
910 ;
911 ;
912 0375 3E F6 → NEXT1: LD A,KEYERR JMP 41A NEXT2
913 0377 D3 00 OUT (ERRDIS),A
914 ;
915 0379 31 00 F0 ; KEYST: LD SP,STK ;*** SET STACK-PO
INTER *** ;
916 037C CD A7 03 ; CALL INVAL1 ;GET INVALID CHAR
ACTER'S ;
917 ;
918 037F DB 41 ; KEY1: IN A,(KEYSTA)
919 0381 1F RRA
920 0382 1F RRA
921 0383 DA 75 03 ; JF C,NEXT1 ;WAIT IF INPUTBUF
FER EMPTY ;
922 0386 3E 00 LD A,0
923 0388 D3 41 ; OUT (KEYCOM),A ;SEND SELF-CHECK-
COMMAND ;
924 ;
925 038A DB 41 ; KEY2: IN A,(KEYSTA)
926 038C 1F RRA
927 038D D2 8A 03 ; JF NC,KEY2
928 0390 DB 40 ; IN A,(KEYDAT) ;GET SELF-TEST-ST
ATUS ;
929 0392 FE 55 ; CP 55H
930 0394 C2 14 06 ; JF NZ,ENDLO ;KEYBOARD-CONTROL
LER NOT O.K. ;
931 ;
932 0397 DB 41 ; KEY3: IN A,(KEYSTA)
933 0399 1F RRA
934 039A D2 97 03 ; JF NC,KEY3
935 039D DB 40 ; IN A,(KEYDAT) ;GET COUNTRY-CODE
1111EVVV ;
936 039F CB 5F ; BIT 3,A ;E=0 --- ERROR ON
KEYBOARD ;
937 03A1 C2 BF 03 ; JF NZ,NEXT2 41A ;JUMP IF KEYBOARD
-TEST O.K. ;
938 03A4 C3 14 06 ; JF ENDLO
939 ;
940 ;
941 ;
942 03A7 DB 41 ; INVAL1: IN A,(KEYSTA)
943 03A9 1F RRA
944 03AA D0 ; RET NC
945 03AB DB 40 ; IN A,(KEYDAT) ;DUMMY READ !!
946 03AD 16 05 ; LD D,5 ;MULTIPLIER OF 10
MS ;
947 03AF 01 FE 04 ; INVAL2: LD BC,4FEH
948 03B2 0B ; INVAL3: DEC BC
949 03B3 78 ; LD A,B
950 03B4 B1 ; OR C
951 03B5 C2 B2 03 ; JF NZ,INVAL3 ;WAIT 10MS
952 03B8 15 ; DEC D
953 03B9 C2 AF 03 ; JF NZ,INVAL2
954 03BC C3 A7 03 ; JF INVAL1 ;GET NEXT CHARACT
ER ;
955 ;
956 ;
    
```

```

958
959
960
*****
961
962
963
964
*****
965
966
967
968
969
970
971 0028
972 0028
973 0028
974 002C
975 0024
976 0025
977 0020
978 002A
979 0026
980 0027
981
982
983
984
985
986
987
988 03BF 3E FA
989 03C1 D3 00
990 03C3 D3 2D
A
991
992 03C5 3E 10
993 03C7 D3 28
994 03C9 3E 0B
995 03CB D3 28
996 03CD DB 28
TUS
997 03CF E6 0F
X0
998 03D1 FE 00
999 03D3 C2 EE 03
1000 03D6 3E 55
1001 03DB D3 26
1002 03DA 3E AA
1003 03DC D3 26
1004 03DE D3 2C
1005 03E0 DB 26
1006 03E2 FE 55
1007 03E4 C2 EE 03
1008 03E7 DB 26
1009 03E9 FE AA
1010 03EB CA F3 03
1011
1012 03EE D3 2C
1013 03F0 C3 14 06
1014
1015

;
;
; *****
;
; DMA-CONTROLLER 8237
;
; *****
;
;
;
; *** DMA EQUATES ***
; *****
;
;
; DMAST EQU 28H ; DMA-STATUS
; DMACOM EQU 28H ; DMA-COMMAND
; DMAMD EQU 2BH ; DMA-MODE
; DMAFF EQU 2CH ; DMA-FIRST/LAST-BUFFER
; C2AD EQU 24H ; DMA-CHANNEL 2 ADDRESS-BUFFER
; C2TC EQU 25H ; DMA-CHANNEL 2 BLOCKLENGTH
; C0AD EQU 20H ; DMA-CHANNEL 0 ADDRESS-BUFFER
; DMAMB EQU 2AH ; DMA-START/STOP-CHANNEL
; C3AD EQU 26H ; DMA-CHANNEL 3 ADDRESS-BUFFER
; C3TC EQU 27H ; DMA-CHANNEL 3 BLOCKLENGTH
;
;
;
; *** TEST OF DMA-CONTROLLER ***
; *****
;
;
;
NEXT2: LD A, DMAERR
OUT (ERRDIS), A
OUT (2DH), A ; RESET DM

;
;
; DMATS: LD A, 10H
; OUT (DMACOM), A
; LD A, 00001011B
; OUT (DMAMD), A
; IN A, (DMAST) ; READ STA
; AND 0FH ; MUST BE

;
;
; CP 0
; JP NZ, ERRDMA
; LD A, 55H
; OUT (C3AD), A
; LD A, 0AAH
; OUT (C3AD), A
; OUT (DMAFF), A
; IN A, (C3AD)
; CP 55H
; JP NZ, ERRDMA
; IN A, (C3AD)
; CP 0AAH
; JP Z, GDCINT
;
; ERRDMA: OUT (DMAFF), A
; JP ENDLO
;
;

```



```

1020
*****
1021
1022
1023
1024
*****
1025
1026
1027
1028
1029 03F3 3E BE
1030 03F5 D3 00
1031 03F7 21 00 FB
ROM OFB00H TO OFFFH
1032 03FA AF
1033 03FB 77
1034 03FC 2C
1035 03FD C2 FB 03
1036 0400 24
1037 0401 C2 FB 03
1038 0404 3E EB
CK
1039 0406 32 0B F9
1040
1041
1042 0409 3E 00
1043 040B D3 A1
1044 040D 21 D0 04
LIST
1045 0410 06 08
1046 0412 CD C6 04
1047
1048 0415 3E 6F
1049 0417 CD A0 04
1050 041A 3E 47
1051 041C CD A0 04
1052 041F 3E 50
SES IN HOR. DIR.
1053 0421 CD A6 04
1054
1055 0424 3E 70
1056 0426 CD A0 04
WAM
1057 0429 21 D8 04
1058 042C 06 04
1059 042E CD C6 04
1060
1061 0431 3E 4B
1062 0433 CD A0 04
1063 0436 3E BF
1064 0438 CD A6 04
1065
1066 043B 3E CE
1067 043D CD A6 04
NG
1068 0440 3E 72
1069 0442 CD A6 04
1070
1071 0445 3E 46
1072 0447 CD A0 04
1073 044A AF
1074 044B CD 9A 04
1075

```

```

*****
:
:   INITIALISATION OF 7220 (GDC)
:
*****
:   *** CLEAR GDC-RAM-PARAMETER-ARRAY***
:   *****
:
:
GDCINT: LD   A,GDCERR
        OUT (ERRDIS),A
        LD   HL,CURSX      ;CLEAR RAM (00) F
:
CLRR:   XOR   A
        LD   (HL),A
        INC  L
        JF   NZ,CLRR
        INC  H
        JF   NZ,CLRR
        LD   A,0EBH      ;SET GREEN OF BLA
:
        LD   (INVFL),A
:
:
        LD   A,GDCRES
        OUT (GDCCOM),A
        LD   HL,INTPAR   ;RESET PARAMETER
:
        LD   B,0BH
        CALL SENPAR      ;SEND PARAMETER
:
INIT10: LD   A,VSYNCH
        CALL OUTCMD
        LD   A,PITCH
        CALL OUTCMD
        LD   A,NRWAPL    ;# OF WORD-ADDRES
:
        CALL  OUTPAR
:
        LD   A,FRAM+FRAMSA
        CALL  OUTCMD     ;LOAD PARAMETER R
:
        LD   HL,FRMPAR
        LD   B,4
        CALL  SENPAR    ;# OF PARAMETER S
:
        LD   A,CCHAR
        CALL  OUTCMD    ;DEFINE CURSOR
        LD   A,BFH
        CALL  OUTPAR    ;OUT COMMAND
:
        LD   A,0CEH
        CALL  OUTPAR    ;INIT CURSOR
:
        LD   A,72H
        CALL  OUTPAR    ;UNDERLINE BLINK1
:
        LD   A,ZOOM
        CALL  OUTCMD
        XOR   A
        CALL  OUTFAZ    ;ZOOM-FACTOR = 0
:

```

```

1077 ;*****
*****
1078 ;
1079 ; GDC-TEST
1080 ;
1081 ;*****
*****
1082 ;
1083 ;
1084 044E CD B2 02 ; CALL CUHOM1 ;SET CURSOR TO 00
00 ;
1085 0451 CD C1 01 CALL SETCCA
1086 0454 3E 20 LD A,WDAT+TYWORD+MOREFL
1087 0456 CD A0 04 CALL OUTCMD
1088 ;
1089 0459 3E 55 ; LD A,55H ;WRITE CHARACTER
55H ;
1090 045B CD A6 04 CALL OUTPAR
1091 045E 2F CPL ;WRITE ATTRIB. 0A
AH ;
1092 045F CD A6 04 CALL OUTPAR
1093 ;
1094 0462 3E E0 LD A,CURD ;READ CURSOR
1095 0464 CD A0 04 CALL OUTCMD
1096 0467 CD C0 04 CALL INPAR ;MUST BE 01
1097 046A FE 01 CF 01
1098 046C C2 14 06 JP NZ,ENDLO ;CURSOR READ/WRITE
E-ERROR ;
1099 046F CD C0 04 CALL INPAR
1100 0472 A7 AND A
1101 0473 C2 14 06 JP NZ,ENDLO ;CURSOR READ/WRITE
E-ERROR ;
1102 ;
1103 0476 CD B2 02 CALL CUHOM1 ;CURSOR - 1
1104 0479 CD D3 01 CALL REACCA
1105 047C 3E A0 LD A,RDAT ;READ WRITTEN CHA
RACTER ;
1106 047E CD A0 04 CALL OUTCMD
1107 0481 CD C0 04 CALL INPAR ;MUST BE 55H AND
0AAH ;
1108 0484 47 LD B,A
1109 0485 CD C0 04 CALL INPAR
1110 0488 80 ADD A,B
1111 0489 3C INC A
1112 048A C2 14 06 JP NZ,ENDLO ;CHARACTER READ/W
RITE-ERROR ;
1113 ;
1114 ;
1115 ;
1116 048D 3E 0C INIT20: LD A,SCRER
1117 048F CD 24 01 CALL MODCRT ;CLEAR SCREEN
1118 ;
1119 0492 3E 6B LD A,START ;START DISPLAY
1120 0494 CD A0 04 CALL OUTCMD
1121 0497 C3 DE 04 JP INITTI ;INIT OF GDC FINI
SHED ;
1122 ;
1123 ;
1124 ;

```

```

1128                                     ;      *** SUBROUTINES OF GDC ***
1129                                     ;      *****
1130                                     ;
1131                                     ;
1132 049A CD AC 04      OUTPAZ: CALL   FIFRDY
1133 049D D3 A2      OUT      (GDCPAZ),A
1134 049F C9      RET
1135                                     ;
1136 04A0 CD AC 04      OUTCMD: CALL   FIFRDY      ;PUT A COMMAND IN
FIFO
1137 04A3 D3 A1      OUT      (GDCCOM),A
1138 04A5 C9      RET
1139                                     ;
1140 04A6 CD AC 04      OUTPAR: CALL   FIFRDY      ;PUT A PARAMETER
TO FIFO
1141 04A9 D3 A0      OUT      (GDCCPAR),A
1142 04AB C9      RET
1143                                     ;
1144 04AC F5      FIFRDY: PUSH   AF
1145 04AD DB A0      FIF010: IN     A,(GDCSTA)      ;READ GDC-STATUS
1146 04AF E6 02      AND     FIFULL
1147 04B1 C2 AD 04      JF     NZ,FIF010      ;JUMP IF FIFO FUL
L
1148 04B4 F1      POP     AF
1149 04B5 C9      RET
1150                                     ;
1151 04B6 F5      CKDATR: PUSH   AF      ;CHECK AND WAIT F
OR DATA READY
1152 04B7 DB A0      CKDA10: IN     A,(GDCSTA)      ;READ GDC-STATUS
1153 04B9 E6 01      AND     DATRDY
1154 04BB CA B7 04      JF     Z,CKDA10      ;JUMP IF DATA-BYT
E NOT READY
1155 04BE F1      POP     AF
1156 04BF C9      RET
1157                                     ;
1158 04C0 CD B6 04      INPAR: CALL   CKDATR      ;READ FROM FIFO
1159 04C3 DB A1      IN     A,(FIFO)
1160 04C5 C9      RET
1161                                     ;
1162 04C6 7E      SENPAR: LD     A,(HL)      ;LOAD PARAMETER
1163 04C7 CD A6 04      CALL   OUTPAR      ;OUTPUT PARAMETER
1164 04CA 23      INC     HL
1165 04CB 05      DEC     B
1166 04CC C2 C6 04      JF     NZ,SENPAR      ;LOOP UNTIL END 0
F PARAMETER-LIST
1167 04CF C9      RET
1168                                     ;
1169                                     ;
1170                                     ;
1171                                     ;
1172                                     ;

```

```

1176
*****
1177
1178
1179
1180
*****
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
RCH
1201
RCH
1202
1203
ORCH
1204
RCH
1205
1206
ORDS/ROW
1207
1208
1209
1210
1211 04D0
1212
1213
1214
1215
1216
1217
====
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228 0007
1229 0006
RCH
1230 000A
RCH
1231 000B
1232 000A
ORCH
1233 000F
RCH
1234 0280
1235 0050
ORDS/ROW
1236 0190
1237

```

```

*****
;
;
; GDC DATA ARRAY
;
*****
;
; IF FARBE-1
;
;
; INITIALIZE PARAMETER OF GDC
; =====
;
; WCLK=2.5 MHz          FIXCLK = 20 MHZ
; TWORD= 400 NS        TDOT  = 50 US
;
; T-LINE = 42.00 US (F=23,80 KHZ)
; T-FRAME= 17808 US (F=56,20 HZ)
; TDISPH = 32,00 US
; TDISPV = 16,8 MS
; T-HBLANK = 10,00 US ( GDC )
; T-VBANLK = 1008 US
;
;
; HSYNC EQU 3 ;1,20 US : HORIZONT. SYNC
; HFP EQU 3 ;1,20 US : " -FRONT-PO
;
; HBP EQU 19 ;7,60 US : " -BACK -PO
;
; VSYNC EQU 12 ;504 US : VERTICAL-SYNC
; VFP EQU 6 ;252 US : " -FRONT-F
;
; VBP EQU 6 ;252 US : " -BACK-PO
;
; PIXEL EQU 640 ; PIXEL/ROW
; ADWPR EQU PIXEL/8 ;32 US : ACTIVE DISPL. W
;
; AL EQU 400 ;16,8 MS : " LINES
;
; ENDIF
;
; IF FARBE
;
; INITIALIZE PARAMETER OF GDC COLOR
; =====
;
;
; WCLK=2.5 MHz          FIXCLK = 20 MHZ
; TWORD= 400 NS        TDOT  = 50 US
;
;
; T-LINE = 41.20 US (F=24,27 KHZ)
; T-FRAME= 17840 US (F=56,06 HZ)
; T-HBLANK = 10,00 US ( GDC )
; T-VBANLK = 1008 US
;
;
; HSYNC EQU 7 ;2,80 US : HORIZONT. SYNC
; HFP EQU 6 ;2,40 US : " -FRONT-PO
;
; HBP EQU 10 ;4,00 US : " -BACK -PO
;
; VSYNC EQU 8 ;330 US : VERTICAL-SYNC
; VFP EQU 10 ;412 US : " -FRONT-F
;
; VBP EQU 15 ;618 US : " -BACK-PO
;
; PIXEL EQU 640 ; PIXEL/ROW
; ADWPR EQU PIXEL/8 ;32 US : ACTIVE DISPL. W
;
; AL EQU 400 ;16,8 MS : " LINES
;
;

```

```

1238                                     ;          ENDIF
1239                                     ;
1240                                     ;
1241                                     ;
1242 04D0 14                               ;INTPAR: DEFB  RESMOP+MIXGAC+NOINTL+DRWRET+DYNRA
M
1243 04D1 4E                               DEFB  ADWPR-2
1244 04D2 06                               DEFB  (VSYNC.AND.00000111B).SHL.5.OR.(H
SYNC-1)
1245 04D3 15                               DEFB  (HFP-1).SHL.2.OR.(VSYNC.SHR.3)
1246 04D4 09                               DEFB  HBP-1
1247 04D5 0A                               DEFB  VFP
1248 04D6 90                               DEFB  .LOW.AL
1249 04D7 3D                               DEFB  VBP.SHL.2.OR.(.HIGH.AL)
1250                                     ;
1251                                     ;
1252 04D8 00 00                             ;PRMPAR: DEFW  0                ;START ADDRESS OF
D-PARTITION 1
1253 04DA F0                               DEFB  0F0H                ;LENGTH OF DP 1
1254 04DB 3F                               DEFB  3FH                ;PART IS GRAPHIC
AREA
1255 04DC 00 00                             DEFW  0                ;DP 2
1256                                     ;

```

```

1261
*****
1262
1263
1264
1265
*****
1266
1267
1268
1269
1270
1271
1272
1273
1274 04DE 3E 52
OT (MODE 1)
1275 04E0 D3 83
1276 04E2 3E 20
1277 04E4 D3 81
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289 04E6 3E 3A
RE TRIGGER STROB (MODE 5)
1290 04E8 D3 83
1291 04EA 3E 48
1292 04EC D3 80
20 HZ )
1293 04EE 3E 03
1294 04F0 D3 80
1295 04F2 16 03
1296 04F4 CD 95 04
1297 04F7 C3 FA 04
1298

;*****
;
;     *** INIT TIMER B253 ***
;
;*****
;
;
;     *** TIMER 1 ***
;     *****
;
;
; INITTI: LD     A,01010010B     ;COUNTER 1 ONE-SH
;
;     OUT     (TIM0),A
;     LD      A,32
;     OUT     (TIM1),A         ;MONO = 12,8 US
;
;
;
;     *** TIMER 2 ***
;     *****
;
;
;     *** TIMER 0 ***
;     *****
;
;
; LD      A,00111010B     ;COUNTER 0 HARDWA
;
;     OUT     (TIM0),A
;     LD      A,046H
;     OUT     (TIM0),A         ;MONO FOR 15 SEC
;                               ;CLOCK=VSYNC (56,
;
;
; LD      A,03H
;     OUT     (TIM0),A
;     LD      D,3
;                               ;WAIT 300MS
;     CALL   TIM100
;     JP     NEXT5
;
;

```

```

1301 ;*****
*****
1302 ;
1303 ; TEST OF DISK-CONTROLLER 8272
1304 ;
1305 ;*****
*****
1306 ;
1307 ;
1308 04FA 3E DE NEXT5: LD A,FDCERR
1309 04FC D3 00 OUT (ERRDIS),A
1310 ;
1311 04FE DB 50 FLEXTS IN A,(RDMST) ;GET MAIN-STATUS
1312 0500 FE 80 CF 1000000B ;MUST BE 80H
1313 0502 CA 08 05 JP Z,LL140
1314 0505 C3 14 06 JP ENDLO
1315 ;
1316 0508 21 17 FB ;
1317 050B 36 03 LL140 LD HL,COMBLD
1318 050D 23 LD (HL),FDCSPE
1319 050E 36 D6 INC HL
LD (HL),11010110B ;SRT= 6MS / HULT=
200MS
1320 0510 23 INC HL
1321 0511 36 2C LD (HL),00101100B ;HLT= 88MS DMA
1322 0513 06 03 LD B,03H
1323 0515 CD 01 09 CALL SENBY1 ;SEND BYTES TO FD
C
1324 0518 C3 DE 05 JP TST16
1325
1326

```

```

1331
*****
1332
1333
) ***
1334
1335
*****
1336
1337 FE00
1338 FE01
1339 FE02
1340
1341 FE04
1342 FE05
1343 FE06
1344
1345
1346 051B CD A7 03
1347 051E 3E 01
1348 0520 32 01 F8
1349 0523 AF
1350 0524 32 00 F8
1351 0527 21 96 05
1352 052A CD 5E 0A
1353
1354 052D CD 13 01
1355 0530 F5
1356 0531 CD 24 01
1357 0534 3E FF
1358 0536 32 05 FE
1359 0539 F1
1360
1361 053A 32 04 FE
1362 053D FE 30
1363 053F CA 5E 05
1364 0542 FE 31
1365 0544 CA 7A 05
1366 0547 FE 32
1367 0549 CA 7D 05
1368 054C FE 33
1369 054E CA 5E 05
1370 0551 FE 34
1371 0553 CA AC 0A
1372
1373 0556 3E 0C
1374 0558 CD 24 01
1375 055B C3 1E 05
1376
1377 055E D3 D0
1378 0560 00
1379 0561 3A 05 FE
1380 0564 A7
1381 0565 C2 71 05
1382
1383 0568 21 B6 05
1384 056B CD 5E 0A
1385 056E C3 1E 05
1386
1387 0571 21 D2 05
1388 0574 CD 5E 0A
1389 0577 C3 1E 05
1390
1391 057A C3 80 05
;*****
;
;   *** LANGUAGE BOARD TEST (16 BIT PROCESSOR)
;
;*****
;
;   CBB   EQU   OFE00H
;   SB8   EQU   OFE01H
;   ADD8  EQU   OFE02H
;
;   CB16  EQU   OFE04H
;   SB16  EQU   OFE05H
;   ADD16 EQU   OFE06H
;
;
;DIAG16: CALL   INVAL1
DIA16:  LD     A,01
        LD     (CURSY),A
        XOR   A
        LD     (CURSX),A
        LD     HL,PHMES
        CALL  DISPL
;
;   CALL  KEYINP
;   PUSH AF
;   CALL  MODCRT
;   LD    A,0FFH
;   LD    (SB16),A
;   POP  AF
;
;   LD    (CB16),A
;   CP   '0'
;   JP   Z,PHASE0
;   CP   '1'
;   JP   Z,PHASE1
;   CP   '2'
;   JP   Z,PHASE2
;   CP   '3'
;   JP   Z,PHASE0
;   CP   '4'
;   JP   Z,MONI
;
;DIAGE: LD     A,0CH
        CALL  MODCRT
        JP   DIA16
;
;PHASE0: OUT   (PROCCH),A
        NOP
        LD    A,(SB16)
        AND  A
        JP   NZ,BRUN
;
;GRUN:  LD     HL,GODDST
        CALL  DISPL
        JP   DIA16
;
;BRUN:  LD     HL,BADSTA
        CALL  DISPL
        JP   DIA16
;
;PHASE1: JP   PRJUMP

```



```

1392                                     ;
1393 057D C3 70 06                     PHASE2: JP      STARTS
1394                                     ;
1395                                     ;
1396                                     ;
1397                                     ;
1398                                     ;
1399                                     ;
1400                                     ;
RE ***                                  ; *** PROGRAM FOR SWITCH CONTROL TO 16 BIT SOFTWA
1401                                     ;
**                                     ; *****
1402                                     ;
1403 F10C                               RAMTOP EQU    OF10CH
1404                                     ;
1405 0580 11 0C F1                     PRJUMP: LD     DE,RAMTOP
1406 0583 21 8E 05                     LD     HL,PROGRA
1407 0586 01 08 00                     LD     BC,8
1408 0589 ED 80                         LDIR
1409 058B C3 0C F1                     JP     RAMTOP
1410                                     ;
1411 058E D3 10                         PROGRA: OUT   (RAMSEL),A
1412 0590 D3 D0                         OUT   (PROCCH),A
1413 0592 00                             NOP                                     ;CONTROL TO 16 BI
T
1414                                     ;
1415 0593 C3 11 F1                     JP     RAMTOP+5                       ;ENTRY POINT' 8 BI
T I/O
1416                                     ;
1417                                     ;
1418                                     ;
1419 0596 1F                             PHMES  DEFB   31,'SELECT THE SWITCH PHASE (0-2)

0597 53
0598 45
0599 4C
059A 45
059B 43
059C 54
059D 20
059E 54
059F 48
05A0 45
05A1 20
05A2 53
05A3 57
05A4 49
05A5 54
05A6 43
05A7 48
05A8 20
05A9 50
05AA 48
05AB 41
05AC 53
05AD 45
05AE 20
05AF 28
05B0 30
05B1 2D
05B2 32
05B3 29
05B4 20
05B5 20
1420 05B6 18                             GOODST DEFB   27,'SUCCESSFUL PROCESSOR SWITCH'
05B7 53
05B8 55
05B9 43
05BA 43

```

	05BE	45			
	05BC	53			
	05BD	53			
	05BE	46			
	05BF	55			
	05C0	4C			
	05C1	20			
	05C2	50			
	05C3	52			
	05C4	4F			
	05C5	43			
	05C6	45			
	05C7	53			
	05C8	53			
	05C9	4F			
	05CA	52			
	05CB	20			
	05CC	53			
	05CD	57			
	05CE	49			
	05CF	54			
	05D0	43			
	05D1	48			
1421	05D2	08	BADSTA	DEFB	11, 'BAD BAD BAD
	05D3	42			
	05D4	41			
	05D5	44			
	05D6	20			
	05D7	42			
	05D8	41			
	05D9	44			
	05DA	20			
	05DB	42			
	05DC	41			
	05DD	44			
1422					!
1423					!

```

1427 ;*****
*****
1428 ;
1429 ; TEST 808B
1430 ;
1431 ;*****
*****
1432 ;
1433 ;
1434 05DE DB 13 TST16: IN A, (SYSSTA)
1435 05E0 E6 02 AND Z
1436 05E2 3E FF LD A, OFFH
1437 05E4 C2 06 06 JP NZ, TSTEND
1438 ;
1439 05E7 3E EE LD A, P16ERR
1440 05E9 D3 00 OUT (ERRDIS), A
1441 ;
1442 05EB 3E FF LD A, OFFH
1443 05ED 32 05 FE LD (SB16), A
1444 05F0 3E 00 LD A, 0 ; (SBB)=9 THEN RAM
-TEST TILL 512K (REDESIGN-BOARD)
1445 05F2 32 01 FE LD (SBB), A
1446 05F5 3E 33 LD A, '3 ; LEVEL 0 TEST PHA
SE 16 BIT
1447 05F7 32 04 FE LD (CB16), A
1448 05FA D3 D0 OUT (PROCCH), A ; SWITCH TO 16 BIT
1449 05FC 3A 05 FE LD A, (SB16)
1450 05FF 32 01 FE LD (SBB), A ; STORE STATUSFLAG
1451 0602 A7 AND A ; CHECK STATUS
1452 0603 C2 0D 06 JP NZ, ENDL02 ; JMP IF NOT 0
1453 0606 3E FF TSTEND: LD A, DIACOM
1454 0608 D3 00 OUT (ERRDIS), A
1455 060A C3 70 06 JP STARTS
1456 ;
1457 ;
1458 ;
1459 060D 3E EF ENDL02: LD A, 0EFH ; SWITCH DIAG-LED
OFF
1460 060F D3 00 OUT (ERRDIS), A ; LED-8
1461 0611 C3 70 06 JP STARTS
1462 ;
1463 ;
1464 ;*****
*****
1465 ;
1466 0614 18 FE ENDL0: JR $ ; LEVEL0-ERROR DET
ECTED
1467 ;
1468 ;*****
*****
1469 ;

```

```

1471
1472
1473
1474
1475
1476
1477
1478 0615 21 06 F8
1479 0619 36 07
1480 061B 23
1481 061C 23
1482 061D 23
1483 061E 77
1484 061F 23
1485 0620 36 00
1486 0622 23
1487 0623 36 01
1488 0625 21 06 F8
1489 0628 11 10 F8
1490 062B CD AB 07
1491 062E 21 10 F8
1492 0631 7E
1493 0632 1F
1494 0633 1F
1495 0634 1F
1496 0635 1F
1497 0636 D2 53 06
1498 0639 3E 18
1499 063B 32 01 F8
1500 063E AF
1501 063F 32 00 F8
1502 0642 21 59 06
1503 0645 CD 5E 0A
1504 0648 CD 13 01
1505 064B FE 88
1506 064D C2 48 06
1507 0650 C3 25 06
1508 0653 3E 0C
1509 0655 CD 24 01
1510 0658 C9
1511
1512
1513 0659 16
      065A 44
      065B 49
      065C 53
      065D 4B
      065E 20
      065F 41
      0660 3A
      0661 20
      0662 4E
      0663 4F
      0664 54
      0665 20
      0666 52
      0667 45
      0668 41
      0669 44
      066A 59
      066B 20
      066C 28
      066D 43
      066E 52
      066F 29

```

```

:
:
:
: *** RESET DISK-DRIVE 0 ***
:
:
REST:  LD      HL,PARAME
      LD      (HL),FDCRE
      INC    HL
      INC    HL
      INC    HL
      LD      (HL),A          ;UNIT = 0
      INC    HL
      LD      (HL),000H      ;TRACK = 0
      INC    HL
      LD      (HL),001H      ;SECTOR = 1
LL141  LD      HL,PARAME
      LD      DE,RESULT
      CALL   FDCDR1 ;RESTORE
      LD      HL,RESULT
      LD      A,(HL)          ;GET RESULT
      RRA
      RRA
      RRA
      RRA
      JP      NC,REST3      ;GOOD STATUS
      LD      A,18H
      LD      (CURSY),A
      XDR    A
      LD      (CURSX),A
      LD      HL,NOTRM
      CALL   DISPL
REST2: CALL   KEYINP
      CP
      JP      NZ,REST2
      JF
LL141
REST3: LD      A,SCRER
      CALL   MODCRT
      RET
:
:
:
NOTRM: DEFB   22,'DISK A: NOT READY (CR)'

```

```

1517
1518
1519 0670 FB
1520 0671 CD 1B 01
1521 0674 FE 0D
1522 0676 CA A5 0A
1523 0679 3E 00
1524 067B CD 16 06
T 0
1525
1526
1527
*****
1528
1529
1530
1531
*****
1532
1533
1534 067E 21 06 FB
AD A TRACK)
1535 0681 36 42
1536 0683 23
1537 0684 23
1538 0685 23
1539 0686 36 00
1540 0688 23
1541 0689 36 00
1542 068B 23
1543 068C 36 01
1544 068E 23
1545 068F 36 00
1546 0691 23
1547 0692 36 40
1548 0694 21 00 20
1549 0697 22 0E FB
H
1550
1551
1552
OOH ***
1553
1554
1555 069A AF
1556 069B 32 20 FB
R
1557
1558 069E 21 06 FB
1559 06A1 11 10 FB
1560 06A4 CD AB 07
1561 06A7 21 10 FB
1562 06AA 7E
1563 06AB E6 C0
1564 06AD CA CD 06
1565 06B0 23
1566 06B1 7E
1567 06B2 E6 24
1568 06B4 C2 BD 06
OR
1569 06B7 CD FF 09
SSAGE
1570 06BA C3 9E 06
1571
1572 06BD 21 20 FB
1573 06C0 34
1574 06C1 7E
1575 06C2 FE 0A
1576 06C4 C2 9E 06
1577 06C7 CD FF 09
SSAGE
1578 06CA C3 9A 06
1579
1580 06CD 2A 0E FB
;
;
; STARTS: EI
; CALL KYLOOK
; CP ODH ; "AM"
; JP Z,MONIT
; LD A,0
; CALL REST ;RESTORE DISK UNI
;
;
; *****
;
; SYSTEM LOADER
;
; *****
;
;
; SLOAD LD HL,PARAME ;SET FUNCTION (RE
; LD (HL),FRTRK
; INC HL
; INC HL
; INC HL
; LD (HL),000H ;UNIT = 0
; INC HL
; LD (HL),000H ;TRACK = 0
; INC HL
; LD (HL),001H ;SECTOR = 1
; INC HL
; LD (HL),000H ;HEAD = 0
; INC HL
; LD (HL),40H ;DENSITY = DOUBLE
; LD HL,2000H
; LD (PARAME+8),HL ;DMA ADDRESS=2000
;
;
;
; *** LOAD 12K FROM TRACK 0 SECTOR 1 TO MEMORY 10
;
;
; SLOAD5 XOR A
; LD (RETRY),A ;SET RETRY-COUNTE
;
;
; SLOAD6 LD HL,PARAME
; LD DE,RESULT
; CALL FDCDR1 ;READ TRACK
; LD HL,RESULT
; LD A,(HL)
; AND 0C0H
; JF Z,SLOAD7 ;JUMP IF OK
; INC HL
; LD A,(HL)
; AND 024H
; JF NZ,SLOAD8 ;JUMP IF DATA-ERR
; CALL DISSTA ;DISPLAY ERROR-ME
;
; SLOAD8 LD HL,RETRY
; INC (HL)
; LD A,(HL)
; CP 00AH
; JF NZ,SLOAD6 ;RETRY
; CALL DISSTA ;DISPLAY ERROR-ME
;
;
; JF SLOAD5 ;RETRY
;
; SLOAD7 LD HL,(PARAME+8)

```

1581 06D0 11 00 10		LD	DE,1000H	
1582				
1583 06D3 19	:	SLOAD9	ADD HL,DE	;UPDATED LOAD-ADD
RESS				
1584 06D4 7C		LD	A,H	
1585 06D5 FE 50		CP	050H	
1586 06D7 CA E4 06		JP	Z,SLOADA	;JUMP TO EXECUTE
PROGRAM				
1587 06DA 22 0E F8		LD	(PARAM+8),HL	;SET NEW DMA-ADDR
ESS				
1588 06DD 21 0A F8		LD	HL,PARAM+4	
1589 06E0 34		INC	(HL)	;TRACK# + 1
1590 06E1 C3 9A 06		JP	SLOAD5	;READ NEXT TRACK
1591	:			
1592 06E4 11 93 07		SLOADA:	LD DE,IDENT1	
1593 06E7 CD 3D 07		CALL	CHECKI	
1594 06EA CA 00 20		JP	Z,2000H	;8-BIT-SYSTEM (CP
M 2.2)				
1595 06ED 11 A1 07		LD	DE,IDENT3	
1596 06F0 CD 3D 07		CALL	CHECKI	
1597 06F3 CA 00 20		JP	Z,2000H	;8-BIT-SYSTEM (CP
M 3.0)				
1598 06F6 11 9A 07		LD	DE,IDENT2	
1599 06F9 CD 3D 07		CALL	CHECKI	
1600 06FC CA 02 07		JP	Z,SLOADB	;16-BIT-SYSTEM
1601 06FF C3 1E 07		JP	SLOADD	;OTHER SOFTWARE
1602	:	SLOADB:	IN A,(SYSSTA)	
1603 0702 DB 13		AND	Z	
1604 0704 E6 02		JP	NZ,N016	;JMP IF NO 16 BIT
1605 0706 C2 18 07		LD	A,(SBB)	
1606 0709 3A 01 FE		AND	A	
1607 070C A7		JP	NZ,N016	;16 BIT PROCESSOR
1608 070D C2 18 07				
ERROR				
1609	:			
1610 0710 3E 32		LD	A,"2"	
1611 0712 32 04 FE		LD	(CB16),A	;COMMAND EXEC SOF
TWARE				
1612 0715 C3 80 05		JP	PRJUMP	;CONTOL TO 16 BIT
1613	:			
1614 0718 21 62 07		N016:	LD HL,MESN16	
1615 071B C3 21 07		JP	SL2	
1616	:			
1617 071E		SLOADC:		
1618 071E 21 4A 07		SLOADD:	LD HL,SLMESS	
1619 0721 3E 18		SL2:	LD A,18H	
1620 0723 32 01 F8		LD	(CURSY),A	
1621 0726 AF		XOR	A	
1622 0727 32 00 F8		LD	(CURSX),A	
1623 072A CD 5E 0A		CALL	DISPL	
1624 072D CD 13 01		SL3:	CALL KEYINP	
1625 0730 FE 88		CP	NEWL	
1626 0732 C2 2D 07		JP	NZ,SL3	
1627 0735 3E 0C		LD	A,SCRER	
1628 0737 CD 24 01		CALL	MODCRT	
1629 073A C3 70 06		JP	STARTS	
1630	:			
1631 073D 21 03 20		CHECKI:	LD HL,2003H	
1632 0740 06 07		LD	B,7	
1633	:			
1634 0742 1A		CHECKI:	LD A,(DE)	
1635 0743 BE		CP	(HL)	
1636 0744 C0		RET	NZ	
1637 0745 13		INC	DE	
1638 0746 23		INC	HL	
1639 0747 10 F9		DJNZ	CHECKI	
1640 0749 C9		RET		
1641	:			
1642	:			
1643	:			

1644	074A 17	SLMESS	DEFB	23, A: MOUNT O.S. DISK (CR)
	074B 41			
	074C 3A			
	074D 20			
	074E 4D			
	074F 4F			
	0750 55			
	0751 4E			
	0752 54			
	0753 20			
	0754 4F			
	0755 2E			
	0756 53			
	0757 2E			
	0758 20			
	0759 44			
	075A 49			
	075B 53			
	075C 4B			
	075D 20			
	075E 2B			
	075F 43			
	0760 52			
	0761 29			
1645	0762 30	MESN16	DEFB	48, 16 BIT NOT IMPLEMENTED,
	0763 31			
	0764 36			
	0765 20			
	0766 42			
	0767 49			
	0768 54			
	0769 20			
	076A 4E			
	076B 4F			
	076C 54			
	076D 20			
	076E 49			
	076F 4D			
	0770 50			
	0771 4C			
	0772 45			
	0773 4D			
	0774 45			
	0775 4E			
	0776 54			
	0777 45			
	0778 44			
	0779 2C			
	077A 20			
1646	077B 54	DEFB		TRY 8 BIT O.S. DISK (CR)
	077C 52			
	077D 59			
	077E 20			
	077F 38			
	0780 20			
	0781 42			
	0782 49			
	0783 54			
	0784 20			
	0785 4F			
	0786 2E			
	0787 53			
	0788 2E			
	0789 20			
	078A 44			

	078E	49			
	078C	53			
	078D	4B			
	078E	20			
	078F	2B			
	0790	43			
	0791	52			
	0792	29			
1647	0793	43	IDENT1	DEFB	CFM 2.2
	0794	50			
	0795	4D			
	0796	20			
	0797	32			
	0798	2E			
	0799	32			
1648	079A	31	IDENT2	DEFB	16BIT
	079B	36			
	079C	42			
	079D	49			
	079E	54			
	079F	20			
	07A0	20			
1649	07A1	43	IDENT3	DEFB	CFM 3.0
	07A2	50			
	07A3	4D			
	07A4	20			
	07A5	33			
	07A6	2E			
	07A7	30			


```

1655
*****
1656
1657
1658
1659
*****
1660
1661
1662 07A8 22 02 FB
LOCK-ADDRESS
1663 07AB EB
1664 07AC 22 04 FB
K-ADDRESS
1665 07AF EB
1666 07B0 7E
1667 07B1 FE 45
1668 07B3 CA DF 07
1669 07B6 FE 46
1670 07B8 CA F0 07
1671 07BB FE 07
1672 07BD CA 01 08
1673 07C0 FE 4A
1674 07C2 CA 4C 08
1675 07C5 FE 4D
1676 07C7 CA 6E 08
1677 07CA FE 04
1678 07CC CA CB 08
1679 07CF FE 42
1680 07D1 CA 9F 08
1681 07D4 23
1682 07D5 23
1683 07D6 23
1684 07D7 7E
1685 07D8 F6 80
1686 07DA 2A 04 FB
1687 07DD 77
1688 07DE C9
;*****
;
; DECIDE WHAT FUNCTION IS TO DO
;
;*****
;
;
;
FDCDR1 LD (PARADR),HL ;SAVE PARAMETER-B
;
LD (RESADR),HL ;SAVE RESULT-BLOC
;
EX DE,HL
LD A,(HL) ;GET FUNCTION
CF FDCWD
JP Z,WRDAT1 ;WRITE DATA
CF FDCRD
JP Z,RDDAT1 ;READ DATA
CF FDCRE
JP Z,RECAL1 ;RECALIBRATE
CF FDCRID
JP Z,RDSID1 ;READ ID
CF FDCFT
JP Z,WRFOR1 ;FORMAT A TRACK
CF FDCSEN
JP Z,DRIVES ;GET SENSE-STATUS
CF FRTRK
JP Z,READTR ;READ A TRACK
HL
INC HL
INC HL
INC HL
LD A,(HL) ;GET UNIT
OR 080H
LD HL,(RESADR)
LD (HL),A ;INVALID COMMAND
RET

```

```

1694
*****
1695
1696
1697
1698
*****
1699
1700
1701
1702
1703
1704
1705 07DF CD CC 08
1706 07E2 CD 92 09
1707 07E5 06 09
1708 07E7 CD 11 09
C
1709 07EA 06 07
1710 07EC CD 2B 09
1711 07EF C9
1712
1716
*****
1717
1718
1719
1720
*****
1721
1722
1723
1724
1725 07F0 CD CC 08
1726 07F3 CD 92 09
1727 07F6 06 09
1728 07F8 CD 11 09
C
1729 07FB 06 07
1730 07FD CD 2B 09
1731 0800 C9
1732

;*****
;
; WRITE DATA
;
;*****
;
;
; *** FUNCTION/DENSITY ***
; *****
;
;
;
WRDAT1: CALL FUBLK
        CALL DMA1 ;ACTIVATE DMA
        LD B,009H
        CALL SENDB1 ;SEND BYTES TO FD
        LD B,007H
        CALL GETBY1 ;GET STATUS-BYTES
        RET
;
;*****
;
; READ DATA
;
;*****
;
;
;
RDDAT1: CALL FUBLK
        CALL DMA1 ;ACTIVATE DMA
        LD B,009H
        CALL SENDB1 ;SEND BYTES TO FD
        LD B,007H
        CALL GETBY1 ;GET STATUS-BYTES
        RET
;

```

```

1736
*****
1737
1738
1739
1740
*****
1741
1742
1743 0801 CD 38 09
1744
1745 0804 CD 71 09
1746 0807 3E 0F
1747 0809 D3 51
1748 080B 11 03 00
1749 080E 2A 02 FB
1750 0811 19
1751 0812 CD 71 09
1752 0815 7E
1753 0816 D3 51
1754 0818 CD 71 09
1755 081B 3E 04
1756 081D D3 51
1757 081F CD 87 09
1758
1759 0822 3E 08
RRUPT-STATUS
1760 0824 D3 51
1761 0826 06 02
1762 0828 CD 2B 09
1763
1764 082B CD 71 09
1765 082E 3E 07
ATE-COMMAND
1766 0830 D3 51
1767 0832 11 03 00
1768 0835 2A 02 FB
1769 0838 19
1770 0839 CD 71 09
1771 083C 7E
1772 083D D3 51
1773 083F CD 87 09
1774
1775 0842 3E 08
1776 0844 D3 51
STATUS-COMMAND
1777 0846 06 02
1778 0848 CD 2B 09
1779 084B C9
1780

```

```

:*****
:
: RECALIBRATE
:
:*****
:
:
: RECAL1: CALL DMON ;SWITCH DISC-MOTOR ON
:
: CALL FDCRD1
: LD A,FDCSEE ;SEEK-COMMAND
: OUT (FDCWR),A
: LD DE,UFARAM
: LD HL,(PARADR)
: ADD HL,DE
: CALL FDCRD1
: LD A,(HL) ;GET UNIT
: OUT (FDCWR),A
: CALL FDCRD1
: LD A,004H ;TRACK
: OUT (FDCWR),A
: CALL FDCINI
:
: LD A,FDCSIS ;SENSINTE
: OUT (FDCWR),A
: LD B,2
: CALL GETBY1
:
: CALL FDCRD1
: LD A,FDCRE ;RECALIBR
: OUT (FDCWR),A
: LD DE,UFARAM
: LD HL,(PARADR)
: ADD HL,DE
: CALL FDCRD1
: LD A,(HL) ;GET UNIT
: OUT (FDCWR),A
: CALL FDCINI
:
: LD A,FDCSIS
: OUT (FDCWR),A ;SENSE INTERRUPT-
: LD B,002H
: CALL GETBY1
: RET
:

```

```

1784
*****
1785
1786
1787
1788
*****
1789
1790
1791
1792
1793
1794
1795 084C
1796 084C FD 24 02 FB
1797 0850 21 17 FB
1798 0853 FD 7E 00
1799 0856 77
1800
1801
1802
1803
1804
1805
1806 0857 23
1807 0858 FD 46 03
1808 085B FD 7E 06
1809 085E B0
T
1810 085F 77
1811 0860 CD 44 09
1812 0863 06 02
1813 0865 CD 01 09
1814 0868 06 07
1815 086A CD 2B 09
1816 086D C9
1817

;*****
;
; READ ID
;
;*****
;
;
; *** FUNCTION/DENSITY ***
; *****
;
;
; RDSID1:
; LD IY,(PARADR)
; LD HL,COMBLO
; LD A,(IY+0) ;GET FUNCTION
; LD (HL),A ;FILL COMBLO
;
;
; *** UNIT/HEAD ***
; *****
;
;
; INC HL ;NEXT COMBLO
; LD B,(IY+3) ;GET UNIT
; LD A,(IY+6) ;GET HEAD
; OR B ;COMBINE WITH UNI
;
; LD (HL),A ;FILL COMBLO
; CALL SEEK1
; LD B,002H
; CALL SENBY1 ;SEND BYTES TO FDC
; LD B,007H
; CALL GETBY1 ;GET STATUS-BYTES
; RET
;

```

```

1821 ;*****
*****
1822 ;
1823 ;   FORMAT A TRACK
1824 ;
1825 ;*****
*****
1826 ;
1827 ;
1828 ;   *** FUNCTION/DENSITY ***
1829 ;   *****
1830 ;
1831 ;
1832 086E WRFOR1:
1833 086E FD 2A 02 FB LD IY,(PARADR)
1834 0872 21 17 FB LD HL,COMBLO
1835 0875 FD 7E 00 LD A,(IY+0) ;GET FUNC
TION
1836 0878 77 LD (HL),A ;FILL COM
BLO
1837 ;
1838 ;
1839 ;   *** UNIT/HEAD ***
1840 ;   *****
1841 ;
1842 ;
1843 0879 23 INC HL ;NEXT COMBLO
1844 087A FD 46 03 LD B,(IY+3) ;GET UNIT
1845 087D FD 7E 06 LD A,(IY+6) ;GET HEAD
1846 0880 80 OR B ;COMBINE WITH UNI
T
1847 0881 77 LD (HL),A ;FILL COMBLO
1848 ;
1849 ;
1850 ;   *** N ***
1851 ;   *
1852 ;
1853 ;
1854 0882 23 INC HL ;NEXT COMBLO
1855 0883 36 02 LD (HL),002H ;FILL COMBLO FOR
DOUBLE
1856 ;
1857 ;
1858 ;   *** SC ***
1859 ;   **
1860 ;
1861 ;
1862 0885 23 INC HL ;NEXT COMBLO
1863 0886 36 08 LD (HL),FDCE0T ;FILL COMBLO
1864 ;
1865 ;
1866 ;   *** GPL ***
1867 ;   ***
1868 ;
1869 ;
1870 0888 23 INC HL ;NEXT COMBLO
1871 0889 36 50 LD (HL),050H ;FILL COMBLO FOR
DOUBLE
1872 ;
1873 ;
1874 ;   *** D ***
1875 ;
1876 ;
1877 088B 23 INC HL ;NEXT COMBLO
1878 088C 36 E5 LD (HL),0E5H ;FILL COMBLO
1879 088E CD 44 09 CALL SEEK1
1880 0891 CD 92 09 CALL DMA1 ;ACTIVÁTE DMA
1881 0894 06 06 LD B,006H
1882 0896 CD 11 09 CALL SENDB1 ;SEND BYTES TO FD
C
1883 0899 06 07 LD B,007H
1884 089B CD 2B 09 CALL GETBY1 ;GET STATUS-BYTES
1885 089E C9 RET
1886

```

```

1890
*****
1891
1892
1893
1894
*****
1895
1896
1897
1898
1899 089F CD CC 08
1900
1901
1902
1903
1904
1905
1906 08A2 2A 02 FB
1907 08A5 3E 47
1908 08A7 D3 2B
1909 08A9 11 08 00
1910 08AC 19
1911 08AD 7E
1912 08AE D3 26
1913 08B0 23
1914 08B1 7E
1915 08B2 D3 26
1916 08B4 3E FF
1917 08B6 D3 27
1918 08B8 3E 0F
1919 08BA D3 27
1920 08BC 3E 03
1921 08BE D3 2A
1922 08C0 06 09
1923 08C2 CD 11 09
C
1924 08C5 06 07
1925 08C7 CD 2B 09
DC
1926 08CA C9
1927
1928
1929
*****
1930
1931
1932
1933
*****
1934
1935
1936 08CB C9
1937

```

```

;*****
;
; READ A TRACK
;
;*****
;
;
;
; READTR: CALL FUBLK
;
;
; *** ACTIVATE DMA ***
; *****
;
;
LD HL,(PARADR)
LD A,01000111B ;WRITE MEMORY
OUT (DMAM0),A ;OUT MODE
LD DE,ALFARA
ADD HL,DE
LD A,(HL)
OUT (C3AD),A
INC HL
LD A,(HL)
OUT (C3AD),A
LD A,OFFH
OUT (C3TC),A
LD A,OFH
OUT (C3TC),A
LD A,3
OUT (DMAMB),A ;ENABLE CH 0
LD B,09
CALL SENDB1 ;SEND BYTES TO FD
LD B,07
CALL GETBY1 ;GET BYTES FROM F
RET
;
;
;*****
; SENSE DRIVE STATUS
;
;*****
;
; DRIVES RET
;

```

```

1941
*****
1942
1943
1944
1945
*****
1946
1947
1948
1949
1950 0BCC FD 2A 02 F8
1951 0BD0 21 17 F8
1952 0BD3 FD 7E 00
1953 0BD6 77
1954
1955
1956
1957
1958
1959 0BD7 23
1960 0BD8 FD 46 03
1961 0BD8 FD 7E 06
1962 0BDE 80
T 1963 0BDF 77
1964
1965
1966
1967
1968
1969
1970 0BE0 23
1971 0BE1 FD 7E 04
1972 0BE4 77
1973
1974
1975
1976
1977
1978
1979 0BE5 23
1980 0BE6 FD 7E 06
1981 0BE7 0F
1982 0BEA 0F
1983 0BEB 77
1984
1985
1986
1987
1988
1989
1990 0BEC 23
1991 0BED FD 7E 05
1992 0BF0 77
1993
1994
1995
1996
1997
1998 0BF1 23
1999 0BF2 36 02
DOUBLE
2000
2001
2002
; *****
;
; FUNCTION-BLOCK
;
; *****
;
;
; PREPARE FUNCTION-BLOCK FOR FDC (8272)
;
FUBLK: LD IY, (PARADR)
LD HL, COMBLO
LD A, (IY+0) ;GET FUNCTION
LD (HL), A ;FILL COMBLO
;
; *** UNIT/HEAD ***
; *****
;
;
; INC HL ;NEXT COMBLO-ADR.
LD B, (IY+3) ;GET UNIT
LD A, (IY+6) ;GET HEAD
OR B ;COMBINE WITH UNI
;
LD (HL), A ;FILL COMBLO
;
;
; *** C ***
; *
;
;
; INC HL ;NEXT COMBLO-ADR.
LD A, (IY+4) ;GET CYLINDER
LD (HL), A ;FILL COMBLO
;
;
; *** H ***
; *
;
;
; INC HL ;NEXT COMBLO-ADR.
LD A, (IY+6) ;GET HEAD
RRCA
LD (HL), A ;FILL COMBLO
;
;
; *** R ***
; *
;
;
; INC HL ;NEXT COMBLO-ADR.
LD A, (IY+5) ;GET SECTOR
LD (HL), A ;FILL COMBLO
;
;
; *** N ***
; *
;
;
; INC HL ;NEXT COMBLO-ADR.
LD (HL), 002H ;FILL COMBLO FOR
;
;
; *** EOT ***

```

```

2003 ; ***
2004 ;
2005 ;
2006 0BF4 23 LL157 INC HL ;NEXT COMBLO-ADR.
2007 0BF5 36 08 LD (HL),FDCEOT ;FILLCOMBLO
2008 ;
2009 ;
2010 ; *** GPL/DTL ***
2011 ; *****
2012 ;
2013 ;
2014 0BF7 23 INC HL ;NEXT COMBLO-ADR.
2015 0BF8 36 2A LD (HL),02AH ;GPL FOR DOUBLE
2016 0BFA 23 INC HL
2017 0BFB 36 FF LD (HL),0FFH ;DTL FOR DOUBLE
2018 0BFD CD 44 09 CALL SEEK1 ;SEEK TRACK
2019 0900 C9 RET
2020 ;

```



```

2023 ;*****
*****
2024 ;
2025 ; SEND COMMAND-BYTES TO FDC
2026 ; B = NUMBER OF BYTES TO SEND
2027 ;
2028 ;*****
*****
2029 ;
2030 ;
2031 0901 CD 38 09 ;SENBY1: CALL DMON ;SWITCH DISK-MOTO
R ON
2032 0904 21 17 F8 LD HL,COMBLO
2033 0907 CD 71 09 LL175 CALL FDCRD1 ;WAIT UNTIL READY
2034 090A 7E LD A,(HL)
2035 090B D3 51 OUT (FDCWR),A ;SEND BYTE TO FDC
2036 090D 23 INC HL
2037 090E 10 F7 DJNZ LL175
2038 0910 C9 RET
2039 ;
2040 ;
2041 ;*****
*****
2042 ;
2043 ; SEND COMMAND-BYTES TO FDC WITH DMA REQUIRED
2044 ; B = NUMBER OF BYTES TO SEND
2045 ;
2046 ;*****
*****
2047 ;
2048 ;
2049 0911 CD 38 09 ;ENDB1: CALL DMON ;SWITCH DISK-MOTO
R ON
2050 0914 21 17 F8 LD HL,COMBLO
2051 0917 05 DEC B
2052 0918 CD 71 09 LL177 CALL FDCRD1 ;WAIT UNTIL READY
2053 091B 7E LD A,(HL)
2054 091C D3 51 OUT (FDCWR),A ;SEND BYTE TO FDC
2055 091E 23 INC HL
2056 091F 10 F7 DJNZ LL177
2057 0921 CD 71 09 CALL FDCRD1
2058 0924 7E LD A,(HL)
2059 0925 D3 51 OUT (FDCWR),A ;SEND LAST BYTE
2060 0927 CD E5 09 CALL DMA01
2061 092A C9 RET
2062

```

```

2066
*****
2067
2068
2069
2070
2071
*****
2072
2073
2074 092B 2A 04 FB
-ADR.
2075 092E CD 7C 09
2076 0931 DB 51
2077 0933 77
K
2078 0934 23
2079 0935 10 F7
2080 0937 C9
2081
2082
2083
2084
*****
2085
2086
2087
2088
*****
2089
2090
2091 0938 DB 13
2092 0934 1F
2093 093B D3 14
R ON
2094 093D D0
IS ALREADY ON
2095 093E 16 0A
2096 0940 CD 95 0A
2097 0943 C9
2098
2099
2100
2101
*****
2102
2103
2104
2105
*****
2106
2107
2108 0944 CD 71 09
2109 0947 3E 0F
2110 0949 D3 51
2111 094B 11 03 00
2112 094E 2A 02 FB
2113 0951 19
2114 0952 CD 71 09
2115 0955 7E
2116 0956 D3 51
2117 0958 23
2118 0959 CD 71 09
2119 095C 7E
2120 095D D3 51
2121 095F CD 87 09
2122 0962 3E 08
2123 0964 D3 51
2124 0966 CD 7C 09
2125 0969 DB 51
2126 096B CD 7C 09
2127 096E DB 51
2128 0970 C9
2129

```

```

;*****
;
; GET STATUS BYTES FROM FDC
; B = NUMBER OF BYTES
;
;*****
;
;
GETBY1 LD HL, (RESADR) ;GET RESULT-BLOCK
LL179 CALL FDCRD2
IN A, (FDCRA) ;READ STATUS
LD (HL), A ;SAVE RESULT-BLOC
;
;
;
;
;*****
;
; *** DISC-MOTOR-ON ***
;
;*****
;
;
DHON: IN A, (SYSSTA) ;TEST IF MOTOR ON
RRA
OUT (MON), A ;SWITCH DISK MOTO
RET NC ;RETURN IF MOTOR
LD D, 10 ;WAIT 1 SEC
CALL TIM100
RET
;
;
;*****
;*****
;
; SEEK SPECIFIED TRACK
;
;*****
;
;
SEEK1 CALL FDCRD1
LD A, FDCSEE
OUT (FDCWR), A ;SEND COMMAND
LD DE, UPARAM
LD HL, (PARAM)
ADD HL, DE
CALL FDCRD1
LD A, (HL) ;GET UNIT
OUT (FDCWR), A ;SEND UNIT
INC HL
CALL FDCRD1
LD A, (HL) ;GET TRACK
OUT (FDCWR), A ;SEND TRACK
CALL FDCINI ;WAIT FOR INTERRUPT
LD A, FDCSIS
OUT (FDCWR), A ;SEND COMMAND
CALL FDCRD2
IN A, (FDCRA) ;GET STATUS
CALL FDCRD2
IN A, (FDCRA)
RET
;

```

```

2133
*****
2134
2135
2136
2137
*****
2138
2139
2140 0971 DB 50
2141 0973 17
2142 0974 D2 71 09
Y
2143 0977 17
2144 0978 DA 71 09
NO INFO
2145 097B C9
2146
2147
2148
*****
2149
2150
2151
2152
*****
2153
2154
2155 097C DB 50
2156 097E 17
EADY TO BE READ
2157 097F D2 7C 09
2158 0982 17
2159 0983 D2 7C 09
2160 0986 C9
2161
2162
2163
*****
2164
2165
2166
2167
*****
2168
2169
2170 0987 DB 13
S
2171 0989 E6 08
2172 098B CA 87 09
UPT
2173 098E CD 71 09
2174 0991 C9
2175
2176
;*****
;
; WAIT UNTIL FDC IS READY TO BE WRITE
;
;*****
;
;
FDCRD1: IN A,(FDCST) ;GET MAINSTATUS
        RLA
        JP NC,FDCRD1 ;JUMP IF NOT READ
        RLA
        JP C,FDCRD1 ;JUMP IF FDC WANT
        RET
;
;
;*****
;
; WAIT UNTIL FDC IS READY TO BE READ
;
;*****
;
;
FDCRD2: IN A,(FDCST) ;GET MAINSTATUS
        RLA ;TEST IF FDC IS R
        JP NC,FDCRD2
        RLA
        JP NC,FDCRD2
        RET
;
;
;*****
;
; WAIT UNTIL FDC HAS BEEN INTERRUPTED
;
;*****
;
;
FDCIN1 IN A,(SYSSTA) ;GET SYSTEM-STATU
        AND FDCIN ;MASK FOR FDC-INT
        JP Z,FDCIN1 ;IF INTRR
FDC1: CALL FDCRD1
        RET
;
;

```

```

2180 ;*****
*****
2181 ;
2182 ; ACTIVATE DMA-CHANNEL 0 FOR FLEX
2183 ;
2184 ;*****
*****
2185 ;
2186 ;
2187 003F DMAID EQU 3FH ;64 BYTES OF ID'S FOR 16 SECTOR
S
2188 007F DMADAS EQU 7FH ;128 BYTES FOR SINGLE
2189 00FF DMADAD EQU 0FFH ;256 BYTES FOR DOUBLE
2190 ;
2191 ;
2192 ;
2193 0992 2A 02 FB DMA1 LD HL, (FARADR)
2194 0995 7E LD A, (HL)
2195 0996 1F RRA
2196 0997 DA 9F 09 JF C,LL184 ;JUMP IF WRITE-FUNCTION
2197 099A 06 04 LD B,004H ;WRITE MEMORY
2198 099C C3 A1 09 JF LL185
2199 099F 06 08 LD B,00BH ;READ MEMORY
2200 09A1 3E 43 LL185 LD A,043H
2201 09A3 B0 OR B
2202 09A4 D3 2B OUT (DMAM0),A ;OUT MODE
2203 09A6 11 08 00 LD DE,ALFARA
2204 09A9 19 ADD HL,DE
2205 09AA 7E LD A, (HL)
2206 09AB D3 26 OUT (C3AD),A
2207 09AD 23 INC HL
2208 09AE 7E LD A, (HL)
2209 09AF D3 26 OUT (C3AD),A
2210 09B1 2A 02 FB LD HL, (FARADR)
2211 09B4 7E LD A, (HL)
2212 09B5 E6 0F AND 00FH
2213 09B7 FE 0D CF 00DH
2214 09B9 C2 C3 09 JF NZ,LL186 ;JUMP IF NOT FORM
AT
2215 09BC 3E 3F LD A, DMAID
2216 09BE D3 27 OUT (C3TC),A ;64 BYTES
2217 09C0 C3 DD 09 JF LL188
2218 09C3 11 07 00 LD DE,DFARAM
2219 09C6 2A 02 FB LL186 LD HL, (FARADR)
2220 09C9 19 ADD HL,DE
2221 09CA 7E LD A, (HL) ;GET DENSITY
2222 09CB FE 00 CF 000H
2223 09CD CA 09 09 JF Z,LL187 ;JUMP IF SINGLE
2224 09D0 3E FF LD A, DMADAD
2225 09D2 D3 27 OUT (C3TC),A ;SET 512 BYTES
2226 09D4 3E 01 LD A, 01
2227 09D6 C3 DE 09 JF LL188
2228 09D9 3E 7F LL187 LD A, DMADAS
2229 09DB D3 27 OUT (C3TC),A ;128 BYTES
2230 09DD AF LL188 XOR A
2231 09DE D3 27 LL188: OUT (C3TC),A
2232 09E0 3E 03 LD A, 3
2233 09E2 D3 2A OUT (DMAMB),A ;ENABLE CHANNEL 0
2234 09E4 C9 RET
2235 ;
2236 ;
2237 ;
2238 09E5 DB 50 DMA01 IN A, (FDCST)
2239 09E7 17 RLA
2240 09E8 DA F2 09 JF C,LL190 ;JUMP IF MASTER REQUEST
2241 09EB DB 28 IN A, (DMAST)
2242 09ED E6 08 AND 008H
2243 09EF CA E5 09 JF Z,DMA01 ;JUMP IF NO TC
2244 09F2 D3 12 LL190 OUT (SETTC),A
2245 09F4 3E 07 LD A, 007H
2246 09F6 D3 2A OUT (DMAMB),A ;DISABLE CHANNEL
O
2247 09FB DB 50 LL191 IN A, (FDCST)
2248 09FA 17 RLA
2249 09FB DB RET C
2250 09FC C3 FB 09 JF LL191
2251 ;

```

```

2253 ;*****
*****
2254 ;
2255 ; DISPLAY STATUS BYTES ON CRT
2256 ;
2257 ;*****
*****
2258 ;
2259 ;
2260 ; *** FLEX DRIVER FOR 5 1/4 " ***
2261 ; *****
2262 ;
2263 ;
2264 FB02 PARADR EQU 0FB02H ;ADR OF PARAMETER BLOCK
2265 FB04 RESADR EQU 0FB04H ;ADR OF RESULT BLOCK
2266 FB06 PARAME EQU 0FB06H ;PARAMETER BLOCK
2267 FB10 RESULT EQU 0FB10H ;RESULT BLOCK
2268 FB17 COMBLO EQU 0FB17H ;COMAND BLOCK
2269 FB20 RETRY EQU 0FB20H ;RETRY COUNTER
2270 ;
2271 0030 CRTBUL EQU 30H
2272 00FB CRTBUH EQU 0FBH
2273 ;
2274 ; LOCATION IS CURSOR POSITION
2275 ; STATUS BYTES ARE IN RESULT-
2276 ; BLOCK B=STATUS BYTE-COUNTER
2277 ;
2278 ;
2279 09FF 3E 17 DISSTA LD A,017H
2280 0A01 32 01 FB LD (CURSY),A
2281 0A04 AF XOR A
2282 0A05 32 00 FB LD (CURSX),A
2283 0A08 21 52 0A LD HL,DIERRM
2284 0A0B 06 0C LD B,12
2285 0A0D 7E LD A,(HL)
LL117 CALL MODCRT ;SEND CHARACTER TO CRT
2286 0A0E CD 24 01 HL
2287 0A11 23 INC HL
2288 0A12 05 DEC B
2289 0A13 C2 0D 0A JF NZ,LL117
2290 0A16 06 07 LD B,007H
2291 0A18 2A 04 FB LD HL,(RESADR) ;ADR OF STATUS-BL
OCC
LL118 LD A,(HL) ;GET STATUS
2292 0A1B 7E RRA
2293 0A1C 1F RRA
2294 0A1D 1F RRA
2295 0A1E 1F RRA
2296 0A1F 1F RRA
2297 0A20 E6 0F AND 00FH
2298 0A22 FE 0A CP 00AH
2299 0A24 DA 2E 0A JF C,LL119
2300 0A27 D6 09 SUB 009H
2301 0A29 F6 40 OR 040H
2302 0A2B C3 30 0A JF LL120
2303 0A2E F6 30 LL119 OR 030H
2304 0A30 CD 24 01 LL120 CALL MODCRT
2305 0A33 7E LD A,(HL)
2306 0A34 E6 0F AND 00FH
2307 0A36 FE 0A CP 00AH
2308 0A38 DA 42 0A JF C,LL121
2309 0A3B D6 09 SUB 009H
2310 0A3D F6 40 OR 040H
2311 0A3F C3 44 0A JF LL122
2312 0A42 F6 30 LL121 OR 030H
2313 0A44 CD 24 01 LL122 CALL MODCRT
2314 0A47 3E 2E LD A,02EH
2315 0A49 CD 24 01 CALL MODCRT
2316 0A4C 23 INC HL ;NEXT STATUSADR.
2317 0A4D 05 DEC B
2318 0A4E C2 1B 0A JF NZ,LL118 ;ALL STATUS DISPL
AYED ?
2319 0A51 C9 RET
2320 ;
2321 0A52 44 DIERRM DEFB 'DISK ERROR'
0A53 49
0A54 53
0A55 4B
0A56 20
0A57 45
0A58 52
0A59 52
0A5A 4F
0A5B 52
0A5C 20
0A5D 20
2322

```

```

2326 ;*****
*****
2327 ;
2328 ; *** DISPLAY ROUTINE ***
2329 ;
2330 ;*****
*****
2331 ;
2332 ;
2333 0A5E 46 DISPL: LD B,(HL)
2334 0A5F 23 INC HL
2335 0A60 7E LD A,(HL)
2336 0A61 CD 24 01 CALL MODCRT
2337 0A64 10 F9 DJNZ DISPL+1
2338 0A66 23 INC HL
2339 0A67 C9 RET
2340 ;
2341 ;

```

```

2346
*****
2347
*****
2348
2349
2350
2351
*****
2352
*****
2353
2354
2355
2356
2357
2358
*****
2359
2360
2361
2362
*****
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374 0A68 C5
2375 0A69 D5
2376 0A6A EB
2377 0A6B 21 00 00
2378 0A6E 01 00 0B
2379 0A71 29
2380 0A72 D2 76 0A
2381 0A75 0C
2382 0A76 07
2383 0A77 D2 7F 0A
2384 0A7A 19
2385 0A7B D2 7F 0A
2386 0A7E 0C
2387 0A7F 05
2388 0A80 C2 71 0A
2389 0A83 79
2390 0A84 B7
2391 0A85 D1
2392 0A86 C1
2393 0A87 C8
2394 0A88 37
2395 0A89 C9
    
```

```

;*****
;*****
;
; SUBROUTINES
;
;*****
;*****
;
;
;*****
;
; MULTIPLY ROUTINE
;
;*****
;
; CALLING ARGUMENTS : <HL>= FIRST OPERAND
;                   <DE>= SECOND OPERAND
;
;
; RETURN ARGUMENTS : <HL>= RESULT
;
;
;
MULT:  PUSH    BC
       PUSH    DE
       EX      DE,HL
       LD     HL,0
       LD     BC,800H
MULT1: ADD    HL,HL
       JP    NC,MUL2
       INC   C
MULT2: RLCA
       JP    NC,MUL3
       ADD  HL,DE
       JP    NC,MUL3
       INC   C
MULT3: DEC    B
       JP    NZ,MUL1
       LD   A,C
       OR   A
       POP  DE
       BC
       SCF
       Z
       RET
    
```

```

2398
*****
2399
2400
2401
2402
*****
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417 0ABA 7C
2418 0ABB BA
2419 0ABC DB
2420 0ABD C0
2421 0ABE 7D
2422 0ABF BB
2423 0A90 C9
2424
2425
2426 0A91 A7
2427 0A92 ED 52
2428 0A94 C9
2429
2430
2431
2432
*****
2433
2434
2435
2436
*****
2437
2438
2439
F
2440
2441
2442
2443
2444 0A95 C5
2445 0A96 01 14 32
2446 0A99 0B
2447 0A9A 78
2448 0A9B B1
2449 0A9C C2 99 0A
2450 0A9F 15
2451 0AA0 C2 96 0A
2452 0AA3 C1
2453 0AA4 C9
2454

```

```

;*****
;
; DOUBLE BYTE COMPARE
;
;*****
;
; COMPARES <DE> WITH <HL>
; ANSWER RETURNS IN FLAGS :
;
;
; CARRY ZERO
; -----
;           C   Z
;
; DE < HL   0   0
; DE = HL   0   1
; DE > HL   1   0
;
;
; COMPDH: LD     A,H
;          CP     D
;          RET    C
;          RET    NZ
;          LD     A,L
;          CP     E
;          RET
;
;
; SUBHD:  AND     A
;          SBC    HL,DE
;          RET
;
;
;*****
;
; TIMER ROUTINE
;
;*****
;
; ENTRY CONDITION : <A> = MULTIPLIER OF 100MSEC I
;
; PROCESSOR-CLOCK-CYCLE = 325NSEC
;
;
;
; TIM100 PUSH  BC
; LL197  LD    BC,03214H
; LL198  DEC  BC
;          LD  A,B
;          OR  C
;          JP  NZ,LL198
;          DEC D
;          JP  NZ,LL197
;          POP BC
;          RET
;
;

```



```

2458 ;*****
*****
2459 ;
2460 ;     *** MONITOR-ROUTINEN ***
2461 ;
2462 ;*****
2463 ;
2464 ;
2465 ;
2466 A000 ; BUFFER EQU 0A000H ;DISK READ BUFFER
2467 5000 ; BUFCON EQU 5000H
2468 5100 ; BUFTRK EQU 5100H
2469 F600 ; RETRY1 EQU 0F600H
2470 F603 ; RETRY2 EQU 0F603H
2471 ;
2472 ;
2473 ;
2474 ;
2475 ;
2476 0AA5 DB 13 ; MONIT: IN A,(SYSSTA)
2477 0AA7 E6 02 ; AND 2
2478 0AA9 CA 1B 05 ; JP Z,DIAG16 ;JUMP TO 16 BIT
2479 0AAC 3E 0C ; LD A,SCRER ;
2480 0AAE CD 24 01 ; CALL MDCCRT ;CLEAR SCREEN
2481 0AB1 3E 21 ; LD A,33
2482 0AB3 32 00 FB ; LD (CURSX),A
2483 ;
2484 ; IF FARBE-1
2485 ; LD A,INVER
2486 ; ENDF
2487 ;
2488 ; IF FARBE
2489 0AB6 3E B4 ; LD A,10000100B ;RED OF Y
ELLOW ; ENDF
2490 ;
2491 ;
2492 0AB8 32 0B F9 ; LD (INVFLG),A ;MESSAGE INVERSE
2493 0ABB 21 53 0D ; LD HL,MESS
2494 0ABE CD 5E 0A ; CALL DISPL
2495 0AC1 3E 24 ; LD A,36
2496 0AC3 32 00 FB ; LD (CURSX),A
2497 0AC6 3E 01 ; LD A,01
2498 0AC8 32 01 FB ; LD (CURSY),A
2499 0ACB 3E EB ; LD A,0EBH
2500 0ACD 32 0B F9 ; LD (INVFLG),A
2501 0AD0 CD 5E 0A ; CALL DISPL
2502 0AD3 AF ; XOR A
2503 0AD4 32 00 FB ; LD (CURSX),A
2504 0AD7 3E 05 ; LD A,5
2505 0AD9 32 01 FB ; LD (CURSY),A
2506 0ADC 16 05 ; LD D,5 ;DISPLAY-COUNTER
2507 ;
2508 0ADE ; MONIT1:
2509 ; IF FARBE-1
2510 ; LD A,06H
2511 ; ENDF
2512 ;
2513 ; COND FARBE
2514 0ADE 3E E6 ; LD A,11100110B
2515 ; ENDC
2516 ;
2517 0AE0 32 0B F9 ; LD (INVFLG),A
2518 0AE3 CD 5E 0A ; CALL DISPL
2519 0AE6 3E EB ; LD A,0EBH
2520 0AEB 32 0B F5 ; LD (INVFLG),A
2521 0AED CD 5E 0A ; CALL DISPL
2522 0AEE 3E 9B ; LD A,NEWL
2523 0AF0 CD 24 01 ; CALL MDCCRT
2524 0AF3 15 ; DEC D
2525 0AF4 C2 DE 0A ; JP NZ,MONIT1
2526 ;
2527 0AF7 CD 13 01 ; CALL KEYINP
2528 0AFA FE 31 ; CP '1'
2529 0AFC CA 70 06 ; JP Z,STARTS ;LOAD SYSTEM
2530 0AFF FE 33 ; CP '3'
2531 0B01 CA 0C 0B ; JP Z,RS1 ;RESTORE DRIVE
2532 0B04 FE 35 ; CP '5'
2533 0B06 CA 2F 0B ; JP Z,READD ;READ SECTOR
2534 0B09 C3 A5 0A ; JP MONIT
2535 ;
2536 ;

```

```

2537
*****
2538
2539
2540
2541
*****
2542
2543
2544 0B0C 3E 0C
2545 0B0E CD 24 01
2546 0B11 3E 0A
2547 0B13 32 01 FB
2548 0B16 32 00 FB
2549 0B19 21 BF 0D
2550 0B1C CD 5E 0A
2551 0B1F CD 13 01
2552 0B22 D6 30
2553 0B24 FE 02
2554 0B26 D2 A5 0A
2555 0B29 CD 16 06
2556 0B2C C3 A5 0A
2557
2558

```

```

*****
;
; RESTORE DRIVE
;
*****
;
;
RS1: LD A,SCRER ;CLEAR SCREEN
CALL MODCRT
LD A,10
LD (CURSY),A
LD (CURSX),A
LD HL, TX1
CALL DISPL
CALL KEYINP
SUB 30H
CP 2
JP NC, MONIT
CALL REST
JP MONIT
;
;

```

```

2562 ; *****
*****
2563 ;
2564 ; READ SECTOR
2565 ;
2566 ; *****
*****
2567 ;
2568 ;
2569 READD: LD A,SCRER
2570 CALL MODCRT
2571 LD A,30
2572 LD (CURSX),A
2573 LD HL, TX6
2574 CALL DISPL
2575 LD A,5
2576 LD (CURSY),A
2577 XOR A
2578 LD (CURSX),A
2579 ;
2580 LD A,40H ; DOUBLE DENSITY
2581 LD (PARAME+7),A
2582 LD A,FDCRD
2583 LD (PARAME),A
2584 LD HL,BUFFER
2585 LD (PARAME+8),HL
2586 ;
2587 LD HL, TX3
2588 CALL DISPL ; GET SECTOR
2589 CALL GET
2590 LD (PARAME+5),A
2591 ;
2592 LD IX,BUFTRK
2593 LD A,NEWL
2594 CALL MODCRT
2595 LD HL, TX4
2596 CALL DISPL
2597 CALL GET ; GET TRACK
2598 LD (PARAME+4),A
2599 ;
2600 LD A,NEWL ; GET HEAD
2601 CALL MODCRT
2602 LD HL, TX5
2603 CALL DISPL
2604 GHD1: CALL KEYINP
2605 CALL CHECKA
2606 JP C,GHD1
2607 CP 2
2608 JP NC,GHD1
2609 CP 0
2610 JP Z,GHD2
2611 LD A,4
2612 GHD2: LD (PARAME+6),A
2613 LD A,B
2614 CALL MODCRT
2615 ;
2616 LD A,NEWL ; GET UNIT
2617 CALL MODCRT
2618 LD HL, TX2
2619 CALL DISPL
2620 GUI: CALL KEYINP
2621 CALL CHECKA
2622 JP C,GUI
2623 CP 2
2624 JP NC,GUI

```

```

2625 OBB9 32 09 F8      LD      (FARAME+3),A
2626 OBBC 78            LD      A,B
2627 OBBD CD 24 01      CALL    MODCRT
2628                    ;
2629 OBC0 3E 06            LD      A,6
2630 OBC2 32 00 F6      LD      (RETRY1),A
2631 OBC5 32 03 F6      LD      (RETRY2),A
2632 OBC8 21 06 F8      RDD1:   LD      HL,PARAME
2633 OBCE 11 10 F8      LD      DE,RESULT
2634 OBCE CD AB 07      CALL    FDCDR1
2635                    ;
2636 OBD1 21 10 F8      RDD3:   LD      HL,RESULT
2637 OBD4 3A 00 F6      LD      A,(RETRY1)
2638 OBD7 3D            DEC     A
2639 OBD8 32 00 F6      LD      (RETRY1),A
2640 OBD8 7E            LD      A,(HL)
2641 OBDC E6 C0        AND     OCOH ;TEST IF ERROR
2642 OBDE CA 06 0C        JF      Z,RDD2 ;JUMP IF NO DISK-
ERROR
2643 OBE1 CD FF 09      RDD4:   CALL    DISSTA
2644 OBE4 21 12 0E      LD      HL,TX7 ;DISPLAY ERROR
2645 OBE7 CD 5E 0A      CALL    DISFL
2646 OBEA 3A 00 F6      LD      A,(RETRY1)
2647 OBED C6 30        ADD     A,30H
2648 OBEF CD 24 01      CALL    MODCRT
2649 OBF2 CD 13 01      RDD5:   CALL    KEYINP
2650 OBF5 FE 88        CF     NEWL
2651 OBF7 C2 F2 0B      JF     NZ,RDD5
2652 OBFA 3A 00 F6      LD      A,(RETRY1)
2653 OBFD 06 00        LD      B,0
2654 OBFF B8          CF     B
2655 OC00 C2 CB 0B      JF     NZ,RDD1
2656 OC03 C3 A5 0A      JF     MONIT
2657                    ;
2658 OC06 23            RDD2:   INC     HL
2659 OC07 7E            LD      A,(HL)
2660 OC08 2B            DEC     HL
2661 OC09 E6 24        AND     24H ;TEST IF NO DATA
2662 OC0B CA 55 0C        JF     Z,DIDI ;JUMP IF READ 0.K
2663 OC0E C3 E1 0B      JF     RDD4
2664                    ;
2665                    ;
2666                    ;
2667 OC11 21 00 50      GET:   LD      HL,BUFCON
2668 OC14 16 02        LD      D,2
2669 OC16 CD 13 01      GET2:  CALL    KEYINP
2670 OC19 CD 32 0C      CALL    CHECKA
2671 OC1C DA 16 0C      JF     C,GET2
2672 OC1F 77          LD      (HL),A
2673 OC20 78          LD      A,B
2674 OC21 DD 77 00      LD      (IX+0),A
2675 OC24 DD 23        INC     IX
2676 OC26 CD 24 01      CALL    MODCRT
2677 OC29 23          INC     HL
2678 OC2A 15          DEC     D
2679 OC2B C2 16 0C      JF     NZ,GET2
2680 OC2E CD 40 0C      CALL    CONVER
2681 OC31 C9          RET
2682                    ;
2683                    ;
2684 OC32 47            CHECKA: LD     B,A
2685 OC33 FE 30        CF     30H
2686 OC35 DB          RET     C
2687 OC36 FE 3A        CF     3AH
2688 OC38 DA 3D 0C      JF     C,CHECKB
2689 OC3B 37          SCF
2690 OC3C C9          RET
2691 OC3D E6 0F        CHECKB: AND    0FH
2692 OC3F C9          RET
2693                    ;
2694                    ;
2695 OC40 21 00 50      CONVER: LD    HL,BUFCON
2696 OC43 7E            LD      A,(HL)
2697 OC44 47            LD      B,A
2698 OC45 FE 00        CF     0
2699 OC47 3E 00        LD      A,0
2700 OC49 CA 52 0C      JF     Z,C1
2701 OC4C C6 0A        C2:   ADD     A,10
2702 OC4E 05          DEC     B
2703 OC4F C2 4C 0C      JF     NZ,C2
2704 OC52 23          C1:   INC     HL
2705 OC53 86          ADD     A,(HL)
2706 OC54 C9          RET
2707                    ;
2708                    ;
2709                    ;

```

```

2714 ;*****
;
; DISPLAY A SECTOR
;*****
;
;
;
2719 ;
2720 ;
2721 0C55 21 00 A0 DID1: LD HL,BUFFER
2722 0C58 CB B2 RES 0,D
;
;
;
2723 ;
2724 0C5A E5 DIDIA: LD HL
2725 0C5B 3E 0C LD A,SCRER
2726 0C5D CD 24 01 CALL MODCRT ;ERASE CRT
;
;
;
2727 ;
2728 0C60 21 41 0E LD HL,TXU
2729 0C63 CD 5E 0A CALL DISPL ;DISPLAY UNIT NR.
2730 0C66 3A 09 F8 LD A,(PARAME+3)
2731 0C69 CD 34 0D CALL CVOUT
2732 0C6C 21 4E 0E LD HL,TXH
2733 0C6F CD 5E 0A CALL DISPL ;DISPLAY HEAD NR.
2734 0C72 3A 0C F8 LD A,(PARAME+6)
2735 0C75 CD 34 0D CALL CVOUT
2736 0C78 21 5C 0E LD HL,TXT
2737 0C7B CD 5E 0A CALL DISPL ;DISPLAY TRACK NR

2738 0C7E 3A 00 51 LD A,(BUFTRK)
2739 0CB1 CD 24 01 CALL MODCRT
2740 0CB4 3A 01 51 LD A,(BUFTRK+1)
2741 0CB7 CD 24 01 CALL MODCRT
2742 0CBA 21 68 0E LD HL,TXS
2743 0CBD CD 5E 0A CALL DISPL ;DISPLAY SECTOR N

R.
2744 0C90 3A 0B F8 LD A,(PARAME+5)
2745 0C93 CD 34 0D CALL CVOUT
2746 0C96 E1 POP HL
;
;
;
2747 ;
2748 0C97 3E 88 LD A,NEWL ;CR,LF,LF
2749 0C99 CD 24 01 CALL MODCRT
2750 0C9C 3E 88 LD A,NEWL
2751 0C9E CD 24 01 CALL MODCRT
2752 0CA1 3E 88 LD A,NEWL
2753 0CA3 CD 24 01 CALL MODCRT
2754 ;
2755 ;
2756 0CA6 7C DIDII: LD A,H
2757 0CA7 CD 34 0D CALL CVOUT ;DISPLAY H.ORDER

ADDRESS
2758 0CAA 7D LD A,L
2759 0CAB CD 34 0D CALL CVOUT ;DISPLAY L.ORDER

ADDRESS
2760 0CAE 06 10 LD B,10H
2761 0CB0 3E 20 LD A,020H
2762 0CB2 CD 24 01 CALL MODCRT ;SPACE
2763 ;
;
;
2764 0CB5 3E 20 DIDI2: LD A,020H
2765 0CB7 CD 24 01 CALL MODCRT ;SPACE
2766 0CBA 7E LD A,(HL)
2767 0CBB CD 34 0D CALL CVOUT ;DISPLAY A DATABY

TE (HEX)
2768 0CBE 23 INC HL
2769 0CBF 10 F4 DJNZ DIDI2 ;LOOP UNTIL 16 BY

TES DISPLAYED
2770 0CC1 7D LD A,L
2771 0CC2 D6 10 SUB 10H ;RET TO ADDR. OF

FIRST BYTE OF LINE
2772 0CC4 6F LD L,A
2773 0CC5 D2 C9 0C JP NC,DID17
2774 0CC8 25 DEC H
2775 ;
;
;
2776 0CC9 06 10 DIDI7: LD B,10H

```

```

2777 OCCB 3E 20          LD      A,020H
2778 OCCD CD 24 01      CALL   MODCRT          ;SPACE
2779 OCD0 3E 20          LD      A,020H
2780 OCD2 CD 24 01      CALL   MODCRT          ;SPACE
2781
;
2782 OCD5 7E            DID13: LD      A,(HL)
2783 OCD6 FE 20          CP     20H
2784 OCD8 DA E0 0C      JF     C,SUBST        ;JUMP IF <20H
2785 OCD8 FE 80          CP     80H
2786 OCDD DA E2 0C      JF     C,VALID
2787 OCE0 3E 2E          LD      A,2EH
2788 OCE2 CD 24 01      CALL   MODCRT        ;SET DOT
;                          ;DISPLAY A ASCII
CHAR.
2789 OCE5 23            INC     HL
2790 OCE6 10 ED          DJNZ   DID13        ;LOOP UNTIL 16 CH
AR. DISPLAYED
2791 OCE8 3E 88          LD      A,NEWL
2792 OCEA CD 24 01      CALL   MODCRT        ;CR,LF
2793
;
2794 OCED CB 42          BIT    0,D
2795 OCEF C2 15 0D      JF     NZ,DID15
2796 OCF2 CB 44          BIT    0,H
2797 OCF4 CA A6 0C      JF     Z,DID11
2798 OCF7 CB C2          SET    0,D
2799 OCF9 3E 18          LD      A,24
2800 OCFB 32 01 FB      LD      (CURSY),A    ;CURSOR TO LINE 2
5
2801 OCFE AF            XOR     A
2802 OCFF 32 00 FB      LD      (CURSX),A    ;CURSOR TO LEFT
2803 OD02 E5            PUSH   HL
2804 OD03 21 1A 0E      LD      HL,TX10
2805 OD06 CD 5E 0A      CALL   DISPL        ;DISPLAY TX10 MES
SAGE AT LAST LINE
2806 OD09 E1            POP     HL
2807 OD0A CD 13 01      DID14: CALL   KEYINP
2808 OD0D FE 88          CP     NEWL
2809 OD0F C2 0A 0D      JF     NZ,DID14      ;WAIT UNTIL NL IS
DEPRESSED
2810 OD12 C3 5A 0C      JP     DID1A
2811
;
2812 OD15 CB 4C          DID15: BIT    1,H
2813 OD17 CA A6 0C      JF     Z,DID11
2814 OD1A 3E 18          LD      A,24
2815 OD1C 32 01 FB      LD      (CURSY),A    ;CURSOR DOWN AND
LEFT
2816 OD1F AF            XOR     A
2817 OD20 32 00 FB      LD      (CURSX),A
2818 OD23 21 30 0E      LD      HL,TX11
2819 OD26 CD 5E 0A      CALL   DISPL        ;DISPLAY TX11 MES
SAGE
2820 OD29 CD 13 01      DID16: CALL   KEYINP
2821 OD2C FE 88          CP     NEWL
2822 OD2E C2 29 0D      JF     NZ,DID16      ;WAIT UNTIL NL IS
DEPRESSED
2823 OD31 C3 A5 0A      JF     MONIT
2824
;
2825
;
2826
;
2827 CONVERT HEX TO ASCII AND DISPLAY IT
2828 *****
2829
;
2830 OD34 F5            CVOUT: PUSH   AF
2831 OD35 0F            RRCA
2832 OD36 0F            RRCA
2833 OD37 0F            RRCA
2834 OD38 0F            RRCA          ;UPPER 4BITS TO L
OWER POSIT.
2835 OD39 E6 0F          AND     00FH        ;MASK OUT UPPER 4
BITS
2836 OD3B CD 4B 0D      CALL   HEASCI
2837 OD3E CD 24 01      CALL   MODCRT        ;CONVERT TO ASCI
;SEND TO CRT
2838 OD41 F1            POP     AF
2839 OD42 E6 0F          AND     00FH
2840 OD44 CD 4B 0D      CALL   HEASCI        ;CONVERT TO ASCII
2841 OD47 CD 24 01      CALL   MODCRT        ;SEND TO CRT
2842 OD4A C9            RET
2843
;
2844 OD4B E7            HEASCI: OR      A
2845 OD4C C6 70          ADD     A,090H
2846 OD4E 27            DAA
2847 OD4F CE 40          ADC     A,040H
2848 OD51 27            DAA
2849 OD52 C9            RET
2850
;
2851
;
2852
;
2853
;

```

2855		:		
2856		:		
2857	0D53 0D	MESS:	DEFB	13, ' DMS MONITOR
	0D54 20			
	0D55 44			
	0D56 4D			
	0D57 35			
	0D58 20			
	0D59 4D			
	0D5A 4F			
	0D5B 4E			
	0D5C 49			
	0D5D 54			
	0D5E 4F			
	0D5F 52			
	0D60 20			
2858	0D61 06		DEFB	6, '(V0.6)'
	0D62 28			
	0D63 56			
	0D64 30			
	0D65 2E			
	0D66 36			
	0D67 29			
2859	0D68 01		DEFB	1, '1'
	0D69 31			
2860	0D6A 0C		DEFB	12, ' LOAD SYSTEM'
	0D6B 20			
	0D6C 4C			
	0D6D 4F			
	0D6E 41			
	0D6F 44			
	0D70 20			
	0D71 53			
	0D72 59			
	0D73 53			
	0D74 54			
	0D75 45			
	0D76 4D			
2861	0D77 01		DEFB	1, '2'
	0D78 32			
2862	0D79 0C		DEFB	12, ' FORMAT DISK'
	0D7A 20			
	0D7B 46			
	0D7C 4F			
	0D7D 52			
	0D7E 4D			
	0D7F 41			
	0D80 54			
	0D81 20			
	0D82 44			
	0D83 49			
	0D84 53			
	0D85 48			
2863	0D86 01		DEFB	1, '3'
	0D87 33			
2864	0D88 0D		DEFB	13, ' RESTORE DISK'
	0D89 20			
	0D8A 52			
	0D8B 45			
	0D8C 53			
	0D8D 54			
	0D8E 4F			
	0D8F 52			
	0D90 45			
	0D91 20			
	0D92 44			
	0D93 49			
	0D94 53			
	0D95 48			
2865	0D96 01		DEFB	1, '4'
	0D97 34			
2866	0D98 12		DEFB	18, ' WRITE DISK-SECTOR'
	0D99 20			
	0D9A 57			
	0D9B 52			
	0D9C 49			
	0D9D 54			
	0D9E 45			
	0D9F 20			
	0DA0 44			
	0DA1 49			
	0DA2 53			
	0DA3 48			
	0DA4 2D			
	0DA5 53			
	0DA6 45			
	0DA7 43			

	ODAB 54		
	ODA9 4F		
	ODAA 52		
2867	ODAB 01	DEFB	1, '5'
	ODAC 35		
2868	ODAD 11	TX6:	DEFB 17, 'READ DISK-SECTOR'
	ODAE 20		
	ODAF 52		
	ODB0 45		
	ODB1 41		
	ODB2 44		
	ODB3 20		
	ODB4 44		
	ODB5 49		
	ODB6 53		
	ODB7 48		
	ODB8 2D		
	ODB9 53		
	ODBA 45		
	ODBB 43		
	ODBC 54		
	ODBD 4F		
	ODBE 52		
2869			
2870	ODBF 1A	TX1:	DEFB 26, RESTORE DRIVE A(0) OR B(1)
	ODC0 52		
	ODC1 45		
	ODC2 53		
	ODC3 54		
	ODC4 4F		
	ODC5 52		
	ODC6 45		
	ODC7 20		
	ODC8 44		
	ODC9 52		
	ODCA 49		
	ODCB 56		
	ODCC 45		
	ODCD 20		
	ODCE 41		
	ODCF 28		
	ODD0 30		
	ODD1 29		
	ODD2 20		
	ODD3 4F		
	ODD4 52		
	ODD5 20		
	ODD6 42		
	ODD7 28		
	ODD8 31		
	ODD9 29		
2871	ODDA 0D	TX2:	DEFB 13, UNIT (0/1) :
	ODDB 55		
	ODDC 4E		
	ODDD 49		
	ODDE 54		
	ODDF 20		
	ODE0 28		
	ODE1 30		
	ODE2 2F		
	ODE3 31		
	ODE4 29		
	ODE5 20		
	ODE6 3A		
	ODE7 20		
2872	ODE8 0D	TX3:	DEFB 13, SECTOR 1-8 :
	ODE9 53		
	ODEA 45		
	ODEB 43		
	ODEC 54		
	ODED 4F		
	ODEE 52		
	ODEF 20		
	ODF0 31		
	ODF1 2D		
	ODF2 38		
	ODF3 20		
	ODF4 3A		
	ODF5 20		
2873	ODF6 0D	TX4:	DEFB 13, TRACK 0-39 :
	ODF7 54		
	ODF8 52		
	ODF9 41		
	ODFA 43		
	ODFB 4B		
	ODFC 20		
	ODFD 30		

	0DFE 2D			
	0DFF 33			
	0E00 39			
	0E01 20			
	0E02 3A			
	0E03 20			
2874	0E04 0D	TX5:	DEFB	13, 'HEAD (0/1) :
	0E05 48			
	0E06 45			
	0E07 41			
	0E08 44			
	0E09 20			
	0E0A 28			
	0E0B 30			
	0E0C 2F			
	0E0D 31			
	0E0E 29			
	0E0F 20			
	0E10 3A			
	0E11 20			
2875	0E12 07	TX7:	DEFB	7, ' (CR)
	0E13 20			
	0E14 20			
	0E15 28			
	0E16 43			
	0E17 52			
	0E18 29			
	0E19 20			
2876	0E1A 15	TX10:	DEFB	21, 'HIT NL FOR NEXT PAGE
	0E1B 48			
	0E1C 49			
	0E1D 54			
	0E1E 20			
	0E1F 4E			
	0E20 4C			
	0E21 20			
	0E22 46			
	0E23 4F			
	0E24 52			
	0E25 20			
	0E26 4E			
	0E27 45			
	0E28 58			
	0E29 54			
	0E2A 20			
	0E2B 50			
	0E2C 41			
	0E2D 47			
	0E2E 45			
	0E2F 20			
2877	0E30 10	TX11:	DEFB	16, 'HIT NL FOR EXIT
	0E31 48			
	0E32 49			
	0E33 54			
	0E34 20			
	0E35 4E			
	0E36 4C			
	0E37 20			
	0E38 46			
	0E39 4F			
	0E3A 52			
	0E3B 20			
	0E3C 45			
	0E3D 58			
	0E3E 49			
	0E3F 54			
	0E40 20			
2878	0E41 0C	TXU:	DEFB	12, 'DISK UNIT :
	0E42 44			
	0E43 49			
	0E44 53			
	0E45 4B			
	0E46 20			
	0E47 55			
	0E48 4E			
	0E49 49			
	0E4A 54			
	0E4B 20			
	0E4C 3A			
	0E4D 20			
2879	0E4E 0D	TXH:	DEFB	13, ' HEAD NO. :
	0E4F 20			
	0E50 20			
	0E51 20			
	0E52 4B			
	0E53 45			
	0E54 41			

	0E55	44				
	0E56	20				
	0E57	4E				
	0E58	4F				
	0E59	2E				
	0E5A	3A				
	0E5B	20				
2880	0E5C	0E	TXT:	DEFB	14,	TRACK NO.:
	0E5D	20				
	0E5E	20				
	0E5F	20				
	0E60	54				
	0E61	52				
	0E62	41				
	0E63	43				
	0E64	4B				
	0E65	20				
	0E66	4E				
	0E67	4F				
	0E68	2E				
	0E69	3A				
	0E6A	20				
2881	0E6B	0F	TXS:	DEFB	15,	SECTOR NO.:
	0E6C	20				
	0E6D	20				
	0E6E	20				
	0E6F	53				
	0E70	45				
	0E71	43				
	0E72	54				
	0E73	4F				
	0E74	52				
	0E75	20				
	0E76	4E				
	0E77	4F				
	0E78	2E				
	0E79	3A				
	0E7A	20				
2882						

```

2883                                     ;*****
*****
2884                                     ;
2885                                     ;     VERSIONS-NUMBER
2886                                     ;
2887                                     ;*****
*****
2888                                     ;
2889                                     ;
2890 OFF7                                ;     ORG     1000H-9
2891                                     ;
2892                                     ;     IF     FARBE-1
2893                                     ;
2894 VERSIO: DEFB 8, ' M.05.01 '
2895                                     ;
2896                                     ;     ENDIF
2897                                     ;
2898                                     ;     COND  FARBE
2899                                     ;
2900 OFF7 08                                ;
      OFFB 20
      OFF9 43
      OFFA 2E
      OFFB 30
      OFFC 35
      OFFD 2E
      OFFE 30
      OFFF 31
2901                                     ;
2902                                     ;     ENDC
2903                                     ;
2904 1000                                  ;     END
    
```

ADD16	FE06	ADDB	FE02	ADWPR	0050	AHPARA	0009
AL	0190	ALFARA	0008	ANFDIA	2000	BADSTA	05D2
BB	0080	BEGIN	0100	BB	0040	BLINK	0002
BR	0020	BRUN	0571	BUFCON	5000	BUFFER	A000
BUFSTA	F830	BUFRK	5100	COAD	0020	C1	0C52
C2	0C4C	C2AD	0024	C2TC	0025	C3AD	0026
C3TC	0027	CALCH1	01FB	CALCH2	01F4	CALCHR	01E5
CAR1	0152	CARRET	013F	CB16	FE04	CB8	FE00
CCHAR	0048	CAD	F902	CEADH	F901	CEADL	F900
CHA1	0178	CHA2	018C	CHAMDD	0020	CHARA	0050
CHECK1	0742	CHECKA	0C32	CHECKB	0C3D	CHECK1	073D
CHP	0158	CHRRAM	FA00	CKDA10	04B7	CKDATR	04B6
CLRL	0220	CLRRA	03FB	COMBL0	F817	COMPDH	0ABA
CONVER	0C40	CRIBUH	00FB	CRIBUL	0030	CUHOM1	02B2
CURD	00E0	CURS	0049	CURSX	F800	CURSY	F801
CVOUT	0D34	DATRDY	0001	DGT1	0025	DGTEST	0042
DI A16	051E	DIACOM	00FF	DIAG16	051B	DIAGE	0556
DIATSK	F700	DIDI	0C55	DID11	0CA6	DID12	0CB5
DIDI3	0CD5	DIDI4	0D0A	DID15	0D15	DID16	0D29
DIDI7	0CC9	DIDIA	0C5A	DIERRM	0A52	DISPL	0A5E
DISSTA	09FF	DMA01	09E5	DMA1	0992	DMACOM	0028
DMADAD	00FF	DMADAS	007F	DMAERR	00FA	DMAEXC	0010
DMAFF	002C	DMAID	003F	DMAMB	002A	DMAM0	002B
DMAR	00A4	DMASEL	0020	DMAT5	0028	DMATS	03C5
DMAW	0024	DMON	0938	DPARAM	0007	DRIVES	08CB
DRWALL	0000	DRWINP	0008	DRWRET	0010	DYNRAM	0004
ENDLD	0614	ENDLD2	060D	ERRDIS	0000	ERRDMA	03EE
FARBE	0001	FB	0010	FDC1	09BE	FDCDR1	07A8
FDCBOT	0008	FDCERR	00DE	FDCFT	004D	FDCIN	0008
FDCIN1	0987	FDCRA	0051	FDCRD	0046	FDCRD1	0971
FDCRD2	097C	FDCRE	0007	FDCRID	004A	FDCSEE	000F
FDCSEL	0050	FDCSEN	0004	FDCSIS	0008	FDCSFE	0003
FDCST	0050	FDCWD	0045	FDCWR	0051	FG	0008
FIFEMP	0004	FIFO	00A1	FIF010	04AD	FIFRDY	04AC
FIFULL	0002	FIGD	006C	FIGS	004C	FLEXTS	04FE
FFARAM	0000	FR	0004	FRTRK	0042	FUBLK	08CC
GCHRD	0068	GDCCOM	00A1	GDCERR	00BE	GDCINT	03F3
GDCPAR	00A0	GDCPAZ	00A2	GDCRES	0000	GDCSTA	00A0
GET	0C11	GET2	0C16	GETBY1	092B	GETRE	019C
GHD1	0884	GHD2	0899	GOODST	05B6	GRAMOD	0002
GRUN	0568	GU1	0BAB	HBP	000A	HEASCI	0D4B
HFF	0006	HOME	0001	HORETR	0040	HPARAM	0006
HSYNC	0007	IDENT1	0793	IDENT2	079A	IDENT3	07A1
INIT10	02E8	INIT10	0415	INIT1A	0276	INIT20	048D
INIT11	04DE	INLRPF	0008	INPAR	04C0	INTLAC	0009
INTPAR	04D0	INVAL1	03A7	INVAL2	03AF	INVAL3	03B2
INVLFG	F90B	KEY1	037F	KEY2	03BA	KEY3	0397
KEYCDM	0041	KEYDAT	0040	KEYERR	00F6	KEYINP	0113
KEYSTA	0041	KEYTST	0379	KYLOOK	011B	LEVEL0	02FD
LLPDET	0080	LL117	0A0D	LL118	0A1B	LL119	0A2E
LL120	0A30	LL121	0A42	LL122	0A44	LL124	032C
LL125	0339	LL126	034F	LL127	0352	LL128	035F
LL140	0508	LL141	0625	LL157	08F4	LL175	0907
LL177	091B	LL179	092E	LL184	099F	LL185	09A1
LL186	09C3	LL187	09D9	LL188	09DD	LL1881	09DE
LL190	09F2	LL191	09FB	LL197	0A96	LL198	0A99
LP11	F905	LF12	F906	LP21	F909	LP22	F90A
LPARAM	0001	LPC	000A	LPR0	00C0	MASK	004A
NCLSCR	0213	MEMERR	00FC	MEMORY	M 0000	MESN16	0762
MESS	0D53	MIXGAC	0000	MOCOMP	0001	MODCRT	0124
MON	0014	MONI	0AAC	MONIT	0AAS	MONIT1	0ADE
MOREPL	0000	MORES	0002	MOSET	0003	MUL1	0A71
MUL2	0A76	MUL3	0A7F	MULT	0A68	NC	0001
NEWL	0088	NEXT1	0375	NEXT2	03BF	NEXT5	04FA
ND16	0718	NDINTL	0000	NOMOV	0000	NOTRM	0659
NRWAPL	0050	OUTCMD	04A0	OUTPAR	04A6	OUTPAZ	049A
P16ERR	00EE	PARADR	F802	PARAME	F806	PHASE0	055E
PHASE1	057A	PHASE2	057D	PHMES	0596	PITCH	0047
PIXEL	0280	PRAM	0070	FRAMSA	0000	PRISEL	0060
PRJUMP	0580	PRMPAR	04DB	PROCCB	00D0	PROCHK	0305
PROERR	0000	PROGRA	058E	RAMEND	0000	RAMSEL	0010
RAMTDP	F10C	RAMTST	0323	RDAT	00A0	RDD1	08CB
RDD2	0C06	RDD3	08D1	RDD4	0BE1	RDD5	0BF2
RDDAT1	07F0	RDMST	0050	RDSID1	084C	REACCA	01D3
READD	0B2F	READTR	089F	RECAL1	0801	REGADR	F804
RESMOP	0000	REST	0616	REST2	0648	REST3	0653
RESULT	F810	RETRY	F820	RETRY1	F600	RETRY2	F603
ROMADR	2006	ROMSEL	0011	ROWS	0018	RS1	0B0C
SB16	FE05	S88	FE01	SCERA	0162	SCRER	000C
SCROFF	000C	SCR0L1	02DA	SCR0LM	0230	SCR0N	000D
SEEK1	0944	SENBY1	0901	SENDB1	0911	SENPAR	04C6
SETCCA	01C1	SETCHA	0168	SETCR	0289	SETCU	02C2
SETMSK	02A4	SETTC	0012	SL2	0721	SL3	072D
SLMESS	074A	SLOAD	067E	SLOAD5	069A	SLOAD6	069E
SLOAD7	06CD	SLOAD8	068D	SLOAD9	06D3	SLOADA	06E4
SLOADB	0702	SLOADC	071E	SLOADD	071E	SP1	F903

SP2	F907	SPARAM	0005	STACK	S 0000	START	006B
STARTS	0670	STATRM	0000	STK	F000	SUBHD	0A91
SUBST	0CE0	SUM1	0314	SUMCHK	0309	SUMERR	007E
SYNCOF	000E	SYNCON	000F	SYSTA	0013	TIMO	00B0
TIM1	00B1	TIM100	0A95	TIM2	00B2	TIMMO	00B3
TPARAM	0004	TST16	05DE	TSTEND	0606	TX1	0DBF
TX10	0E1A	TX11	0E30	TX2	0DDA	TX3	0DEB
TX4	0DF6	TX5	0E04	TX6	0DAD	TX7	0E12
TXH	0E4E	TXS	0E6B	TXT	0E5C	TXU	0E41
TYHIBY	001B	TYLOBY	0010	TYWORD	0000	UPARAM	0003
VALID	0CE2	VBF	000F	VERETR	0020	VERSIO	0FF7
VFP	000A	VSYNC	000B	VSYNOM	006F	VSYNCS	006E
VV200	1000	VV205	F80D	VV206	F80E	VV207	F820
WDAT	0020	WDAT1	0002	WRDAT1	07DF	WRFOR1	0B6E
WRMCHR	01A1	ZOOM	0046				

KEYBOARD FIRMWARE (VERSION 1)

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	000010		;				*****
	000020		;		*	THIS DOCUMENT, FILES AND INFORMATION HEREIN	*
	000030		;		*	ARE CONFIDENTIAL AND PROPRIETARY PROPERTY	*
	000040		;		*	OF NCR CORPORATION AND ALL UNAUTHORIZED	*
	000050		;		*	USE AND REPRODUCTION IS PROHIBITED	*
	000060		;				*****
	000070		;				
	000080		;				
	000090		;				*****
	000100		;		*		*
	000110		;		*	COPYRIGHT (C) 1983	*
	000120		;		*	BY	*
	000130		;		*	NCR CORPORATION	*
	000140		;		*	DAYTON, OHIO U.S.A.	*
	000150		;		*	ALL RIGHTS RESERVED	*
	000160		;		*	PRINTED IN U.S.A.	*
	000170		;		*	CONFIDENTIAL, UNPUBLISHED	*
	000180		;		*	PROPERTY OF NCR CORPORATION	*
	000190		;		*		*
	000200		;				*****

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CND	OPERAND	COMMENT
			000290	:		*****	
			000300	:	*	*	
			000310	:	*	* PERSONAL-COMP.-KEYBOARD*	
			000320	:	*	*	
			000330	:	*	* ASCII ENCODED PLL/SER. *	
			000340	:	*	*	
			000350	:	*	APRIL 83	*
			000360	:	*	*	
			000370	:	*	X-TAL= 6.00 MHZ	*
			000380	:	*	*	
			000390	:		*****	
			000400	:			
			000410	:			
			000420	:	REGISTER BANK 0		REGISTER BANK 1
			000430	:	-----		-----
			000440	:	R0...TEMPORARY STORAGE		R0...TEMPORARY STORAGE
			000450	:	R1...STATUS ARRAY POINTER		R1...TEMPORARY STORAGE
			000460	:	R2...TEMPORARY STORAGE		R2...ERRCNT(BELL MULTIPLIER)
			000470	:	R3...TABLE STORAGE		R3...XMIT BIT COUNTER
			000480	:	R4...REPEAT DELAY CTR		R4...BAUD CLOCK (BEEP CYCLES)
			000490	:	R5...QUEUE TAIL POINTER		R5...XMIT CHARACTER BUFFER
			000500	:	R6...QUEUE HEAD POINTER		R6...XMIT SHIFT REGISTER
			000510	:	R7...8 KEY STATUS		R7...SAVE ACC DURING INTRPTS
			000520	:			
			000530	:	RAM: 26...3F	KEY STATUS 2 BITS/KEY	4 KEYS/BYTE
			000540	:		STATUS TURNAROUND 2 BITS LEFT FOR EACH KEY	
			000550	:	10...17	QUEUE OF CODES TO BE SENT (FIFO)	
			000560	:			
			000570	:	TABLES: 300...36C	MARKBIT-TABLE	
			000580	:			
			000590	:	36D...3DB	ASCII CODE TABLE	
			000600	:	FD	SERIAL KB TOGGLE	
			000610	:			
			000620	:	MARKBIT ASSIGNMENTS:		
			000630	:	BIT 0...0		
			000640	:	BIT 1...0		
			000650	:	BIT 2... >	EXCEPTION OFFSET	
			000660	:	BIT 3...0		
			000670	:	BIT 4.../	SHIFKY(B7=1)	
			000680	:	BIT 5...IGNORE	SHIFT(B0...4,7=0)/CTLKY(B7=1)/	
			000690	:	IF B0...4>0	EXCEPTION CODES UN-/SHIFT-FLAG DEPEND.	
			000700	:	BIT 6...IGNORE	CTRL-FLAG/LOCKKY(B7=1)/SPEZ.XLAT(B0...4>0)	
			000710	:	BIT 7...FUNCTION	KEY/NO AUTOREPEAT INDICATOR/SHF.CTL.LOCK-	
			000720	:			
			000730	:	80(FUNCTION KEYS F1...F20)	90(SHIFT KEYS) A0(CONTROL KEY	
			000740	:	C0(ALPHA LOCK KEY) 20(SHIFT COND. IGNORED) 40(CTRL. COND. IGN		
			000750	:	1...38 (SINGLE CODE EXCEPTIONS) 4X(SPECIAL EXCEPTIONS)		
			000760	:	DO NO EXCEPTIONS, NORMAL CODE CALCULATION:		
			000770	:	UNSHIFT CODE EXORED BY SHIFT OR CONTROL FLAG.		
			000780	:	SHIFT FLAG INVERTED FOR LETTER CODES DURING ALPHA LOCK ON		
			000790	:	AND CONTROL FLAG OFF.		
			000800	:	CONTROL MODE IGNORED FOR CHARACTER CODES < \$		
			000810	:	SHIFT- AND CONTROL-MODE COINCIDENCE DISABLED FOR ALL KEYS		

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CND	OPERAND	COMMENT
			000830	RBO	EQUATES		
			000840		-----		
			000850				
	0001		000860	STPTR	EQV	R1	:STATUS POINTER
			000870				:BIT 0=0 KEY UP AT LAST SCAN
			000880				:1 KEY DOWN AT LAST SCAN
			000890				:AFTER CHECKING KEY DOWN BIT 6&7 ARE ROTATED TO
			000900				
	0003		000910	TBLPTR	EQV	R3	:POINTER TO R/C VALUE IN PAGE 3
	0004		000920	REPDLY	EQV	R4	:REPEAT DELAY COUNTER FOR AUTO REPEA
	0005		000930	QTPTR	EQV	R5	:QUEUE TAIL POINTER
	0006		000940	QHPTR	EQV	R6	:QUEUE HEAD POINTER
	0007		000950	KYDOWN	EQV	R7	:8 KEY DOWN REGISTER
			000960				:TIMES HOW LONG KEYS LOCKED OUT
			000970				
			000980		RB1	EQUATES	
			000990		-----		
			001000				
	0002		001010	ERRCNT	EQV	R2	:BEEP MULTIPLIER FOR LONG TONE(.5SEC
	0003		001020	BITCTR	EQV	R3	:COUNTS SERIAL BITS SHIFTED OUT
	0004		001030	BDCLK	EQV	R4	:BIT COUNTER FOR BAUD CLOCK
	0005		001040	CHARBF	EQV	R5	:CHARACTERBUFFER FOR SERIAL OUTPUT
	0006		001050	SHFTR6	EQV	R6	:SHIFT REGISTER
	0007		001060	ATEMP	EQV	R7	:SAVE ACC DURING INTERRUPT
			001070				
			001080		OTHERS		
			001090		-----		
			001100				
	007E		001110	CAFL6	EQV	F1	:CHARACTER AVAILABLE FLAG
	0060		001120	LONGDLY	EQV	96	
	0008		001130	REPRAT	EQV	8	:APPROX. 11 HZ
	FFFD		001140	THRCNT	EQV	-3	
	0008		001150	DOUBLO	EQV	11	:DOUBLE ZERO DISPLACEMENT
	0020		001160	STRAPS	EQV	32	:STRAPPING INFO
	0021		001170	FLAGB	EQV	33	:MP-,CTL-,SHIFT FLAGB
	0022		001180	LASTCD	EQV	34	:INTERIM STORAGE CELL
	0023		001190	LASTKY	EQV	35	:LAST KEY POSITION
	0010		001200	FIFO	EQV	16	:FIFO START ADDRESS
	0017		001210	FIFOEND	EQV	23	:LAST ADDRESS
	0024		001220	DECP01	EQV	36	:SOFTADDRESS OF DEC.POINT/COMMA

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0000		001240		ORG	0	
			001250	:			
			001260	:			RESTART, INTERRUPT VECTORS
			001270	:			
	0000		001280		RESTR:		
	0000	35	001290		DIS	TCNTI	;DISABLE TIMER INTERRUPT
	0001	0484	001300		JMP	START	;START KB SCAN
			001310	:			
	0003		001320		INTRUPT:		
	0003	D5	001330		SEL	RB1	
	0004	AF	001340		MOV	ATEMP,A	
	0005	042A	001350		JMP	INTRPT	
			001360	:			
			001370	:			TIMER INTERRUPT
			001380	:			*****
			001390	:			
			001400	:			THIS INTERRUPT IS EVERY 240MYSEC(XTAL=6. MHZ) OR 4MSEC
			001410	:			THRCNT=3 IT SERVES TWO FUNCTIONS:
			001420	:			
			001430	:			1) BEEP TONE GENERATION(240MYSEC)
			001440	:			2) EMPTY KB QUEUE (4MS)
			001450	:			
	0007		001460		TIMER:		
	0007	D5	001470		SEL	RB1	
	0008	AF	001480		MOV	ATEMP,A	;SAVE ACC
	0009	23CE	001490		MOV	A,#-50	;4MS PAUSE
	0008	62	001500		MOV	T,A	
			001510	:			
			001520	:			OUTPUT BAUD CLOCK TO PORT 26
			001530	:			
	000C	FC	001540		MOV	A,BDCLK	;ODD/EVEN COUNT TEST
	000D	F221	001550		JB7	\$4	
	000F	9617	001560		JNZ	\$5	;BEEP TONE IS SET
	0011	FA	001570		MOV	A,ERRCNT	;TEST BEEP MULTIPLIER
	0012	C617	001580		JZ	\$5	;NO, CONTINUE NORMAL
	0014	CA	001590		DEC	ERRCNT	;ELSE DECREMENT ERROR COUNTER
	0015	BC3C	001600		MOV	BDCLK,#60	;SET COUNT OF CYCLES FOR BEEP
	0017		001610		\$5:		
	0017	0A	001620		IN	A,P2	;LOAD BEEP CONDITION
	0018	D34D	001630		XRL	A,#4DH	;SWAP PHASE
	001A	3A	001640		OUTL	P2,A	
	001B	CC	001650		DEC	BDCLK	
	001C		001660		\$1:		
	001C	23FD	001670		MOV	A,#THRCNT	;RELOAD TIMER REGISTER
	001E	62	001680		MOV	T,A	
	001F	046B	001690		JMP	EXINT	
	0021		001700		\$4:		
	0021	D63D	001710		JNIBF	MTKB0	;TEST ON BELL LATCH SET
	0023	22	001720		IN	A,DBB	;RESET BELL LATCH
	0024	BA23	001730		MOV	ERRCNT,#35	;SET MULTIPLIER
	0026	8C3C	001740		MOV	BDCLK,#60	
	0028	041C	001750		JMP	\$1	
			001760	:			

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	002A		001780	INTRPT:			:IBF IS SET
	002A	D66B	001790	JNIBF	EXINT		:IBF OFF BUT INT.REQUEST PENDING
	002C	22	001800	IN	A,08B		:CLEAR IBF
	002D	B82C	001810	MOV	RD,#44		:SET DELAY COUNTER 300 USEC
	002F	E82F	001820	DJNZ	RD,#		
	0031	D63D	001830	JNIBF	MTKBQ		:THERE WAS NO 2ND POLL(WR/) SIGNAL
	0033	B820	001840	MOV	RD,#STRAPS		:LOAD LANGUAGE CODE
	0035	FD	001850	MOV	A,\$RD		
	0036	43E0	001860	ORL	A,#-32		:SET UP UPPER BITS +ASSUME GOOD KB S
	0038	53EF	001870	ANL	A,#-17		
	003A	AD	001880	MOV	CHARBF,A		
	0038	D46D	001890	JMP	SEND		
			001900	:			
			001910	:	MTKBQ-	EMPTY KEYBOARD QUEUE	
			001920	:	-----		
			001930	:	LOOKS INTO QUEUE ONCE EVERY INTERRUPT.		
			001940	:	IF THERE IS A CODE, IT IS PUT INTO CHARF		
			001950	:	THE CHARACTER AVAILABLE FLAG IS SET		
			001960	:	THE QUEUE POINTER IS UPDATED UNTIL EMPTY CONDITION		
			001970	:			
	003D		001980	MTKBQ:			
	003D	B820	001990	MOV	RD,#STRAPS		:SELECT SERIAL INTERFACE
	003F	FD	002000	MOV	A,\$RD		
	0040	37	002010	CPL	A		:INVERT INDICATOR
	0041	AB	002020	MOV	R3,A		:SAVE FOR FURTHER USE
	0042	9246	002030	JB4	#+4		:SKIP MASTER SLAVE IF RESPECTS
	0044	B66B	002040	JOBFF	EXINT		:EXIT IF LAST KEY NOT YET FETCHED
	0046	B805	002050	MOV	RD,#PTR		:ELSE LOOK INTO QUEUE TO FIND CODES
	0048	FD	002060	MOV	A,\$RD		: FOR TRANSMIT
	0049	18	002070	INC	RD		
	004A	D0	002080	XRL	A,\$RD		: HEAD=TAIL MEANS EMPTY
	0048	C66B	002090	JZ	EXINT		:FIFO EMPTY
			002100	:	ELSE GET A CODE FROM QUEUE		
	004D	FD	002110	MOV	A,\$RD		:PUT POINTER INTO R1
	004E	A9	002120	MOV	R1,A		
	004F	F1	002130	MOV	A,\$R1		:GET THE CODE
	0050	AD	002140	MOV	CHARBF,A		:PUT INTO BUFFER
	0051	FD	002150	MOV	A,\$RD		:UPDATE HEAD POINTER
	0052	17	002160	INC	A		
	0053	5317	002170	ANL	A,#17H		:MAKE RING COUNTER
	0055	AD	002180	MOV	\$RD,A		:AND REPLACE
	0056		002190	#:1:			
	0056	F8	002200	MOV	A,R3		:RELOAD STRAPINFO
	0057	F26D	002210	JB7	SEND		
	0059	18	002220	INC	RD		:POINT TO FLAG
	005A	FD	002230	MOV	A,\$RD		
	005B	F25F	002240	JB7	#+3		:IF TRUE THEN SKIP
	005D	B03C	002250	MOV	RDCLK,#60		:OTHERWISE INITIATE A BEEP
	005F		002260	#:3:			
	005F	FD	002270	MOV	A,CHARBF		:LOAD CODE
	0060	D3E0	002280	XRL	A,#0E0H		:INVERT UPPER 3 BITS
	0062	02	002290	OUT	08B,A		:AND OUTPUT TO BUS PORT
	0063	9ADF	002300	ANL	P2,#0DFH		:SWITCH ON BUSDRIVER
	0065	9AEF	002310	ANL	P2,#0DFH		:AND NOW SET STROBE ACTIVE
	0067	8A10	002320	ORL	P2,#16		:SETOFF STROBE PULSE

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0069	8A20	002330		ORL	P2,#32	;SET BUS TO TRISTATE
	006B		002340	EXINT:			;EXIT TIMER INTERRUPT
	006B	FF	002350		MOV	A,ATEMP	
	006C	93	002360		RETR		
			002370				
	006D		002380	SEND:			
	006D	8B08	002390		MOV	BITCTR,#8	;SET COUNTER FOR 8 BITS
	006F		002400	\$1:			
	006F	9A7F	002410		ANL	P2,#7FH	;SET SYNCH BIT
	0071	1491	002420		CALL	CNTDWN+2	;TIME IS 25 USEC
	0073	FD	002430		MOV	A,CHARBF	
	0074	77	002440		RR	A	;SEND LSB FIRST
	0075	AD	002450		MOV	CHARBF,A	
	0076	F287	002460		JB7	\$4	;HOLD ON BIT
	0078	8A80	002470		ORL	P2,#80H	;ELSE RESET IT
	007A		002480	\$2:			
	007A	1489	002490		CALL	BITIME	;TIME IS NOW 80 USEC
	007C	8A80	002500		ORL	P2,#80H	;RESET NOW UNCONDITIONAL
	007E	148D	002510		CALL	PAUSE	;WAIT NOW 100 USEC
	0080	C8	002520		DEC	BITCTR	
	0081	FB	002530		MOV	A,BITCTR	
	0082	966F	002540		JNZ	\$1	;REPEAT 8 TIMES
	0084	22	002550		IN	A,DBB	;CLEAR IBF FROM TRANSM.ACTIONS
	0085		002560	\$3:			
	0085	046B	002570		JMP	EXINT	
			002580				
	0087	047A	002590	\$4:	JMP	\$2	;ROUTINE LENGTH EGALIZATION
			002600				
	0089		002610	BITIME:			
	0089	8808	002620		MOV	RD,#11	
	008B	048F	002630		JMP	CNTDWN	
			002640				
	008D		002650	PAUSE:			
	008D	88DE	002660		MOV	RD,#14	
	008F		002670	CNTDWN:			
	008F	E88F	002680		DJNZ	RD,\$	
	0091	83	002690		RET		
			002700				
	0092		002710	LDKBQ:			;LOAD A RECOGNIZED CODE INTO FIFO
			002720				
	0092	2383	002730		MOV	A,#0B3H	;SELECT RESET CODE
	0094	DA	002740		XRL	A,R2	
	0095	969B	002750		JNZ	\$1	
	0097	BD1D	002760		MOV	QTPTR,#FIFO	
	0099	BE10	002770		MOV	QHPTR,#FIFO	;SET FIFO EMPTY
	009B		002780	\$1:			
	009B	FD	002790		MOV	A,QTPTR	;TEST FOR PLACE
	009C	17	002800		INC	A	
	009D	5317	002810		ANL	A,#17H	;MASK MODULO 8
	009F	DE	002820		XRL	A,QHPTR	;COMPARE WITH HEAD POINTER
	00A0	C6AB	002830		JZ	\$2	;FULL,IGNORE ENTRY AND INITIATE ERR
			002840				
	00A2	FD	002850		MOV	A,QTPTR	;PUT POINTER INTO RD
	00A3	AB	002860		MOV	RD,A	
	00A4	FA	002870		MOV	A,R2	;GET CODE FOR FIFO

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	00A5	AD	002880		MOV	\$R0,A	
	00A6	FD	002890		MOV	A,@TPTR	;UPDATE POINTER
	00A7	17	002900		INC	A	
	00A8	5317	002910		ANL	A,#17H	;MASK INTO FIFO AREA
	00AA	AD	002920		MOV	@TPTR,A	
	00AB		002930	\$2:			
	00AB	83	002940		RET		
			002950	;			
	00AC		002960	NSPD:			
	00AC	030C	002970		ADD	A,#12	;SELECT F5..FF
	00AE	E6B2	002980		JNC	\$+4	
	00B0	441A	002990		JMP	TRANS-2	
	00B2	2491	003000		JMP	N3	
			003010	;			
			003020	;			*****INITIALIZE SCAN*****
			003030	;			
	00B4		003040	START:			
	00B4	9ABF	003050		ANL	P2,#0BFH	;PREPARE MATRIX READ STRAP INFO
			003060	;			
			003070	;			
	00B6	95	003080		CPL	FO	;PRESET GOOD KB
	00B7	B83F	003090		MOV	RO,#63	;SET RAM MSLOCATION
	00B9		003100	CLEAR:			
	00B9	27	003110		CLR	A	
	00BA	37	003120		CPL	A	
	00BB	AD	003130		MOV	\$R0,A	
	00BC	00	003140		XRL	A,\$R0	;TEST FF
	00BD	C6C0	003150		JZ	\$1	
	00BF	85	003160		CLR	FO	;SET RAM ERROR FLAG
	00C0		003170	\$1:			
	00C0	AD	003180		MOV	\$R0,A	;CLEAR RAM
	00C1	FD	003190		MOV	A,\$R0	;AND TEST
	00C2	C6C5	003200		JZ	\$2	
	00C4	85	003210		CLR	FO	
	00C5		003220	\$2:			
	00C5	E8B9	003230		DJNZ	RO,CLEAR	
	00C7	22	003240		IN	A,DBB	;CLEAR IBF
			003250	;			
			003260	;			;PRESET POINTERS,TIMER
			003270	;			
	00C8	8D10	003280		MOV	@TPTR,#FIFO	;FIFO ADDRESS
	00CA	BE10	003290		MOV	@HPTR,#FIFO	
			003300	;			;DIAGNOSTIC CHECK SUM ROUTINE
	00CC	27	003310	ROMCK:	CLR	A	
	00CD	A8	003320		MOV	RO,A	
	00CE		003330	\$1:			
	00CE	A3	003340		MOVP	A,\$A	;GET A ROM BYTE
	00CF	6A	003350		ADD	A,R2	
	00D0	AA	003360		MOV	R2,A	;SAVE SUM
	00D1	18	003370		INC	RO	
	00D2	F8	003380		MOV	A,RO	;REPEAT 256 TIMES
	00D3	96CE	003390		JNZ	\$1	
	00D5	34F9	003400		CALL	CKSU1	;CONTINUE IN PAGE 1
	00D7		003410	CKSU3:			
	00D7		003420	\$1:			;COMPUTE LAST PAGE

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	00D7	E3	003430		MOV	A,#A	:GET A ROM BYTE
	00D8	6A	003440		ADD	A,R2	
	00D9	AA	003450		MOV	R2,A	:SAVE SUM
	00DA	18	003460		INC	RD	
	00DB	F8	003470		MOV	A,RO	:REPEAT 256 TIMES
	00DC	96D7	003480		JNZ	\$1	
	00DE	FA	003490		MOV	A,R2	:TEST RESULT
	00DF	C6E2	003500		JZ	#+3	:GOOD
	00E1	85	003510		CLR	FO	
	00E2	B82D	003520		MOV	RD,#STRAPS	:CHANGE GOOD FLAG
			003530				:READ NOW STRAPPING INFO
			003540				:
	00E4	09	003550		IN	A,P1	
	00E5	4308	003560		ORL	A,#8	:SET GOOD FLAG
	00E7	B6EB	003570		JFO	#+4	:SKIP
	00E9	53F7	003580		ANL	A,#-9	:CLEAR BITS TO BAD STATUS
	00EB	9AFD	003590		ANL	P2,#-3	:SET PORT2 TO LAST COLUMN+1
	00ED	85	003600		CLR	FO	:CLEAR FLAG AGAINST MULTIPLE KEY ENT
	00EE	A0	003610		MOV	\$RO,A	
	00EF	F2F4	003620		JBT	\$2	:TEST PLL-CABLE PLUGGED
	00F1	05	003630		EN	I	:ELSE ALLOW IBF INTERRUPTS VIA POLL
	00F2	04FA	003640		JMP	\$3	
			003650				:
	00F4		003660				\$2:
	00F4	55	003670		STRT	T	:INITIATE TIMER
	00F5	25	003680		EN	TCNTI	:AND ALLOW TO INTERRUPT
	00F6	B81A	003690		MOV	RD,#26	:POINT TO ERRCNT IN REG.BANK 1
	00F8	B023	003700		MOV	\$RO,#35	:SET BEEP MULTIPLIER
	00FA		003710				\$3:
	00FA	B824	003720		MOV	RD,#DECPO1	:PRESET DECIMAL POINT
	00FC	B02E	003730		MOV	\$RO,#2EH	
	00FE	5306	003740		ANL	A,#6	:SELECT ALL NON-ENGLISH COUNTRIES
	01D0	C604	003750		JZ	#+4	:US+UK SKIP
	01D2	R02C	003760		MOV	\$RO,#2CH	:ELSE CHANGE TO COMMA

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
			003780				;*****ACTUAL SCAN*****
			003790				;
			003800				;A NEW SCAN STARTS AT NEWS.
			003810				;THE 4 LSB'S OF PORT2 SELECTS A COLUMN OF 8 KEYS.
			003820				;PORT1 INPUTS THEN DATA FROM MATRIX ROWS.
			003830				;KEYS ACTUATED IN COLUMN SELECTED RESPONDS WITH
			003840				;LOGICAL 1. DATA IS SAVED IN REGISTER 7.
			003850				;A 2 BIT HISTORY IS RESERVED FOR EACH KEY IN THE
			003860				;MATRIX.LSB WILL BE SET AS LONG A KEY HAS BEEN FOUND
			003870				;ACTIVE, 2ND BIT INDICATES STATUS KEY HAS GONE FIRST
			003880				;TIME DOWN OR UP. PROPER RECOGNITION OF KEY IS AC-
			003890				;CEPTED AFTER FIRST KEY GOING DOWN AND WITH NEXT
			003900				;SCAN CYCLE THIS KEY IS FURTHERON ACTIVE. SUCH WAY
			003910				;NOISE AND DEBOUNCE IS PERFORMED OVER A WHOLE SCAN CYCLE.
			003920				;4 KEYS ARE CONTAINED IN EACH BYTE OF STATUS ARRAY.
			003930				;A TABLE POINTER IS USED FOR GETTING MARKBITS FOR EACH KEY
			003940				;AND ALSO CODE.SYNCHRONIZATION IS FROM TABLEPOINTER TO
			003950				;STATUS POINTER EVERY 4TH KEY AND TO MATRIXCOLUMN EVERY
			003960				;8TH KEY.
			003970				;KEYS ACCEPTED ARE SELECTED IN GROUPS:
			003980				;SHIFT-,CONTROL-,FUNCTION-,ALPHA-,NUMERIC-,SPECIAL-KEYS
			003990				;THE INDICATION WHERE ANY KEY ACCOMPANIES IS CONTAINED IN
			004000				;MARKBIT TABLE.
			004010				;THE SHIFT- AND CONTROL-KEYS ACTIVATED ARE SETTING FLAG#
			004020				; (20H,40H,80H) WHICH ARE STORED AT ADDR.21H.
			004030				;SHIFT CODE 20H IS EXORED WITH UNSHIFT ASCII CODE IF MARK-
			004040				;BIT ENABLES. CONTROL CODE 40H IS EXORED WITH UNSHIFTED
			004050				;ASCII CODE ALSO ENABLED BY MARKBIT.IF BOTH SHIFT AND CTL-
			004060				;FLAG ARE SET AND STRAPPING BIT 4 IS NOT ACTIVE,THEN A 80H
			004070				;IS EXORED WITH ACCORDING ASCII CODE OF KEY.
			004080				;IF WORDPROCESSING FLAG (80H) IS SET ,THEN UPPER AND LOWER
			004090				;CASE CHARACTERS A..Z ARE EXCHANGED SYNCHRONOUS WITH
			004100				;SHIFT POSITION,E.G. LOWER CASE IN UNSHIFT- AND UPPER CASE
			004110				;LETTERS IN SHIFT MODE.IF SHIFT LOCK KLEY IS RELEASED VICE
			004120				;VERSA.
			004130				;FOR COUNTRY VERSIONS WHICH REQUIRES MODIFIED LAYOUTS
			004140				;OF KEY DISPLACEMENTS THE STRAPPING SWITCHES ARE SET TO
			004150				;A SPECIFIED COUNTRY CODE (0...7).BEFORE TRANSLATIONING
			004160				;THE TABLE OFFSET INTO ASCII-CODES, DEPENDING ON STRAP-
			004170				;INFO A SEARCH ROUTINE IS EXERCISED WHETHER THE ACCORDING
			004180				;KEYPOSITION IS CONTAINED IN THE MODIFY TABLE OR NOT.
			004190				;IF TRUE ,THEN THE INSTANT OFFSET IS REPLACED BY ANOTHER
			004200				;AS SPECIFIED.A ZERO CONTENT IN THESE TABLES MEANS END OF
			004210				;TABLE CONDITION AND CODE REMAINS UNCHANGED.
			004220				;MOST KEYS ARE AUTO REPEATABLE ELSE ASSOCIATED MARKBIT
			004230				;DETERMINES OTHERWISE. AFTER KEY FIRST GOING DOWN A LONG
			004240				;REPEAT DELAY IS SET(800MS).IF KEY IS HOLD DOWN FURTHERON
			004250				;THEN ACCORDING ASCII CODES ARE ISSUED DEPENDING ON
			004260				;SHIFT AND CONTROL KEY STATUS,WITH AN REPEAT RATE OF
			004270				;APPROX. 12 HZ.
			004280				;CODES TO BE ISSUED TO A MAIN STATION ARE FIRST PUT INTO
			004290				;A QUEUE WHICH WORKS LIKE A FIFO(FIRST IN FIRST OUT)
			004300				;STORAGE BUFFER.
			004310				;THIS USEFUL FOR THE N-KEY ROLL OVER FEATURE WITH FAST
			004320				;KEY DEPRESSIONS AND GIVES THE MAIN STATION SUFFICIENT

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
			004330				;TIME TO POLL THE DATA FROM KEYBOARD AND INCREASES PRO-
			004340				;CESSOR THRUPUT BY NONINTERUPTIVE INTERFACING.
			004350				;IF FIFO GOES OVERFLOW A LONG ERROR TONE IS INITIATED
			004360				;LAST CODE ENTERED IS LOST!
			004370				;THE KEYS IN THE F-ROW ARE PROVIDED FOR FUNCTION KEY
			004380				;ASSIGNMENTS ITS CODES ARE FOR FREE PROGRAMMING AND
			004390				;ALL SHIFT MODES ARE ALLOWED.
			004400				;
			004410	NEWS:			
0104			004420		MOV	STPTR,#64	;TOP STATUS BYTE
0104	B940		004430		MOV	TBLPTR,#KEYS	;LAST KEY ADDRESS
0106	BB68		004440		DEC	REPDLY	;COUNT DOWN REPEAT DELAY EVERY NEW S
0108	CC		004450		ORL	P2,#13	
0109	8A00		004460	SCAN:			
0108			004470		CLR	C	;RESET CARRY FOR POSSIBLE ROTATES TH
0108	97		004480		DEC	TBLPTR	
010C	CB		004490		MOV	A,TBLPTR	
010D	FB		004500		J87	NEWS	;TABLE POINTER UNDER FLOW
010E	F204		004510		CPL	A	;TEST BIT0+1 FOR STATUSPOINTER SYNC
0110	J7		004520		J80	\$1	
0111	1221		004530		J81	\$1	
0113	J221		004540		DEC	STPTR	;POINT TO NEXT 4KEY GROUP
0115	C9		004550		J82	\$1	;ODD,SKIP MULTIPLEXER UPDATE
0116	5221		004560		STOP	TCNT	;TISR MAY NOT OCCUR IN NEXT ROUTINE
0118	65		004570		IN	A,P2	;ELSE D0
0119	0A		004580		DEC	A	
011A	07		004590		OUTL	P2,A	
011B	3A		004600		STRT	T	;LET TIMER RUN AGAIN
011C	55		004610		MOVP	A,\$A	;SPEND A SHORT DELAY
011D	A3		004620		IN	A,P1	;READ NEW MATRIX ROW DATA
011E	09		004630		CPL	A	
011F	J7		004640		MOV	R7,A	;SAVE INFO FOR SHIFT AROUND
0120	AF		004650	\$1:			
0121			004660		MOV	A,R7	;ROTATE KEY DATA SYNC KEY TO SAME BI
0122	FF		004670		RL	A	
0123	E7		004680		MOV	R7,A	
0124	AF		004690		MOV	A,\$STPTR	;ROTATE ALSO STATUS
0125	F1		004700		RL	A	
0126	E7		004710		RL	A	
0127	A1		004720		MOV	\$STPTR,A	
0128	324A		004730		J81	UNLOCK	
012A	1238		004740		J80	KYDOWN	
012C	FF		004750		MOV	A,R7	;TEST KEY IS DOWN
012D	1208		004760		J80	SCAN	;NO
012F	B608		004770		JFO	SCAN	;INHIBIT MULTIPLE KEY ENTRIES
0131	95		004780		CPL	FO	;ELSE SET INHIBITOR
0132			004790	LOCKY:			
0132	F1		004800		MOV	A,\$STPTR	;ELSE SET STATUS BITS
0133	D303		004810		XRL	A,#3	
0135	A1		004820		MOV	\$STPTR,A	
0136	2408		004830		JMP	SCAN	
0138			004840	KYDOWN:			
0138	FC		004850		MOV	A,REPDLY	;LOCK OUT KEY NEXT 8 SCANS
0139	53F8		004860		ANL	A,#-8	
0138	D358		004870		XRL	A,#58H	

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0130	C60B	004880		JZ	SCAN	;DON'T CHECK KEY ON ACTIVE
	013F	FF	004890		MOV	A,R7	;TEST KEY HOLD DOWN
	0140	37	004900		CPL	A	
	0141	12A2	004910		JBO	REPCK	;THEN CHECK REPEAT COND.MET
	0143	FB	004920		MOV	A,TBLPTR	;ELSE FIRST KEY UP CONDITION
	0144	E3	004930		MOVJP3	A,\$A	;SELECT SHIFT/CTL KEYS
	0145	F7	004940		RLC	A	
	0146	F6F2	004950		JC	RESHFG	;RESET SHIFT FLAGB
	0148	2432	004960		JMP	LOCKY	;ALL OTHERS KEYUP DEBOUNCE
	014A		004970	UNLOCK:			
	014A	85	004980		CLR	FD	;CLEAR MULTIPLE KEY INHIBITOR
	014B	F1	004990		MOV	A,\$STPTR	;CLEAR DEBOUNCE LOCK
	014C	0302	005000		XRL	A,#2	
	014E	A1	005010		MOV	\$STPTR,A	
	014F	37	005020		CPL	A	
	0150	120B	005030		JBO	SCAN	;WAS KEYUP UNLOCK
	0152	FF	005040		MOV	A,R7	;TEST KEY STILL DOWN
	0153	1232	005050		JBO	LOCKY	;NO
	0155		005060	ACCPCKY:			
	0155	BC60	005070		MOV	REPLY,#LONGDLY	;SET UP 600MS DELAY
	0157		005080	REPEAT:			
	0157	B821	005090		MOV	RO,#FLAGB	
	0159	F0	005100		MOV	A,\$RO	
	015A	37	005110		CPL	A	
	015B	5360	005120		ANL	A,#60H	
	015D	C60B	005130		JZ	SCAN	IGNORE CTL/SHIFT COINCIDENCE
	015F	B823	005140		MOV	RO,#LASTKY	;POINT TO LASTKY
	0161	FB	005150		MOV	A,TBLPTR	
	0162	A0	005160		MOV	\$RO,A	
	0163	5456	005170		CALL	NATVING	;CHECK NATIVE LANGU.XLAT
	0165	B822	005180		MOV	RO,#LASTCD	;INTERIM SAVE
	0167	A0	005190		MOV	\$RO,A	
	0168	C8	005200		DEC	RO	;POINT TO FLAGB
	0169	F2E0	005210		JB7	SPZEXC	
	016B	E3	005220		MOVJP3	A,\$A	;GET MARKBIT INFO
	016C	AA	005230		MOV	R2,A	;SAVE INTERIM
	016D	F2B3	005240		JB7	FUKY	;SELECT FUNCTION- SHIFT- CTL-LOCK-KE
	016F	531F	005250		ANL	A,#31	;TEST EXCEPTION INDICATION
	0171	9689	005260		JNZ	EXCEPT	;IF 50 TEST LANGUAGE,SHIFT COND.
	0173	FA	005270		MOV	A,R2	;RELOAD MARKS
	0174		005280	NORML5:			;ALL OTHERS HANDLE NORMALLY
	0174	D360	005290		XRL	A,#60H	;INVERT DISABLES TO ENABLES
	0176	AA	005300		MOV	R2,A	;PRESAVE
	0177	F0	005310		MOV	A,\$RO	;GET FLAGB AGAIN
	0178	5A	005320		ANL	A,R2	;COINCIDE WITH MARKS
	0179		005330	NO:			
	0179	AA	005340		MOV	R2,A	;FLAG IS CODE CHANGE OPERAND
	017A	18	005350		INC	RO	;GET LAST KEY AGAIN
	017B	F0	005360		MOV	A,\$RO	
	017C	036D	005370		ADD	A,#ASCITB	;RELATE TO ASCII TABLE
	017E	E3	005380		MOVJP3	A,\$A	;GET UNSHIFTED CODE
	017F	D28B	005390		JB6	\$1	;SELECT LETTER CODES SKIP SHIFT/CTL
	0181	2A	005400		XCH	A,R2	
	0182	C68B	005410		JZ	\$1	
	0184	B289	005420		JB5	\$1-2	;SELECT SHIFTS

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0186	FA	005430		MOV	A,R2	;TEST CTL FOR 20..3F CODES
	0187	B20B	005440		JB5	SCAN	;IF TRUE IGNORE
	0189	2310	005450		MOV	A,#10H	;MAY BE CHANGED TO 10
	018B		005460	#1:			
	018B	DA	005470		XRL	A,R2	;MAY CHANGE CODE ACCORDING SHIFT MO
	018C		005480	N1:			
	018C	AA	005490		MOV	R2,A	
	018D	F29C	005500		JB7	N4	;EXTRACT FUNCTIONAL KEYS
	018F	542C	005510		CALL	TXTCHG	;TEST UPPER/LOWER CASE LETTER EXCHAN
	0191		005520	N3:			
	0191	1492	005530		CALL	LDKB@	;PUT IT NOW INTO FIFO
			005540				
	0193	FB	005550		MOV	A,TBLPTR	
	0194	D30B	005560		XRL	A,#DOUBLO	;SELECT DOUBLE ZERO KEY
	0196	960B	005570		JNZ	SCAN	
	0198	1492	005580		CALL	LDKB@	;IF TRUE OUTPUT A 2ND 0
	019A	240B	005590		JMP	SCAN	;NEXT KEY
			005600				
	019C		005610	N4:			
	019C	D2A0	005620		JB6	N5	
	019E	440B	005630		JMP	COND	;SELECT DEC.POINT FEATURE
			005640				
	01A0	04AC	005650	N5:	JMP	N5PO	;GO INTO PAGE 2
			005660				
			005670				
			005680				;CHECK ON REPEAT FUNCTION
			005690				REPCK:
	01A2		005700		MOV	A,REPLY	;TEST COUNTER ZERO
	01A2	FC	005700		MOV	A,REPLY	;TEST COUNTER ZERO
	01A3	960B	005710		JNZ	SCAN	;NO,IGNORE
	01A5	FB	005720		MOV	A,TBLPTR	
	01A6	E3	005730		MOVPI3	A,\$A	;GET MARKBITS
	01A7	F20B	005740		JB7	SCAN	;EXIT IF NON REPEATABLE KEYS
			005750				
	01A9		005760	#1:			
	01A9	B823	005770		MOV	RO,#LASTKY	;GET LAST KEY
	01AB	F0	005780		MOV	A,\$RO	
	01AC	0B	005790		XRL	A,TBLPTR	;IS INSTANT KEY=LAST KEY OUTPUT
	01AD	960B	005800		JNZ	SCAN	
	01AF	B00B	005810		MOV	REPLY,#REPRAT	;SET REPEAT RATE AGAIN
	01B1	2457	005820		JMP	REPEAT	
			005830				
	01B3		005840	FUKY:			
	01B3	F7	005850		RLC	A	
	01B4	96EE	005860		JNZ	STSHF6	;SET SHIFT,OR CTL OR ALPHALOCK FLAG
	01B6	27	005870		CLR	A	
	01B7	2474	005880		JMP	NORML5	;RECEIVE WITH 0 MARKS
			005890				
	01B9		005900	EXCEPT:			;TEST ON EXCEPTIONAL CONDITIONS
	01B9	FA	005910		MOV	A,R2	;SELECT SPECIAL EXCHANGES
	01BA	D2DF	005920		JB6	SPZEXC-1	
	01BC	00	005930		XRL	A,\$RO	;TEST SHIFT-COND COINCIDENCE
	01BD	D2B6	005940		JB6	EXCEPT-3	;HANDLE NORMAL IN CTL-MODE
	01BF	B2B6	005950		JB5	EXCEPT-3	;HANDLE NORMAL NO SHIFT COINCIDENCE
	01C1	C8	005960		DEC	RO	;LOAD LANGUAGE
	01C2	F0	005970		MOV	A,\$RO	

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	01C3	5307	005980		ANL	A,#7	
	01C5	A8	005990		MOV	RO,A	
	01C6	18	006000		INC	RO	;CLUDGE
	01C7	2380	006010		MOV	A,#128	
	01C9		006020	\$1:			
	01C9	E7	006030		RL	A	;MAKE MASK
	01CA	E8C9	006040		DJNZ	RO,\$1	
	01CC	A8	006050		MOV	RO,A	;SAVE RESULT
	01CD	FA	006060		MOV	A,R2	
	01CE	531F	006070		ANL	A,#31	
	01D0	03CB	006080		ADD	A,#MSKTB-1	;REFER TO LANGUAGE MASKTABLE
	01D2	AA	006090		MOV	R2,A	;SAVE INTERIM
	01D3	546C	006100		CALL	XL	
	01D5	58	006110		ANL	A,RO	;MASK NOW
	01D6	B821	006120		MOV	RO,#FLAGB	;RELOAD POINTER
	01D8	C674	006130		JZ	NORMLS	;LANGUAGE NOT CONTAINED
	01DA	FA	006140		MOV	A,R2	
	01DB	031A	006150		ADD	A,#XLCD-MSKTB	;CALCULATE EXCEPT.CODE OFF
	01D0	441C	006160		JMP	TRANS	;GET CODE FROM NEXT PAGE
			006170	:			
	01DF	E7	006180		RL	A	
	01E0		006190	SPZEXC:			
	01E0	AA	006200		MOV	R2,A	
	01E1	FD	006210		MOV	A,\$RO	;GET FLAGB
	01E2	536D	006220		ANL	A,#60H	
	01E4	77	006230		RR	A	
	01E5	47	006240		SWAP	A	
	01E6	6A	006250		ADD	A,R2	
	01E7	6A	006260		ADD	A,R2	
	01E8	6A	006270		ADD	A,R2	
	01E9	035A	006280		ADD	A,#SPZXC0-128	
	01EB	E3	006290		MOV3	A,\$A	
	01EC	441D	006300		JMP	TRANS+1	
			006310	:			
	01EE		006320	STSHFG:			
	01EE	4D	006330		ORL	A,\$RO	;SET UP BIT
	01EF	A0	006340		MOV	\$RO,A	;AND STORE
	01F0	240B	006350		JMP	SCAN	
			006360	:			
	01F2		006370	RESHFG:			
	01F2	B821	006380		MOV	RO,#FLAGB	
	01F4	37	006390		CPL	A	
	01F5	50	006400		ANL	A,\$RO	;CLEAR FLAG
	01F6	A0	006410		MOV	\$RO,A	;STORE BACK
	01F7	2432	006420		JMP	LOCKY	
	01F9		006430		ORG	1F9H	
	01F9		006440	CKSU1:			
	01F9		006450	\$1:			
	01F9	A3	006460		MOV3	A,\$A	;GET A ROM BYTE
	01FA	6A	006470		ADD	A,R2	
	01FB	AA	006480		MOV	R2,A	;SAVE SUM
	01FC	18	006490		INC	RO	
	01FD	F8	006500		MOV	A,RO	;REPEAT 256 TIMES
	01FE	96F9	006510		JNZ	\$1	
	0200		006520	CKSU2:			

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0200		006530	\$1:			
	0200	A3	006540		MOVP	A,\$A	
	0201	6A	006550		ADD	A,R2	
	0202	AA	006560		MOV	R2,A	;SAVE SUM
	0203	18	006570		INC	RD	
	0204	F8	006580		MOV	A,RO	;REPEAT 256 TIMES
	0205	9600	006590		JNZ	\$1	
	0207	83	006600		RET		
	0208		006610	COND:			
	0208	B824	006620		MOV	RD,#DECP01	
	020A	D38A	006630		XRL	A,#8AH	;SELECT DECIMAL POINT ORIGINAL CODE
	020C	9611	006640		JNZ	\$1	
	020E	F0	006650		MOV	A,\$RO	;GET POINT OR COMMA
	020F	4410	006660		JMP	TRANS+1	
	0211	D310	006670	\$1:	XRL	A,#16	;SELECT ^DECP01
	0213	961E	006680		JNZ	TRANS+2	;IS ANOTHER FUNCTIONCODE
	0215	F0	006690		MOV	A,\$RO	
	0216	D302	006700		XRL	A,#2	;SWAP POINT TO COMMA OR BACK
	0218	24EF	006710		JMP	STSHFG+1	;STORE AND GO SCAN
	021A	0320	006720		ADD	A,#SUPLFU	
	021C		006730	TRANS:			
	021C	A3	006740		MOVP	A,\$A	
	021D	AA	006750		MOV	R2,A	
	021E	2491	006760	\$1:	JMP	N3	
			006770	;			
			006780	;			
	0220		006790	SUPLFU:			
	0220	343536373839	006800		DB	'456789**123/'	
		2A2B3132332F					

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	022C		006820	TXTCHG:			
	022C	37	006830		CPL	A	
	022D	0235	006840		JB6	UNCH	:NO LETTER CODE
	022F	C8	006850		DEC	RO	
	0230	FD	006860		MOV	A,\$RO	:GET WP-FLAG
	0231	37	006870		CPL	A	
	0232	F236	006880		JB7	UNCH+1	:TRUE
	0234	FA	006890		MOV	A,R2	
	0235		006900	UNCH:			
	0235	83	006910		RET		
	0236		006920	\$1:			
	0236	C8	006930		DEC	RO	
	0237	FA	006940		MOV	A,R2	
	0238	531F	006950		ANL	A,#1FH	
	023A	C635	006960		JZ	UNCH	
	023C	03E2	006970		ADD	A,#-30	:SELECT ALL } U
	023E	F635	006980		JC	UNCH	
	0240	0303	006990		ADD	A,#3	:SELECT ALL } Z
	0242	E651	007000		JNC	\$3	
	0244	FD	007010		MOV	A,\$RO	:GET LANGUAGE CODE
	0245	07	007020		DEC	A	
	0246	07	007030		DEC	A	
	0247	37	007040		CPL	A	
	0248	5251	007050		JB2	\$3	
	024A		007060	\$2:			
	024A	37	007070		CPL	A	
	024B	07	007080		DEC	A	
	024C	5235	007090		JB2	UNCH	
	024E	FA	007100		MOV	A,R2	
	024F	1235	007110		JB0	UNCH	:CAN'T BE 50 OR 70(FOR SPAIN
	0251		007120	\$3:			
	0251	2320	007130		MOV	A,#32	:SET EXCHANGE OPERATOR
	0253	DA	007140		XRL	A,R2	
	0254	AA	007150		MOV	R2,A	
	0255	83	007160		RET		

ERR-C	ADDR	OBJ-CO	PG-LIN	LABEL	CMD	OPERAND	COMMENT
				007180	;		
				007190	;	LANGUAGE DEPENDENT KEY LOCATION EXCHANGE	
				007200	;		
				007210	;	US(0) UK(1) Dk(2) Ge(3) Sw(4) Nw(5) Sp(6) It(7)	
				007220	;		
				007230	;	E7 ^X1	D6 ^B1 D2 ^B1
				007240	;	E10^X2 E10^S6	B1 ^D6 B1 ^D2
				007250	;	E11^B10 E11^E10	E11^X6 E11^X9 E11^X7
				007260	;	E12^X3 E12^B^C12	E12^C11 E12^S9
				007270	;	B00^C10 B00^C10	
				007280	;	D11^C11 D11^E12	D11^C12
				007290	;	D12^D11 D12^C11 D12^C12	D12^C12
				007300	;	C10^X4 C10^X4 C10^C11	C10^S2
				007310	;	C11^S5 C11^D12 C11^C10	C11^E12 C11^S4
				007320	;	C12^D11 C12^X8 C12^S11 C12^X8	C12^D11 C12^S7
				007330	;	B10^E11 B10^E11	
				007340	;		
				007350	NATVLNG:		
0256				007360	MOV	R2,A ;SAVE TABLE OFFSET	
0256	AA			007370	MOV	RO,#STRAPS	
0257	B820			007380	MOV	A,#RO ;GET LANGUAGE CODE	
0259	F0			007390	ANL	A,#7	
025A	5307			007400	ADD	A,#LNGOFS ;GET OFFSET TO TABLES	
025C	D36E			007410	MOVP	A,#A	
025E	A3			007420	MOV	RO,A ;SAVE IN REG	
025F	A8			007430	;		
0260				007440	MOV	A,RO	
0260	F8			007450	MOVP	A,#A ;ADDRESS INTO PROG.MEMORY	
0261	A3			007460	JZ	UNCH-1 ;END OF TABLE REACHED	
0262	C634			007470	XRL	A,R2 ;COMPARE WITH ORIGINAL DISPLACEMENT	
0264	DA			007480	INC	RO	
0265	18			007490	JZ	\$2 ;FOUND CONDITION	
0266	C66B			007500	INC	RO ;SKIP TO NEXT CODE	
0268	18			007510	JMP	\$1 ;LOOP	
0269	4460			007520	;		
026B				007530	MOV	A,RO ;ADDRESS TO SUBSEQUENT CODE	
026C	A3			007540	XL: MOVP	A,#A ;LOAD REPLACE CODE	
026D	83			007550	RET		
026E				007560	LNGOFS:		
026E	768BA9ADADA0			007570	DB	\$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8	
	B2B0			007580	;		
0276				007590	DB	78,52,94,074,62,59,10,73,93,89,61,93	;US-ENGLISH
0276	4E345E4A3E3B			007600	;		
	DA495059305D			007610	DB	73,60,89,133,74,094,86,51,0	E10X2,E11B10,E12X3,B00C10,D11C11,D12D11
0282	493C59854A5E			007620	;		
	563300			007630	;		
028B				007640	DB	78,134,94,78,10,73,93,62,61,89,73,60	C10X4,C11S5,B10E11,E7X1 ;UK+INT'L ENGLISH
0288	4E865E4E0A49			007650	;		
	503E3059493C			007660	DB	89,61,57,93,74,94,62,57,0	E10S6,E11E10,B00C10,D11E12,D12C11,C10X4
0297	593D395D4A5E			007670	;		
	3E3900						

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	02A0		007680	\$4:			;GERMAN AUSTRIAN
	02A0	452A2A455E69	007690		DB	69,42,42,69,94,105,57,139,0	
		398B00					
			007700	;		D6B1,8106,E11X6,C12S11	
			007710	;			DANISH
	02A9	49595949	007720	\$3:	DB	73,89,89,73	
			007730	;			SWEDISH,FINNISH
	02A0		007740	\$5:			
	02A0		007750	\$6:			NORWEGIAN
	02A0	3039396B00	007760		DB	61,57,57,107,0	
			007770	;		D12C12,C12X8	
	02B2		007780	\$7:			SPANISH
	02B2	5039593E395D	007790		DB	93,57,89,62,57,93,62,89,94,108,0	
		3E595E6C00					
			007800	;		D11C12,C11E12,C12D11,E12C11,E11X9	
	02B0		007810	\$8:			ITALIAN
	02B0	49825984252A	007820		DB	73,130,89,132,37,42,42,37,62,137	
		2A253E89					
			007830	;		C10S4,C11S6,D2B1,B1D2,E12S10	
	02C7	5E6A398700	007840		DB	94,106,57,135,0	
			007850	;		E11X7,C12S7	
	02CC		007860	MSKTB:			
	02CC	FCFCFCFCFC8	007870		DB	-4,-4,-4,-4,-4,-56	
	02D2	FF0008344008	007880		DB	-1,0,8,052,64,8,124,64,128	
		7C4080					
	02DB	0009	007890		DS	9,1	
	02E4	3440	007900		DB	52,64	;SKANDIN.,SPAIN
	02E6		007910	XLCD:			
	02E6	2F303E383A2B	007920		DB	'/O>::+U\$+uB',39,123,39,'\$*8*(+): '\$\$'	
		7F5D402B5D7E					
		277827405E26					
		2A2B292B3A60					
		4040					

```

007940 ; EXCEPTIONAL KEYCODE MODIFICATION EXPLANATION
007950 ; LANGUAGE CODE AND BIT ASSIGNMENT:
007960 ; 0=US-ENGLISH(BIT0=1) 1=UK+INT'L ENGLISH(BIT1=2)
007970 ; 2=DANISH (BIT2=4) 3=GERMAN,AUSTRIAN (BIT3=8)
007980 ; 4=SWED/FINN.(BIT4=16) 5=NORWEGIAN (BIT5=32)
007990 ; 6=SPANISH (BIT6=64) 7=ITALIAN (BIT7=128)
008000 ;
008010 ; OFFS MARK LANGU. EXC.COD -SYMB. NORM-CD -SYMB. POSITION
008020 ;
008030 ; 1 21 FC 2F 7 / 37 2F 7 ' E7
008040 ; 2 02 FC 30 0 = 20 30 - = E10
008050 ; 3 23 FC 3E < > 3C 3E < > B00
008060 ; 4 24 FC 3B , ; 2B 3B , < B8
008070 ; 5 25 FC 3A . : 2E 3E . > B9
008080 ; 6 06 C8 2B + * 3A 2A : * D12,C11
008090 ; 7 27 FF 7F CLR 86 7F 96 CLR 050
008100 ; 8 28 00 50 3Par 33 23 3 # E3,X5
008110 ; 9 29 08 40 3Par 33 23 3 # E3
008120 ; 10 0A 34 2B + ? 2F 3F / ? E11,B10
008130 ; 11 28 40 50 i ?/ 5B 7B 1B ä ä E12,C11
008140 ; 12 0C 08 7E ss ? 2F 3F / ? E11,X6
008150 ; 13 00 7C 27 ' ' 40 60 00 § ' E12,C11
008160 ; 14 0E 40 7B 'ee'ä ü 50 7D 1D ü ü D11,C12
008170 ; 15 0F 80 27 ' ? 2F 3F / ? E11,X7
008180 ; 16 30 01 40 2 § 32 22 2 " E2
008190 ; 17 31 01 5E 6 ^ 36 26 6 & E6
008200 ; 18 32 01 26 7 & 37 27 7 ' E7,X1
008210 ; 19 33 01 2A 8 * 38 28 8 ( E8
008220 ; 20 34 01 28 9 ( 39 29 9 ) E9
008230 ; 21 35 01 29 0 ) 30 0 E10,X2
008240 ; 22 36 01 2B = + 30 2D = - E12,X3
008250 ; 23 37 01 3A ; ; 3B 2B ; + C10,X4
008260 ; 24 38 01 60 ' \B 5E 7E 1E ^ B D12,C12
008270 ; 25 19 34 40 § # 3A 2A : # C12,X8
008280 ; 26 1A 40 40 § ? 2F 3F / ? E11,X9
008290 ;
008300 ; SPECIAL EXCEPTIONS:
008310 ;
008320 ; 00 80 04 60 23 00 ' # C12,S0
008330 ; 01 81 04 40 5C 1C a'c, D11,S1
008340 ; 02 82 80 78 5C 1C a'c, C10,S2
008350 ; 03 83 04 7C 5B 1B u', C11,S3
008360 ; 04 84 80 60 5B 1B u', C11,S4
008370 ; 05 85 01 27 22 00 ' " C11,S5
008380 ; 06 86 02 30 5F 1F 0 _ E10,S6
008390 ; 07 87 80 7C 40 00 o'Par C12,S7
008400 ; 08 44 FD 20 5F 1F - _ B10,E11
008410 ; 09 89 80 7E 5E 1E i'^A E12,S9
008420 ; 10 45 FF 20 F4 F5 - B51
008430 ; 11 8B 08 23 5E 1E # ^ C12,S11
008440 ;
008450 ; MARKBIT 5 ON MEANS EXCEPT IN SHIFTMODE,OFF MEANS EXCEPT UNS
008460 ; MARKBIT 6 OR 7 ON MEANS EXCEPT IN ALL MODES

```


ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0300		008480		ORG	300H	
	0300		008490	MRKTB1:			
	0300	A00000000000	008500		DB	160,,,,,,128	:D13CB4D51D53DEF4
		0080					
	0308	60C0236020A0	008510		DB	96,-64,35,96,32,160,96,128	:A50CB0A5253DEF0
		6080					
	0310	270000000000	008520		DB	39,,,,,,41,128	:D50CB3D52540EF3
		2980					
	0318	600000000000	008530		DB	96,,,,,,128	:ACB5C5153DEF5
		0080					
	0320	900000002000	008540		DB	144,,,32,,48,128	:899CB2B51540EF2
		3080					
	0328	000000000000	008550		DB	0,,,,,,128	:850CB185253DEF1
		0080					
	0330	200025323500	008560		DB	32,,37,50,53,,52,128	:E13CB9X1X2DEF9
		3480					
	0338	801820363706	008570		DB	128,24,32,54,55,6,13,128	:F13C1213X3X4DEF
		0080					
	0340	450000000000	008580		DB	69,,,,,,49,128	:C50CB6C52540EF6
		3180					
	0348	800044802000	008590		DB	128,,68,128,20H,,2,128	:F50CB10FE520EF1
		0280					
	0350	600000802000	008600		DB	96,,,128,32,,33,128	:E14CB7FE53DEF7
		2180					
	0358	80289080200E	008610		DB	128,43,144,128,32,14,10,128	:F14CB11FE510EF1
		0A80					
	0360	200024802000	008620		DB	32,,36,128,32,,51,128	:E50CB8FE540EF8
		3380					
	0068		008630	KEYS	EQ0	\$(MRKTB1	
	0368	280C0F191A	008640		DB	40,12,15,25,26	:X5..9
	0360		008650	ASCITB:			
	0360	004656F7F952	008660		DB	0,'FV',-9,-7,'R4',-28	:D13CB4D5153DEF4
		34E4					
	0375	30003C308A00	008670		DB	48,'0',8AH,,27,-32	:A50CB0A5253DEF0
		18E0					
	0370	864443F8FF45	008680		DB	134,'0C',-8,-1,'E3',-29	:D50CB3D52540EF3
		33E3					
	0385	204742F4F654	008690		DB	'68',-12,-10,'T5',-27	:ACB5C5153DEF5
		35E5					
	0380	005358FC8857	008700		DB	'SX',-4,88H,'W2',-30	:899CB2B51540EF2
		32E2					
	0395	F8415AFDFE51	008710		DB	-5,'AZ',-3,-2,'01',-31	:850CB185253DEF1
		31E1					
	0390	88422E37304F	008720		DB	139,'L,7009',-23	:E13CB9X1X2DEF9
		39E9					
	03A5	E05E88303B3A	008730		DB	-19,94,136,'=:5',-20	:F13C1213X3X4DEF12
		40EC					
	03A0	20484EF5FA59	008740		DB	45,'HN',-11,-6,'Y6',-26	:C50CB6C52540EF6
		36E6					
	0385	EF5C20F18350	008750		DB	-17,'0',-15,131,80,45,-22	:F50CB10FE520EF10
		2DEA					
	03B0	094A40F28455	008760		DB	9,'JM',-14,84H,'U7',-25	:E14CB7FE53DEF7
		37E7					
	03C5	EE5800F0825D	008770		DB	-18,91,,-16,130,93,47,-21	:F14CB11FE510EF1
		2FEB					

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	03C0	814B2CF38549	008780		DB	81H,'K',-13,85H,'I8',-24	:E50C88FE540EF838E8
	03D5	332F2F3A2F	008790		DB	'3//:'	
	03DA		008800	SPZXC:			
	03DA	602300405C1C	008810		DB	96,35,,64,92,28,123,92,28,124,91,27	
		785C1C7C5818					
	03E6	605818272200	008820		DB	96,91,27,39,34,,48,95,31,124,64,0,45,95,31	
		305F1F7C4000					
		205F1F					
	03F5	7E5E1E20F4F5	008830		DB	126,94,30,45,-12,-11,35,94,30	
		235E1E					
	03FE	00	008840	CHKSUM:	DB	0	
			008850				
			008860				
			008870	:CROSS REFERENCE		WIRING/KEY LOCATIONS	
			008880				
			008890	: COLUMN/ROW	RD	R1 R2 R3 R4 R5 R6 R7	
			008900				
			008910	: 0 COL 0	D13	C4 B4 D51 D53 D4 E4 F4	
			008920				
			008930	: 1 COL 1	A50	C0 B80 A52 +A53 D0 E0 F0	
			008940				
			008950	: 2 COL 2	+D50	C3 B3 D52 D54 D3 B3 F3	
			008960				
			008970	: 3 COL 3	A5	C5 B5 C51 C53 D5 E5 F5	
			008980				
			008990	: 4 COL 4	B99	C2 B2 B51 +B54 B02 B2 F2	
			009000				
			009010	: 5 COL 5	B50	BC1 B81 B52 B53 D1 E1 F1	
			009020				
			009030	: 6 COL 6	+E13	C9 B89 X1 X2 D9 B9 F9	
			009040				
			009050	: 7 COL 7	F13	BC12 +C13 X3 X4 B012 B2 F12	
			009060				
			009070	: 8 COL 8	C50	C6 B6 C52 C54 B06 B6 F6	
			009080				
			009090	: 9 COL 9	F50	BC10 B810 F52 +E52 D10 B2 F10	
			009100				
			009110	: A COL 10	E14	C7 B7 F53 +E53 D7 B7 F7	
			009120				
			009130	: B COL 11	F14	BC11 B11 F51 +E51 B011 B2 F11	
			009140				
			009150	: C COL 12	+E50	C8 B88 F54 +E54 D8 B8 F8	
			009160				
			009170	: PSEUDO 12	X5	X6 X7	
			009180				
			009190	: KEY LOCATIONS WITH B		ARE RELOCATED LANGUAGE DEPENDENT	
			009200	: KEY LOCATIONS WITH +		TRANSLATABLE FOR 7900,7901 USAGE	

ERR-C	ADDR	OBJ-CO	PG-LIN	LABEL	CMD	OPERAND	COMMENT
			009220			;NUMERIC KEY PAD PART	
			009230			;EXPLANATION: 1ST LINE KEYLOCATION 2ND MATRIXPOSITION	
			009240			3RD LINE MARKBITS	4TH SYMBOLS
			009250			5TH LINE CTL-CODE, 6TH SHIFT-CODE, 7TH UNSHIFT	
			009260			-----	MATRIXCODE COMBINED:
			009270			!F50!F51!F52!F53!F54!	HO-DIGIT COLUMN#
			009280			! 90! 83! 93! A3! C3!	LO-DIGIT ROW-#
			009290			! 80! 80! 80! 80! 80!	
			009300			!F16!F17!F18!F19!F20!	
			009310			! AF! B0! B1! B2! B3!	
			009320			! CF! D0! D1! D2! D3!	
			009330			! EF! F0! F1! F2! F3!	
			009340			!-----!	
			009350			!E50!E51!E52!E53!E54!	
			009360			! C0! B4! 94! A4! C4!	
			009370			! 20! 20! 20! 20! 20!	
			009380			Home!(<--!Dwn!Up!-->!	
			009390			! 91! 92! 93! 94! 95!	
			009400			! 81! 82! 83! 84! 85!	
			009410			! 81! 82! 83! 84! 85!	
			009420			!-----!	
			009430			!D50!D51!D52!D53!D54!	
			009440			! 20! 03! 23! 04! 24!	
			009450			! 27! 00! 00! 00! 00!	
			009460			!CLR!7 !8 !9 !/ !	
			009470			! 96! B7! B8! B9! BF!	
			009480			! 7F! D7! D8! D9! DF!	
			009490			! 86! 37! 38! 39! 2F!	
			009500			!-----!	
			009510			!C50!C51!C52!C53!C54!	
			009520			! 80! 33! 83! 34! 84!	
			009530			! 00! 00! 00! 00! 00!	
			009540			!- !4 !5 !6 !* !	
			009550			! F5! B4! B5! B6! BA!	
			009560			! F4! D4! D5! D6! DA!	
			009570			! 20! 34! 35! 36! 2A!	
			009580			!-----!	
			009590			!B50!B51!B52!B53!B54!	
			009600			! 50! 43! 53! 54! 44!	
			009610			! 00! 00! 00! 00! 20!	
			009620			!+ !1 !2 !3 !FT!	
			009630			! 88! BC! BD! BE! 98!	
			009640			! D8! DC! DD! DE! 88!	
			009650			! 28! 31! 32! 33! 88!	
			009660			!-----!	
			009670			!A50 !A52!A53!	
			009680			! 10 ! 13! 14! !	
			009690			! 60 ! 60! 20! !	
			009700			!0 !00 !. ! !	
			009710			! 30 !*30! 9A!<-----used to switch fm . to , or back	
			009720			! 30 !*30! 8A!<-----OUTPUT: 2EH(.) or 2CH(,) as set	
			009730			! 30 !*30! 8A! !	
			009740			-----	

```

009760 ; US-ENGLISH
009770 ;EXPLANATION: 1ST LINE KEYLOCATION 2ND MATRIXPOSITION
009780 ; 3RD LINE MARKBITS 4TH SYMBOLS
009790 ; 5TH LINE CTL-CODE, 6TH SHIFT-CODE, 7TH UNSHIFT
009800 ;-----
009810 ;F00!F01!F02!F03!F04!F05!F06!F07!F08!F09!F10!F11!F12!F13!F14!
009820 ; 17! 57! 47! 27! 07! 37! 87! A7! C7! 67! 97! B7! 77! 70! 80!
009830 ; 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80!
009840 ;F1 !F2 !F3 !F4 !F5 !F6 !F7 !F8 !F9 !F10!F11!F12!F13!F14!F15!
009850 ; A0! A1! A2! A3! A4! A5! A6! A7! A8! A9! AA! AB! AC! AD! AE!
009860 ; C0! C1! C2! C3! C4! C5! C6! C7! C8! C9! CA! CB! CC! CD! CE!
009870 ; E0! E1! E2! E3! E4! E5! E6! E7! E8! E9! EA! EB! EC! ED! EE!
009880 ;-----+-----+-----+-----+-----+-----+-----+-----+
009890 ;E00!E01!E02!E03!E04!E05!E06!E07!E08!E09!E10!E11!E12!E13!E14!
009900 ; 16! 56! 46! 26! 06! 36! 86! 46! C6! 66! 96! B6! 76! 60! A0!
009910 ; 60! 00! 30! 29! 00! 00! 31! 32! 33! 34! 35! 44! 36! 20! 60!
009920 ;ESC!1! 2$ !3# !4$ !5X !6^ !7& !8* !9( !0) !- !+ !BSP!TAB!
009930 ; 18! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
009940 ; 18! 21! 40! 23! 24! 25! 5E! 26! 2A! 28! 29! 5F! 28! 88! 09!
009950 ; 18! 31! 32! 33! 34! 35! 36! 37! 38! 39! 3G! 20! 30! 88! 09!
009960 ;-----
009970 ; D00 !D01!D02!D03!D04!D05!D06!D07!D08!D09!D10!D11!D12! D13 !
009980 ; 15 ! 55! 45! 25! 05! 35! 85! A5! C5! 65! 95! B5! 75! 00 !
009990 ; A0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 28! 0E! A0 !
010000 ; !a0 !wW !eE !rR !Tt !yY !uU !iI !oO !pP !Aa !00 ! ! !
010010 ; ! 11! 17! 05! 12! 14! 19! 15! 09! 0F! 10! 18! 10! ! ! !
010020 ;CONTL! 51! 57! 45! 52! 54! 59! 55! 49! 4F! 50! 78! 70!CONTL!
010030 ; ! 71! 77! 65! 72! 74! 79! 75! 69! 6F! 70! 58! 50! ! !
010040 ;-----
010050 ;!C00 !C01!C02!C03!C04!C05!C06!C07!C08!C09!C10!C11!C12!C13!
010060 ;! 11 ! 51! 41! 21! 01! 31! 81! A1! C1! 61! 91! B1! 71! 72!
010070 ;! C0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 37!95 ! 38! 20!
010080 ;! !aA !sS !dD !fF !gG !hH !jJ !kK !lL !: !" !' !B ! ! !
010090 ;! ! 01! 13! 04! 06! 07! 08! 0A! 0B! 0C! ! ! 00! 1E! 98!
010100 ;!LOCK! 41! 53! 44! 46! 47! 48! 4A! 4B! 4C! 3A! 22! 7E! 88!
010110 ;! ! 61! 73! 64! 66! 67! 68! 6A! 6B! 6C! 38! 27! 60! 88!
010120 ;-----FT!
010130 ;!899!800!801!802!803!804!805!806!807!808!809!810! 811 ! !
010140 ;! 40! 12! 52! 42! 22! 02! 32! 82! A2! C2! 62! 92! B2 ! ! !
010150 ;! 90! 00! 00! 00! 00! 00! 00! 00! 00! 00! 24! 25! 0A! 90 ! !
010160 ;! !06 !zZ !xX !cC !vV !bB !nN !mM !,C !, > !/? ! ! !
010170 ;! ^ ! 1C! 1A! 18! 03! 16! 02! 0E! 0D! ! ! ! ! ! ^ !
010180 ;!SHF! 7C! 5A! 58! 43! 56! 42! 4E! 4D! 3C! 3E! 3F!SHIFT! !
010190 ;! ! 5C! 7A! 78! 63! 76! 62! 6E! 6D! 2C! 2E! 2F! ! !
010200 ;-----
010210 ; ! ! ! A05 ! !
010220 ; ! ! ! 30 ! !
010230 ; ! ! ! 60 ! !
010240 ; ! ! ! ! !
010250 ; ! ! ! 20 ! !
010260 ; ! ! ! 20 ! !
010270 ; ! ! ! 20 ! !
010280 ; -----

```

ERR-C	ADDR	OBJ-CD	P6-LIN	LABEL	CMD	OPERAND	COMMENT
	010300						UK-ENGLISH
	010310						EXPLANATION: 1ST LINE KEYLOCATION 2ND MATRIXPOSITION
	010320						3RD LINE MARKBITS 4TH SYMBOLS
	010330						5TH LINE CTL-CODE, 6TH SHIFT-CODE, 7TH UNSHIFT
	010340						-----
	010350						F00!F01!F02!F03!F04!F05!F06!F07!F08!F09!F10!F11!F12!F13!F14!
	010360						17! 57! 47! 27! 07! 37! 87! A7! C7! 67! 97! 87! 77! 70! 80!
	010370						80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80!
	010380						F1 !F2 !F3 !F4 !F5 !F6 !F7 !F8 !F9 !F10!F11!F12!F13!F14!F15!
	010390						A0! A1! A2! A3! A4! A5! A6! A7! A8! A9! AA! AB! AC! AD! AE!
	010400						C0! C1! C2! C3! C4! C5! C6! C7! C8! C9! CA! CB! CC! CD! CE!
	010410						E0! E1! E2! E3! E4! E5! E6! E7! E8! E9! EA! EB! EC! ED! EE!
	010420						-----
	010430						E00!E01!E02!E03!E04!E05!E06!E07!E08!E09!E10!E11!E12!E13!E14!
	010440						16! 56! 46! 26! 06! 36! 86! 46! 66! 96! 86! 76! 60! AD!
	010450						60! 00! 30! 29! 00! 00! 31! 21! 33! 34!S6 ! 02! 38! 20! 60!
	010460						ESC!1! 12" 13# 14\$ 15% 16& 17' 18(19) !0_ !- = !^B !8\$!TAB!
	010470						18! ! ! ! ! ! ! ! ! ! 1F! ! 1E! 98! 09!
	010480						18! 21! 22! 23! 24! 25! 26! 27! 28! 29! 5F! 30! 7E! 88! 09!
	010490						18! 31! 32! 33! 34! 35! 36! 37! 38! 39! 30! 20! 5E! 88! 09!
	010500						-----
	010510						000 !001!002!003!004!005!006!007!008!009!010!011!012! 013 !
	010520						15 ! 55! 45! 25! 05! 35! 85! A5! C5! 65! 95! 85! 75! 00 !
	010530						A0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 2B! A0 !
	010540						!q !w !e !r !t !y !u !i !o !p !s !a ! !
	010550						! 11! 17! 05! 12! 14! 19! 15! 09! 0F! 10! 00! 18! !
	010560						CONTL! 51! 57! 45! 52! 54! 59! 55! 49! 4F! 50! 60! 7B!CONTL!
	010570						! 71! 77! 65! 72! 74! 79! 75! 69! 6F! 70! 40! 5B! !
	010580						-----
	010590						C00 !C01!C02!C03!C04!C05!C06!C07!C08!C09!C10!C11!C12!C13!
	010600						! 11 ! 51! 41! 21! 01! 31! 81! A1! C1! 61! 91! 81! 71! 72!
	010610						C0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 37! 06! 0E! 20!
	010620						!a !eS !d !fF !g6 !hM !jJ !kK !lL !t+ !* !v0 ! !
	010630						! 01! 13! 04! 06! 07! 08! 0A! 0B! 0C! ! ! 10! 9B!
	010640						LOCK! 41! 53! 44! 46! 47! 48! 4A! 4B! 4C! 2B! 2A! 70! 8B!
	010650						! 61! 73! 64! 66! 67! 68! 6A! 6B! 6C! 3B! 3A! 50! 8B!
	010660						----- FT!
	010670						B99!B00!B01!B02!B03!B04!B05!B06!B07!B08!B09!B10! B11 !
	010680						40! 12! 52! 42! 22! 02! 32! 82! A2! C2! 62! 92! 82 ! !
	010690						90! 00! 00! 00! 00! 00! 00! 00! 00! 24! 25! 0A! 90 ! !
	010700						! 06 !zZ !xX !cC !vV !bB !nN !mM !, < !. > !/? ! ! !
	010710						! ^ ! 1C! 1A! 18! 03! 16! 02! 0E! 0D! ! ! ! ^ ! !
	010720						SHF! 7C! 5A! 58! 43! 56! 42! 4E! 4D! 3C! 3E! 3F!SHIFT!
	010730						! 7C! 7A! 78! 63! 76! 62! 6E! 6D! 2C! 2E! 2F! ! !
	010740						-----
	010750						! ! A05 !
	010760						! ! 30 !
	010770						! ! 60 !
	010780						! ! !
	010790						! ! 20 !
	010800						! ! 20 !
	010810						! ! 20 !
	010820						-----

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
			010840			DANISH	
			010850			EXPLANATION: 1ST LINE KEYLOCATION 2ND MATRIXPOSITION	
			010860			3RD LINE MARKBITS 4TH SYMBOLS	
			010870			5TH LINE CTL-CODE, 6TH SHIFT-CODE, 7TH UNSHIFT	
			010880			-----	
			010890			F00!F01!F02!F03!F04!F05!F06!F07!F08!F09!F10!F11!F12!F13!F14!	
			010900			!7! 57! 47! 27! 07! 37! 87! A7! C7! 67! 97! B7! 77! 70! 80!	
			010910			!80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80!	
			010920			!F1 !F2 !F3 !F4 !F5 !F6 !F7 !F8 !F9 !F10!F11!F12!F13!F14!F15!	
			010930			!A0! A1! A2! A3! A4! A5! A6! A7! A8! A9! AA! AB! AC! AD! AE!	
			010940			!C0! C1! C2! C3! C4! C5! C6! C7! C8! C9! CA! CB! CC! CD! CE!	
			010950			!E0! E1! E2! E3! E4! E5! E6! E7! E8! E9! EA! EB! EC! ED! EE!	
			010960			-----	
			010970			E00!E01!E02!E03!E04!E05!E06!E07!E08!E09!E10!E11!E12!E13!E14!	
			010980			!16! 56! 46! 26! 06! 36! 86! 46! C6! 66! 96! B6! 76! 60! A0!	
			010990			!60! 00! 30! 29! 00! 00! 31! 21! 33! 34! 02! 0A! 0D! 20! 60!	
			011000			!ESC!1! 12! 13 !4\$!5X !66 !7/ !8(!9) !0= !+? !'! !BSP!TAB!	
			011010			!18! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011020			!18! 21! 22! 23! 24! 25! 26! 2F! 28! 29! 30! 3F! 60! 88! 09!	
			011030			!18! 31! 32! 33! 34! 35! 36! 37! 38! 39! 30! 28! 27! 88! 09!	
			011040			-----	
			011050			!000 !001!002!003!004!005!006!007!008!009!010!011!012! 013 !	
			011060			! 15 ! 55! 45! 25! 05! 35! 85! A5! C5! 65! 95! B5! 75! 60! 00 !	
			011070			!A0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00 !	
			011080			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011090			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011100			!CONTL! 51! 57! 45! 52! 54! 59! 55! 49! 4F! 50! 5D! 7E!CONTL!	
			011110			! ! 71! 77! 65! 72! 74! 79! 75! 69! 6F! 70! 7D! 5E! !	
			011120			-----	
			011130			!C00 !C01!C02!C03!C04!C05!C06!C07!C08!C09!C10!C11!C12!C13!	
			011140			! ! 11 ! 51! 41! 21! 01! 31! 81! A1! C1! 61! 91! B1! 71! 72!	
			011150			!C0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00 !	
			011160			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011170			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011180			!LOCK! 41! 53! 44! 46! 47! 48! 4A! 4B! 4C! 5B! 5C! 2A! 88!	
			011190			! ! ! 61! 73! 64! 66! 67! 68! 6A! 6B! 6C! 7B! 7C! 40! 86!	
			011200			-----	
			011210			!899!800!801!802!803!804!805!806!807!808!809!810! 811 !	
			011220			! 40! 12! 52! 42! 22! 02! 32! 82! A2! C2! 62! 92! B2 ! !	
			011230			! 90! 23! 00! 00! 00! 00! 00! 00! 00! 00! 24! 25! 44! 90 ! !	
			011240			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011250			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011260			!SHF! 3E! 5A! 58! 43! 56! 42! 4E! 4D! 3B! 3A! 5F!SHIFT! !	
			011270			! ! ! 2C! 7A! 78! 63! 76! 62! 6E! 6D! 2C! 2E! 20! !	
			011280			-----	
			011290			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011300			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011310			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011320			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011330			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011340			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011350			! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011360			-----	

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CND	OPERAND	COMMENT
			011380	!		GERMAN,AUSTRIAN	
			011390	!		EXPLANATION: 1ST LINE KEYLOCATION 2ND MATRIXPOSITION	
			011400	!		3RD LINE MARKBITS 4TH SYMBOLS	
			011410	!		5TH LINE CTL-CODE, 6TH SHIFT-CODE, 7TH UNSHIFT	
			011420	!		-----	
			011430	!		F00!F01!F02!F03!F04!F05!F06!F07!F08!F09!F10!F11!F12!F13!F14!	
			011440	!		17! 57! 47! 27! 07! 37! 87! A7! C7! 67! 97! B7! 77! 70! B0!	
			011450	!		80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80!	
			011460	!		F1 !F2 !F3 !F4 !F5 !F6 !F7 !F8 !F9 !F10!F11!F12!F13!F14!F15!	
			011470	!		A0! A1! A2! A3! A4! A5! A6! A7! A8! A9! AA! AB! AC! AD! AE!	
			011480	!		C0! C1! C2! C3! C4! C5! C6! C7! C8! C9! CA! CB! CC! CD! CE!	
			011490	!		E0! E1! E2! E3! E4! E5! E6! E7! E8! E9! EA! EB! EC! ED! EE!	
			011500	!		-----	
			011510	!		E00!E01!E02!E03!E04!E05!E06!E07!E08!E09!E10!E11!E12!E13!E14!	
			011520	!		16! 56! 46! 26! 06! 36! 86! 46! C6! 66! 96! B6! 76! 60! A0!	
			011530	!		60! 00! 30! 29! 00! 00! 31! 21! 33! 34! 02! 0C! 00! 20! 60!	
			011540	!		ESC!1! !2" !3# !4\$!5% !6& !7/ !8(!9) !0= !1\$?!'! !B\$P!TAB!	
			011550	!		18! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !	
			011560	!		18! 21! 22! 40! 24! 25! 26! 2F! 28! 29! 30! 3F! 60! 88! 09!	
			011570	!		18! 31! 32! 33! 34! 35! 36! 37! 38! 39! 30! 7E! 27! 88! 09!	
			011580	!		-----	
			011590	!		000 !001!002!003!004!005!006!007!008!009!010!011!012! 013 !	
			011600	!		15 ! 55! 45! 25! 05! 35! 85! A5! C5! 65! 95! B5! 75! 00 !	
			011610	!		A0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 0E! 06! A0 !	
			011620	!		!a0 !vW !eE !rR !tT !zZ !uU !iI !oO !pP !00 !** ! !	
			011630	!		! !1! !7! 05! 12! 14! 1A! 15! 0A! 0F! 10! 10! ! !	
			011640	!		CONTL! 51! 57! 45! 52! 54! 5A! 55! 4A! 4F! 50! 5D! 2A!CONTL!	
			011650	!		! 71! 77! 65! 72! 74! 7A! 75! 6A! 6F! 70! 70! 2B! !	
			011660	!		-----	
			011670	!		C00 !C01!C02!C03!C04!C05!C06!C07!C08!C09!C10!C11!C12!C13!	
			011680	!		! 11 ! 51! 41! 21! 01! 31! 81! A1! C1! 61! 91! B1! 71! 72!	
			011690	!		C0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 2B!S41! 20!	
			011700	!		! !aA !sS !dD !FF !gG !hH !jJ !kK !L !00 !A0 !*! ! !	
			011710	!		! ! 01! 13! 04! 06! 07! 08! 0A! 0B! 0C! 10! 1B! 1E! 98!	
			011720	!		!LOCK! 41! 53! 44! 46! 47! 48! 4A! 4B! 4C! 50! 5B! 5E! 88!	
			011730	!		! ! 61! 73! 64! 66! 67! 68! 6A! 6B! 6C! 70! 7B! 23! 88!	
			011740	!		-----	FT!
			011750	!		!B99!B00!B01!B02!B03!B04!B05!B06!B07!B08!B09!B10! B11 ! !	
			011760	!		! 40! 12! 52! 42! 22! 02! 32! 82! A2! C2! 62! 92! B2 ! !	
			011770	!		! 90! 23! 00! 00! 00! 00! 00! 00! 00! 24! 25! 44! 90 ! !	
			011780	!		! !<> !yY !xX !zZ !vV !bB !nN !mM !,; !. : !- ! !	
			011790	!		! A ! ! 19! 1B! 03! 16! 02! 0E! 00! ! ! ! 1F! A ! !	
			011800	!		!SHF! 3E! 59! 58! 43! 56! 42! 4E! 4D! 3B! 3A! 5F!SHIFT! !	
			011810	!		! ! 3C! 79! 78! 63! 76! 62! 6E! 6D! 2C! 2E! 20! ! !	
			011820	!		-----	
			011830	!		! ! ! ! 805 ! ! !	
			011840	!		! ! ! ! 30 ! ! !	
			011850	!		! ! ! ! 60 ! ! !	
			011860	!		! ! ! ! ! ! !	
			011870	!		! ! ! ! 20 ! ! !	
			011880	!		! ! ! ! 20 ! ! !	
			011890	!		! ! ! ! 20 ! ! !	
			011900	!		-----	

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	012460						;NORWEGIAN
	012470						;EXPLANATION: 1ST LINE KEYLOCATION 2ND MATRIXPOSITION
	012480						; 3RD LINE MARKBITS 4TH SYMBOLS
	012490						; 5TH LINE CTL-CODE, 6TH SHIFT-CODE, 7TH UNSHIFT
	012500						-----
	012510						;F00!F01!F02!F03!F04!F05!F06!F07!F08!F09!F10!F11!F12!F13!F14!
	012520						; 17! 57! 47! 27! 07! 37! 87! A7! C7! 67! 97! B7! 77! 70! 80!
	012530						; 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80!
	012540						;F1 !F2 !F3 !F4 !F5 !F6 !F7 !F8 !F9 !F10!F11!F12!F13!F14!F15!
	012550						; AD! A1! A2! A3! A4! A5! A6! A7! A8! A9! AA! AB! AC! AD! AE!
	012560						; CD! C1! C2! C3! C4! C5! C6! C7! C8! C9! CA! CB! CC! CD! CE!
	012570						; ED! E1! E2! E3! E4! E5! E6! E7! E8! E9! EA! EB! EC! ED! EE!
	012580						-----
	012590						;E00!E01!E02!E03!E04!E05!E06!E07!E08!E09!E10!E11!E12!E13!E14!
	012600						; 16! 56! 46! 26! 06! 36! 86! 46! C6! 66! 96! B6! 76! 60! A0!
	012610						; 60! 00! 30! 29! 00! 00! 31! 21! 33! 34! 02! 0A! 00! 20! 60!
	012620						;ESC!1! 12" !34!4\$!5X !66 !7/ !8(!9) !0= !+? !' !' !BSP!TAB!
	012630						; 18! ! ! ! ! ! ! ! ! ! ! ! ! ! ! 00! 98! 09!
	012640						; 18! 21! 22! 23! 24! 25! 26! 2F! 28! 29! 30! 3F! 60! 88! 09!
	012650						; 18! 31! 32! 33! 34! 35! 36! 37! 38! 39! 30! 28! 27! 88! 09!
	012660						-----
	012670						;000 !001!002!003!004!005!006!007!008!009!010!011!012! 013 !
	012680						; 15 ! 55! 45! 25! 05! 35! 85! A5! C5! 65! 95! B5! 75! 00 !
	012690						; AD ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 0E! 18! A0 !
	012700						; !q@ !vw !eE !rR !tT !yY !uU !iI !oO !pP !Aa !Bb ! !
	012710						; ! ! 11! 17! 05! 12! 14! 19! 15! 09! 0F! 10! 10! 1E! ! !
	012720						;CONTL! 51! 57! 45! 52! 54! 59! 55! 49! 4F! 50! 5D! 7E!CONTL!
	012730						; ! 71! 77! 65! 72! 74! 79! 75! 69! 6F! 70! 7D! 5E! !
	012740						-----
	012750						;!C00 !C01!C02!C03!C04!C05!C06!C07!C08!C09!C10!C11!C12!C13!
	012760						;! 11 ! 51! 41! 21! 01! 31! 81! A1! C1! 61! 91! B1! 71! 72!
	012770						;! C0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 28! 19! 20!
	012780						;! ! !aA !sS !dD !ff !gG !hH !jJ !kK !L !äÄ !öÖ !*# ! !
	012790						;! ! 01! 13! 04! 06! 07! 08! 0A! 0B! 0C! 18! 1C! ! 98!
	012800						;!LOCK! 41! 53! 44! 46! 47! 48! 4A! 4B! 4C! 5B! 5C! 2A! 68!
	012810						;! ! 61! 73! 64! 66! 67! 68! 6A! 6B! 6C! 7B! 7C! 40! 88!
	012820						-----
	012830						;!B99!B00!B01!B02!B03!B04!B05!B06!B07!B08!B09!B10! B11 ! !
	012840						;! 40! 12! 52! 42! 22! 02! 32! 82! A2! C2! 62! 92! B2 ! !
	012850						;! 90! 23! 00! 00! 00! 00! 00! 00! 00! 24! 25! 44! 90 ! !
	012860						;! !< !zZ !xX !cC !vV !bB !nN !mM !,!;! !_ ! ! !
	012870						;! ! A ! ! 1A! 18! 03! 16! 02! 0E! 00! ! ! !F! A ! !
	012880						;!SHF! 3E! 5A! 58! 43! 56! 42! 4E! 4D! 3B! 3A! 5F!SHIFT! !
	012890						;! ! 2C! 7A! 78! 63! 76! 62! 6E! 6D! 2C! 2E! 20! ! !
	012900						-----
	012910					A05	!
	012920					30	!
	012930					60	!
	012940						!
	012950					20	!
	012960					20	!
	012970					20	!
	012980					-----	!

```

013000 ;SPANISH
013010 ;EXPLANATION: 1ST LINE KEYLOCATION 2ND MATRIXPOSITION
013020 ;      3RD LINE MARKBITS      4TH SYMBOLS
013030 ;      5TH LINE CTL-CODE, 6TH SHIFT-CODE, 7TH UNSHIFT
013040 ;-----
013050 ;F00!F01!F02!F03!F04!F05!F06!F07!F08!F09!F10!F11!F12!F13!F14!
013060 ; 17! 57! 47! 27! 07! 37! 87! 47! C7! 67! 97! B7! 77! 70! 80!
013070 ; 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80!
013080 ;F1 !F2 !F3 !F4 !F5 !F6 !F7 !F8 !F9 !F10!F11!F12!F13!F14!F15!
013090 ; A0! A1! A2! A3! A4! A5! A6! A7! A8! A9! AA! AB! AC! AD! AE!
013100 ; C0! C1! C2! C3! C4! C5! C6! C7! C8! C9! CA! CB! CC! CD! CE!
013110 ; E0! E1! E2! E3! E4! E5! E6! E7! E8! E9! EA! EB! EC! ED! EE!
013120 ;-----
013130 ;E00!E01!E02!E03!E04!E05!E06!E07!E08!E09!E10!E11!E12!E13!E14!
013140 ; 16! 56! 46! 26! 06! 36! 86! 46! C6! 66! 96! B6! 76! 60! A0!
013150 ; 60! 00! 30! 29! 00! 00! 31! 21! 33! 34! 02! 1A! 2B! 20! 60!
013160 ;ESC!1! 12" !3# !4$ !5% !6& !7/ !8( !9) !0= !1$ !i? !BSP!TAB!
013170 ; 1B! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
013180 ; 1B! 21! 22! 23! 24! 25! 26! 2F! 28! 29! 30! 3F! 50! 8B! 09!
013190 ; 1B! 31! 32! 33! 34! 35! 36! 37! 38! 39! 30! 40! 5B! 8B! 09!
013200 ;-----
013210 ; 000 !001!002!003!004!005!006!007!008!009!010!011!012! 013 !
013220 ; 15 ! 55! 45! 25! 05! 35! 85! A5! C5! 65! 95! B5! 75! 00 !
013230 ; A0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 38! 06! A0 !
013240 ; !a@ !vw !eE !rR !tT !yY !uU !iI !oO !pP !AB !++ ! !
013250 ; ! 11! 17! 05! 12! 14! 19! 15! 09! 0F! 10! 1E! ! ! !
013260 ;CONTL! 51! 57! 45! 52! 54! 59! 55! 49! 4F! 50! 7E! 2A!CONTL!
013270 ; ! 71! 77! 65! 72! 74! 79! 75! 69! 6F! 70! 5E! 2B! !
013280 ;-----
013290 ;!C00 !C01!C02!C03!C04!C05!C06!C07!C08!C09!C10!C11!C12!C13!
013300 ;! 11 ! 51! 41! 21! 01! 31! 81! A1! C1! 61! 91! B1! 71! 72!
013310 ;! C0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 0E! 20!
013320 ;! !aA !sS !dD !fF !gG !hH !jJ !kK !lL !nNB!'' !äü ! !
013330 ;! ! 01! 13! 04! 06! 07! 08! 0A! 0B! 0C! 1C! 00! 10! 98!
013340 ;!LOCK! 41! 53! 44! 46! 47! 48! 4A! 4B! 4C! 5C! 60! 70! 88!
013350 ;! ! 61! 73! 64! 66! 67! 68! 6A! 6B! 6C! 7C! 27! 7B! 8B!
013360 ;-----FT!
013370 ;!B99!B00!B01!B02!B03!B04!B05!B06!B07!B08!B09!B10! B11 ! !
013380 ;! 40! 12! 52! 42! 22! 02! 32! 82! A2! C2! 62! 92! B2 ! !
013390 ;! 90! 23! 00! 00! 00! 00! 00! 00! 00! 24! 25! 44! 90 ! !
013400 ;! (< !zZ !xX !cC !vV !bB !nN !mM !,; ! !- ! ! !
013410 ;! A ! ! 1A! 1B! 03! 16! 02! 0E! 00! ! ! ! 1F! A ! !
013420 ;!SHF! 3E! 5A! 5B! 43! 56! 42! 4E! 4D! 3B! 3A! 5F!SHIFT! !
013430 ;! ! 2C! 7A! 7B! 63! 76! 62! 6E! 6D! 2C! 2E! 20! ! !
013440 ;!-----
013450 ; ! ! ! A05 ! !
013460 ; ! ! ! 30 ! !
013470 ; ! ! ! 60 ! !
013480 ; ! ! ! ! !
013490 ; ! ! ! 20 ! !
013500 ; ! ! ! 20 ! !
013510 ; ! ! ! 20 ! !
013520 ; !-----

```

ERR-C ADDR OBJ-CO PG-LIN LABEL CMD OPERAND COMMENT

```

013540 ;ITALIAN
013550 ;EXPLANATION: 1ST LINE KEYLOCATION 2ND MATRIXPOSITION
013560 ;      3RD LINE MARKBITS      4TH SYMBOLS
013570 ;      5TH LINE CTL-CODE, 6TH SHIFT-CODE, 7TH UNSHIFT
013580 ;-----
013590 ;F00!F01!F02!F03!F04!F05!F06!F07!F08!F09!F10!F11!F12!F13!F14!
013600 ; 17! 57! 47! 27! 07! 37! 87! A7! C7! 67! 97! 87! 77! 70! 80!
013610 ; 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80!
013620 ;F1 !F2 !F3 !F4 !F5 !F6 !F7 !F8 !F9 !F10!F11!F12!F13!F14!F15!
013630 ; A0! A1! A2! A3! A4! A5! A6! A7! A8! A9! AA! AB! AC! AD! AE!
013640 ; C0! C1! C2! C3! C4! C5! C6! C7! C8! C9! CA! CB! CC! CD! CE!
013650 ; E0! E1! E2! E3! E4! E5! E6! E7! E8! E9! EA! EB! EC! ED! EE!
013660 ;-----
013670 ;E00!E01!E02!E03!E04!E05!E06!E07!E08!E09!E10!E11!E12!E13!E14!
013680 ; 16! 56! 46! 26! 06! 36! 86! 46! 66! 66! 96! 86! 76! 60! AD!
013690 ; 60! 00! 30! 29! 00! 00! 31! 21! 33! 34! 02! 0F!S9 ! 20! 60!
013700 ;ESC!1! 12! 13 !4$ !5% !6& !7! !8( !9) !0= !? !!'"!BSP!TAB!
013710 ; 18! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! 1E! 98! 09!
013720 ; 18! 21! 22! 23! 24! 25! 26! 2F! 28! 29! 30! 3F! 5E! 88! 09!
013730 ; 18! 31! 32! 33! 34! 35! 36! 37! 38! 39! 30! 27! 7E! 88! 09!
013740 ;-----
013750 ;000 !001!002!003!004!005!006!007!008!009!010!011!012! 013 !
013760 ; 15 ! 55! 45! 25! 05! 35! 85! A5! C5! 65! 95! 85! 75! 00 !
013770 ; A0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 0E! 06! A0 !
013780 ; !a@ !zZ !eE !rR !tT !yY !uU !iI !oO !pP !ee!++ ! !
013790 ; ! 11! 1A! 05! 12! 14! 19! 15! 09! 0F! 10! 10! ! !
013800 ;CONTL! 51! 5A! 45! 52! 54! 59! 55! 49! 4F! 50! 7D! 2A!CONTL!
013810 ; ! 71! 7A! 65! 72! 74! 79! 75! 69! 6F! 70! 5D! 2B! ! !
013820 ;-----
013830 ;!C00 !C01!C02!C03!C04!C05!C06!C07!C08!C09!C10!C11!C12!C13!
013840 ;! 11 ! 51! 41! 21! 01! 31! 81! A1! C1! 61! 91! 81! 71! 72!
013850 ;! C0 ! 00! 00! 00! 00! 00! 00! 00! 00!S2 !S4 !S7 ! 20!
013860 ;! !aA !sS !dD !fF !gG !hH !jJ !kK !L !a'c'u'.!o'&! !
013870 ;! ! 01! 13! 04! 06! 07! 08! 0A! 0B! 0C! 1C! 1B! 00! 9B!
013880 ;!LOCK! 41! 53! 44! 46! 47! 48! 4A! 4B! 4C! 5C! 5B! 40! 8B!
013890 ;! ! 61! 73! 64! 66! 67! 68! 6A! 6B! 6C! 7B! 60! 7C! 8B!
013900 ;-----
013910 ;!B99!B00!B01!B02!B03!B04!B05!B06!B07!B08!B09!B10! B11 ! !
013920 ;! 40! 12! 52! 42! 22! 02! 32! 82! A2! C2! 62! 92! 82 ! !
013930 ;! 90! 23! 00! 00! 00! 00! 00! 00! 00! NN! NN! 2B! 90 ! !
013940 ;! !< !wW !xX !cC !vV !bB !nN !mM !,;! !- ! ! !
013950 ;! B ! ! 17! 18! 03! 16! 02! 0E! 00! ! ! !F! A ! !
013960 ;!SHF! 3E! 57! 58! 43! 56! 42! 4E! 4D! 3B! 3A! 5F!SHIFT! !
013970 ;! ! 3C! 77! 78! 63! 76! 62! 6E! 6D! 2C! 2E! 2D! ! !
013980 ;-----
013990 ; ! ! ! A05 ! !
014000 ; ! ! ! 30 ! !
014010 ; ! ! ! 60 ! !
014020 ; ! ! ! ! !
014030 ; ! ! ! 20 ! !
014040 ; ! ! ! 20 ! !
014050 ; ! ! ! 20 ! !
014060 ; ! ! ! ! !
    
```

ACCPK	LABL	0155H	005060						
ASCITB	LABL	0360H	008650	005370					
ATEMP	EQU	0007H	001060						
BDCLK	EQU	0004H	001030						
BITCTR	EQU	0003H	001020						
BITIME	LABL	0089H	002610	002490					
BUS	EQU	0000H	000000						
C	EQU	0070H	000000						
CAFLG	EQU	007EH	001110						
CHARBF	EQU	0005H	001040						
CHKSUM	LABL	03FEH	008840						
CKSU1	LABL	01F9H	006440	003400					
CKSU2	LABL	0200H	006520						
CKSU3	LABL	0007H	003410						
CLEAR	LABL	00B9H	003100	003230					
CNT	EQU	0000H	000000						
CNTDWN	LABL	008FH	002670	002420	002630				
COND	LABL	0208H	006610	005630					
DBB	EQU	001AH	000000						
DECP0I	EQU	0024H	001220	003720	006620				
DMJ	EQU	00E0H	000000						
DOUBL0	EQU	000BH	001150	005560					
ERRCNT	EQU	0002H	001010						
EXCEPT	LABL	0189H	005900	005260	005940	005950			
EXINT	LABL	006BH	002340	001690	001790	002040	002090	002570	
FO	EQU	005EH	000000						
F1	EQU	007EH	000000	001110					
FIFO	EQU	0010H	001200	002760	002770	003280	003290		
FIFOEN	EQU	0017H	001210						
FLAGB	EQU	0021H	001170	005090	006120	006380			
FLAGS	EQU	00FDH	000000						
FUKY	LABL	0183H	005840	005240					
I	EQU	0000H	000000						
INTRPT	LABL	002AH	001780	001350					
INTRUP	LABL	0003H	001320						
KEYS	EQU	0068H	008630	004430					
KYDOWN	LABL	0138H	004840	004740					
KYDWN	EQU	0007H	000950						
LASTCD	EQU	0022H	001180	005180					
LASTKY	EQU	0023H	001190	005140	005770				
LDKBQ	LABL	0092H	002710	005530	005580				
LNGOFS	LABL	026EH	007560	007400					
LOCKY	LABL	0132H	004790	004960	005050	006420			
LONGDL	EQU	0060H	001120	005070					
MB0	EQU	0020H	000000						
MB1	EQU	0030H	000000						
MRKTBL	LABL	0300H	008490	008630					
MSKT8	LABL	02CCH	007860	006080	006150				
MTKBQ	LABL	003DH	001980	001710	001830				
N0	LABL	0179H	005330						
N1	LABL	018CH	005480						
N3	LABL	0191H	005520	003000	006760				
N4	LABL	019CH	005610	005500					
N5	LABL	01A0H	005650	005620					
N5PD	LABL	00ACH	002960	005650					

NATVLN	LABL	0256H	007350	005170																
NEWS	LABL	0104H	004410	004500																
NORML5	LABL	0174H	005280	005880	006130															
P0	EQU	0000H	000000																	
P1	EQU	0001H	000000																	
P2	EQU	0002H	000000																	
P4	EQU	0004H	000000																	
P5	EQU	0005H	000000																	
P6	EQU	0006H	000000																	
P7	EQU	0007H	000000																	
PAUSE	LABL	0080H	002650	002510																
PSW	EQU	00CFH	000000																	
QHPTR	EQU	0006H	000940																	
QTPTR	EQU	0005H	000930	002050																
R0	EQU	0000H	000000																	
R1	EQU	0001H	000000	000860																
R2	EQU	0002H	000000	001010																
R3	EQU	0003H	000000	000910	001020															
R4	EQU	0004H	000000	000920	001030															
R5	EQU	0005H	000000	000930	001040															
R6	EQU	0006H	000000	000940	001050															
R7	EQU	0007H	000000	000950	001060															
R80	EQU	0000H	000000																	
RB1	EQU	0010H	000000																	
REFCK	LABL	01A2H	005690	004910																
REPDLY	EQU	0004H	000920																	
REPEAT	LABL	0157H	005080	005820																
REPRAT	EQU	0008H	001130	005810																
RESHFG	LABL	01F2H	006370	004950																
RESTRT	LABL	0000H	001280																	
ROMCK	LABL	00CCH	003310																	
SCAN	LABL	0108H	004460	004760	004770	004830	004880	005030	005130	005440										
			005570	005590	005710	005740	005800	006350												
SEND	LABL	0060H	002380	001890	002210															
SHFTRG	EQU	0006H	001050																	
SPZEXC	LABL	01E0H	006190	005210	005920															
SPZXCD	LABL	03DAH	008800	006280																
START	LABL	0084H	003040	001300																
STPTR	EQU	0001H	000860																	
STRAPS	EQU	0020H	001160	001840	001990	006320	007370													
STS	EQU	00E8H	000000																	
STSHFG	LABL	01EEH	006320	005860	006710															
SUPLFU	LABL	0220H	006790	006720																
T	EQU	004AH	000000																	
TBLPTR	EQU	0003H	000910																	
TCNTI	EQU	0020H	000000																	
TIMER	LABL	0007H	001460																	
TMRCNT	EQU	FFF0H	001140	001670																
TRANS	LABL	021CH	006730	002990	006160	006300	006660	006680												
TXTCHG	LABL	022CH	006820	005510																
UNCH	LABL	0235H	006900	006840	006880	006960	006980	007090	007110	007460										
UNLOCK	LABL	014AH	004970	004730																
XL	LABL	026CH	007540	006190																
XLCD	LABL	02E6H	007910	006150																

KEYBOARD FIRMWARE (VERSION 2)

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
			00010 ;				*****
			00020 ;		*		* THIS DOCUMENT, FILES AND INFORMATION HEREIN *
			00030 ;		*		* ARE CONFIDENTIAL AND PROPRIETARY PROPERTY *
			00040 ;		*		* OF NCR CORPORATION AND ALL UNAUTHORIZED *
			00050 ;		*		* USE AND REPRODUCTION IS PROHIBITED *
			00060 ;				*****
			00070 ;				
			00080 ;				
			00090 ;				*****
			00100 ;		*		*
			00110 ;		*	COPYRIGHT (C) 1983	*
			00120 ;		*	BY	*
			00130 ;		*	NCR CORPORATION	*
			00140 ;		*	DAYTON, OHIO U.S.A.	*
			00150 ;		*	ALL RIGHTS RESERVED	*
			00160 ;		*	PRINTED IN U.S.A.	*
			00170 ;		*	CONFIDENTIAL, UNPUBLISHED	*
			00180 ;		*	PROPERTY OF NCR CORPORATION	*
			00190 ;		*		*
			00200 ;				*****
			00210 ;				
			00220 ;				
			00230 ;			LANGUAGEGROUP II : Swiss,France,Australia,Canada	
			00240 ;			South-Africa,Portugal,Yugoslavia	
			00250 ;			PCB must contain 8048/8748 strappings,if 8749 or 8049's	
			00260 ;			are used. No strappings are required,if 8742 or 8742's	
			00270 ;			are used,but firmware is different from 49'ers!	
			00280 ;			Decimal point/comma switching using ctrl+dec.point key	
			00290 ;			ctrl+F20 inforces a clear FIFO ,code B3 hex is trans-	
			00300 ;			mitted to kb-IF chip ,which initiates a COLD-Start-	
			00310 ;			RESET to the computer. Kb-power-up routine is n o t	
			00320 ;			repeated in this case!!!!	

EKR-C ADDR OBJ-CD PG-LIN LABEL CMD OPERAND COMMENT

```

000340 ; *****
000350 ; * *
000360 ; * PERSONAL-COMP.-KEYBOARD*
000370 ; * *
000380 ; * ASCII ENCODED PLL/SER. *
000390 ; * LANGUAGEGROUP II *
000400 ; * APRIL 83 *
000410 ; * USE 8049 OR 8749 *
000420 ; * X-TAL= 6.00 MHZ *
000430 ; * *
000440 ; *****
000450 ;
000460 ;
000470 ; REGISTER BANK 0 REGISTER BANK 1
000480 ; -----
000490 ; R0...TEMPORARY STORAGE R0...TEMPORARY STORAGE
000500 ; R1...STATUS ARRAY POINTER R1...TEMPORARY STORAGE
000510 ; R2...TEMPORARY STORAGE R2...ERRCNT(BELL MULTIPLIER)
000520 ; R3...TABLE STORAGE R3...XMIT BIT COUNTER
000530 ; R4...REPEAT DELAY CTR R4...BAUD CLOCK (BEEP CYCLES)
000540 ; R5...QUEUE TAIL POINTER R5...XMIT CHARACTER BUFFER
000550 ; R6...QUEUE HEAD POINTER R6...XMIT SHIFT REGISTER
000560 ; R7...8 KEY STATUS R7...SAVE ACC DURING INTRPTS
000570 ;
000580 ; RAM: 26...3F KEY STATUS 2 BITS/KEY 4 KEYS/BYTE
000590 ; STATUS TURNAROUND 2 BITS LEFT FOR EACH KEY
000600 ; 10...17 QUEUE OF CODES TO BE SENT (FIFO)
000610 ;
000620 ; TABLES: 300...367 MARKBIT-TABLE
000630 ;
000640 ; 368...3CF ASCII CODE TABLE
000650 ; F0 SERIAL KB TOGGLE
000660 ;
000670 ; MARKBIT ASSIGNMENTS:
000680 ; BIT 0...0
000690 ; BIT 1...3
000700 ; BIT 2... ) EXCEPTION OFFSET
000710 ; BIT 3...6
000720 ; BIT 4.../ SHIFKY(B7=1)
000730 ; BIT 5...IGNORE SHIFT(B0..4,7=0)/CTLY(B7=1)/
000740 ; IF B0..4)9 EXCEPTION CODES FOR SYMBOLCODES 21...3FH
000750 ; BIT 6...IGNORE CTRL-FLAG/LOCKKY(B7=1)/SPEZ.XLAT(B0..4)0
000760 ; BIT 7...FUNCTION KEY/NO AUTOREPEAT INDICATOR/SWF,CTRL,LOCK-
000770 ;
000780 ; 80(FUNCTION KEYS F1...F20) 90(SHIFT KEYS) A0(CONTROL KEY
000790 ; C0(ALPHA LOCK KEY) 20(SHIFT COND. IGNORED) 40(CTRL.COND IGN
000800 ; 1...15 (SYMBOL CODE EXCEPTIONS) 4X(SPECIAL EXCEPTIONS)
000810 ; B0 NO EXCEPTIONS,NORMAL CODE CALCULATION:
000820 ; UNSHIFT CODE EXORED BY SHIFT OR CONTROL FLAG,
000830 ; SHIFT FLAG INVERTED FOR LETTER CODES DURING ALPHA LOCK ON
000840 ; AND CONTROL FLAG OFF,
000850 ; CONTROL MODE IGNORED FOR CHARACTERCODES ( 5
000860 ; SHIFT- AND CONTROL-MODE COINCIDENCE DISABLED FOR ALL KEYS

```


ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
			000880	;	R00	EQUATES	
			000890	;	-----		
			000900	;			
0001			000910	STPTR	EQW	R1	;STATUS POINTER
			000920	;		BIT 0=0	KEY UP AT LAST SCAN
			000930	;		1	KEY DOWN AT LAST SCAN
			000940	;			AFTER CHECKING KEY DOWN BIT 6&7 ARE ROTATED TO
			000950	;			
0003			000960	TBLPTR	EQW	R3	;POINTER TO R/C VALUE IN PAGE 3
0004			000970	REPLY	EQW	R4	;REPEAT DELAY COUNTER FOR AUTO REPEA
0005			000980	QTPTR	EQW	R5	;QUEUE TAIL POINTER
0006			000990	QHPTR	EQW	R6	;QUEUE HEAD POINTER
0007			001000	KYDWM	EQW	R7	;8 KEY DOWN REGISTER
			001010	;			TIMES HOW LONG KEYS LOCKED OUT
			001020	;			
			001030	;	R01	EQUATES	
			001040	;	-----		
			001050	;			
0002			001060	ERRCNT	EQW	R2	;BEEP MULTIPLIER FOR LONG TONE(.5SEC
0003			001070	BITCTR	EQW	R3	;COUNTS SERIAL BITS SHIFTED OUT
0004			001080	BDCLK	EQW	R4	;BIT COUNTER FOR BAUD CLOCK
0005			001090	CHARBF	EQW	R5	;CHARACTERBUFFER FOR SERIAL OUTPUT
0006			001100	SHFTRG	EQW	R6	;SHIFT REGISTER
0007			001110	ATEMP	EQW	R7	;SAVE ACC DURING INTERRUPT
			001120	;			
			001130	;	OTHERS		
			001140	;	-----		
			001150	;			
007E			001160	CAFLG	EQW	F1	;CHARACTER AVAILABLE FLAG
0060			001170	LONGDLY	EQW	96	
0008			001180	REPRAT	EQW	8	;APPROX. 11 HZ
FFF0			001190	TMRcnt	EQW	-3	
0008			001200	DOUBLO	EQW	11	;DOUBLE ZERO DISPLACEMENT
0020			001210	STRAFS	EQW	32	;STRAPPING INFO
0021			001220	FLAGB	EQW	33	;WP-,CTL-,SHIFT FLAGB
0022			001230	LASTCD	EQW	34	;INTERIM STORAGE CELL
0023			001240	LASTKY	EQW	35	;LAST KEY POSITION
0010			001250	FIFO	EQW	16	;FIFO STARTADDRESS
0017			001260	FIFOEND	EQW	23	;LAST ADDRESS
0024			001270	DECPOI	EQW	36	;SOFTADDRESS OF DEC.POINT/COMMA

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0000		001290		ORG	0	
			001300				
			001310				; RESTART, INTERRUPT VECTORS
			001320				
	0000		001330		RESTR:		
	0000	35	001340		DIS	TCNTI	;DISABLE TIMER INTERRUPT
	0001	04BA	001350		JMP	START	;START KB SCAN
			001360				
	0003		001370		INTRUPT:		
	0003	05	001380		SEL	RB1	
	0004	AF	001390		MOV	ATEMP,A	
	0005	0429	001400		JMP	INTRPT	
			001410				
			001420				; TIMER INTERRUPT
			001430				; *****
			001440				
			001450				;THIS INTERRUPT IS EVERY 240MYSEC(XTAL=6 MHZ) OR 4MSEC
			001460				; TMRcnt=3 IT SERVES TWO FUNCTIONS:
			001470				
			001480				1) BEEP TONE GENERATION(240MYSEC)
			001490				2) EMPTY KB QUEUE (4MS)
			001500				
	0007		001510		TIMER:		
	0007	05	001520		SEL	RB1	
	0008	AF	001530		MOV	ATEMP,A	;SAVE ACC
	0009	23CE	001540		MOV	A,#-50	;4MS PAUSE
	000B	62	001550		MOV	T,A	
			001560				
			001570				; OUTPUT BAUD CLOCK TO PORT 26
			001580				
	000C	FC	001590		MOV	A,BDCLK	;ODD/EVEN COUNT TEST
	000D	F221	001600		JB7	\$4	
	000F	9617	001610		JNZ	\$5	;BEEP TONE IS SET
	0011	FA	001620		MOV	A,ERRCNT	;TEST BEEP MULTIPLIER
	0012	C617	001630		JZ	\$5	;NO, CONTINUE NORMAL
	0014	CA	001640		DEC	ERRCNT	;ELSE DECREMENT ERROR COUNTER
	0015	BC3C	001650		MOV	BDCLK,#60	;SET COUNT OF CYCLES FOR BEEP
	0017		001660		\$5:		
	0017	0A	001670		IN	A,P2	;LOAD BEEP CONDITION
	0018	D340	001680		XRL	A,#40H	;SWAP PHASE
	001A	3A	001690		OUTL	P2,A	
	001B	CC	001700		DEC	BDCLK	
	001C		001710		\$1:		
	001C	23FD	001720		MOV	A,#TMRcnt	;RELOAD TIMER REGISTER
	001E	62	001730		MOV	T,A	
	001F	0464	001740		JMP	EXINT	
	0021		001750		\$4:		
	0021	0442	001760		JMP	MTKBQ	;TEST ON BELL LATCH SET
	0023	BA23	001770		MOV	ERRCNT,#35	;SET MULTIPLIER
	0025	BC3C	001780		MOV	BDCLK,#60	
	0027	041C	001790		JMP	\$1	
			001800				

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0029		001820	INTRPT:			;IBF IS SET
	0029	862D	001830		JNI	\$+4	;POLL IS PENDING
	002B	0464	001840		JMP	EXINT	;ELSE IGNORE INTERRUPT
	002D	23FD	001850		MOV	A,\$-3	
	002F	8816	001860		MOV	RO,\$22	;SET DELAY COUNTER 300 USEC
	0031	17	001870		INC	A	
	0032	8631	001880		JNI	\$-1	;A SECOND POLL IS PENDING
	0034	E832	001890		DJMZ	RO,\$-2	
	0036	F242	001900		JB7	HTKBQ	;THERE WAS NO 2ND POLL(WR/) SIGNAL
	0038	B82D	001910		MOV	RO,\$STRAPS	;LOAD LANGUAGE CODE
	003A	F0	001920		MOV	A,\$RO	
	003B	530F	001930		ANL	A,\$15	;MASK OUT UPPER STRAP-INFO
	003D	43D0	001940		ORL	A,\$-48	;SET UP UPPER BITS +ASSUME GOOD KB S
	003F	AD	001950		MOV	CHARBF,A	
	0040	0466	001960		JMP	SEND	
			001970				
			001980				; NTKBQ- EMPTY KEYBOARD QUEUE
			001990				; -----
			002000				; LOOKS INTO QUEUE ONCE EVERY INTERRUPT.
			002010				; IF THERE IS A CODE, IT IS PUT INTO CHARBF
			002020				; THE CHARACTER AVAILABLE FLAG IS SET
			002030				; THE QUEUE POINTER IS UPDATED UNTIL EMPTY CONDITION
			002040				
	0042		002050				NTKBQ:
	0042	8820	002060		MOV	RO,\$STRAPS	;SELECT SERIAL INTERFACE
	0044	F0	002070		MOV	A,\$RO	
	0045	37	002080		CPL	A	;INVERT INDICATOR
	0046	AB	002090		MOV	R3,A	;SAVE FOR FURTHER USE
			002100		JB4	\$+4	;SKIP MASTER SLAVE IF RESPECTS
			002110		JOB7	EXINT	;EXIT IF LAST KEY NOT YET FETCHED
	0047	8805	002120		MOV	RO,\$PTPTR	;ELSE LOOK INTO QUEUE TO FIND CODES
	0049	F0	002130		MOV	A,\$RO	; FOR TRANSMIT
	004A	18	002140		INC	RO	
	004B	D0	002150		XRL	A,\$RO	; HEAD=TAIL MEANS EMPTY
	004C	C664	002160		JZ	EXINT	;FIFO EMPTY
			002170				ELSE GET A CODE FROM QUEUE
	004E	F0	002180		MOV	A,\$RO	;PUT POINTER INTO R1
	004F	A9	002190		MOV	R1,A	
	0050	F1	002200		MOV	A,\$R1	;GET THE CODE
	0051	AD	002210		MOV	CHARBF,A	;PUT INTO BUFFER
	0052	F0	002220		MOV	A,\$RO	;UPDATE HEAD POINTER
	0053	17	002230		INC	A	
	0054	5317	002240		ANL	A,\$17H	;MAKE RING COUNTER
	0056	A0	002250		MOV	\$RO,A	;AND REPLACE
	0057		002260	\$1:			
	0057	FB	002270		MOV	A,R3	;RELOAD STRAPINFO
	0058	F266	002280		JB7	SEND	
	005A	18	002290		INC	RO	;POINT TO FLAGB
	005B	F0	002300		MOV	A,\$RO	
	005C	F260	002310		JB7	\$3	;IF TRUE THEN SKIP
	005E	8C3C	002320		MOV	BDCLK,\$60	;OTHERWISE INITIATE A BEEP
	0060		002330	\$3:			
	0060	F0	002340		MOV	A,CHARBF	;LOAD CODE
	0061	D3E0	002350		XRL	A,\$0EDH	;INVERT UPPER 3 BITS
	0063	02	002360		OUTL	BUS,A	;AND OUTPUT TO BUS PORT

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0064		002370	EXINT:			;EXIT TIMER INTERRUPT
	0064	FF	002380		MOV	A,ATEMP	
	0065	93	002390		RETR		
			002400				;
	0066		002410	SEND:			
	0066	8808	002420		MOV	BITCTR,#8	;SET COUNTER FOR 8 BITS
	0068		002430	\$1:			
	0068	9A7F	002440		ANL	P2,#7FH	;SET SYNCH BIT
	006A	1489	002450		CALL	CNTDWN+2	;TIME IS 25 USEC
	006C	FD	002460		MOV	A,CHARBF	
	006D	77	002470		RR	A	;SEND LSB FIRST
	006E	AD	002480		MOV	CHARBF,A	
	006F	F27F	002490		JB7	\$4	;HOLD ON BIT
	0071	8A80	002500		ORL	P2,#80H	;ELSE RESET IT
	0073		002510	\$2:			
	0073	1481	002520		CALL	BITIME	;TIME IS NOW 80 USEC
	0075	8A80	002530		ORL	P2,#80H	;RESET NOW UNCONDITIONAL
	0077	1485	002540		CALL	PAUSE	;WAIT NOW 100 USEC
	0079	CB	002550		DEC	BITCTR	
	007A	FB	002560		MOV	A,BITCTR	
	007B	9668	002570		JNZ	\$1	;REPEAT 8 TIMES
	007D		002580	\$3:			
	007D	0464	002590		JMP	EXINT	
			002600				;
	007F	0473	002610	\$4:	JMP	\$2	;ROUTINE LENGTH EGALIZATION
			002620				;
	0081		002630	BITIME:			
	0081	8808	002640		MOV	RO,#11	
	0083	0487	002650		JMP	CNTDWN	
			002660				;
	0085		002670	PAUSE:			
	0085	880E	002680		MOV	RO,#14	
	0087		002690	CNTDWN:			
	0087	E887	002700		DJNZ	RO,\$	
	0089	83	002710		RET		
			002720				;
	008A		002730	LOKBO:			;LOAD A RECOGNIZED CODE INTO FIFO
	008A	2383	002740		MOV	A,#0B3H	;SELECT RESET CODE
	008C	DA	002750		XRL	A,R2	
	008D	9693	002760		JNZ	\$1	
	008F	8D10	002770		MOV	QTPTR,#FIFO	
	0091	BE10	002780		MOV	QHPTR,#FIFO	;SET FIFO EMPTY
	0093		002790	\$1:			
			002800				;
	0093	FD	002810		MOV	A,QTPTR	;TEST FOR PLACE
	0094	17	002820		INC	A	
	0095	5317	002830		ANL	A,#17H	;MASK MODULO 8
	0097	DE	002840		XRL	A,QHPTR	;COMPARE WITH HEAD-POINTER
	0098	C6A3	002850		JZ	\$2	;FULL,IGNORE ENTRY AND INITIATE ERR
			002860				;
	009A	FD	002870		MOV	A,QTPTR	;PUT POINTER INTO RO
	009B	A8	002880		MOV	RO,A	
	009C	FA	002890		MOV	A,R2	;GET CODE FOR FIFO
	009D	AD	002900		MOV	SRO,A	
	009E	FD	002910		MOV	A,QTPTR	;UPDATE POINTER

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	009F	17	002920		INC	A	
	00A0	5317	002930		ANL	A,#17H	;MASK INTO FIFO AREA
	00A2	AD	002940		MOV	QTPTR,A	
	00A3		002950	\$2:			
	00A3	83	002960		RET		
			002970				
			002980	NSPD:			
	00A4	030C	002990		ADD	A,#12	;SELECT F5..FF
	00A6	E6AC	003000		JMC	\$1	
	00A8	03AE	003010		ADD	A,#SUPLFU	
	00AA	A3	003020		MOV	A,#5A	
	00AB	AA	003030		MOV	R2,A	
	00AC	249E	003040	\$1:	JMP	N3	
			003050				
			003060	SUPLFU:			
00AE			003070		DB	'456789**+123/'	
00AE	343536373839		003070				
			2A2B3132332F				
			003080				;*****INITIALIZE SCAN*****
			003090				
00BA			003100	START:			
00BA	9ABF		003110		ANL	P2,#0BFH	;PREPARE MATRIX READ STRAP INFO
			003120				
			003130				
00BC	95		003140		CPL	FD	;PRESET GOOD KB
00BD	B87F		003150		MOV	RD,#127	;SET RAM NSLOCATION
00BF			003160	CLEAR:			
00BF	27		003170		CLR	A	
00C0	37		003180		CPL	A	
00C1	AD		003190		MOV	SRO,A	
00C2	DD		003200		XRL	A,SRO	;TEST FF
00C3	C6C6		003210		JZ	\$1	
00C5	85		003220		CLR	FD	;SET RAM ERROR FLAG
00C6			003230	\$1:			
00C6	AD		003240		MOV	SRO,A	;CLEAR RAM
00C7	FD		003250		MOV	A,SRO	;AND TEST
00C8	C6CB		003260		JZ	\$2	
00CA	85		003270		CLR	FD	
00CB			003280	\$2:			
00CB	EBBF		003290		DJNZ	RD,CLEAR	
00CD	22		003300		IN	A,DBB	;CLEAR IBF
			003310				
			003320				;PRESET POINTERS,TIMER
			003330				
00CE	BD10		003340		MOV	QTPTR,#FIFO	;FIFO ADDRESS
00DD	BE10		003350		MOV	QHPTR,#FIFO	
			003360				;DIAGNOSTIC CHECK SUM ROUTINE
00D2	27		003370	ROMCK:	CLR	A	
00D3	A8		003380		MOV	RD,A	
00D4			003390	\$1:			
00D4	A3		003400		MOV	A,#A	;GET A ROM BYTE
00D5	6A		003410		ADD	A,R2	
00D6	AA		003420		MOV	R2,A	;SAVE SUM
00D7	18		003430		INC	RD	
00D8	F8		003440		MOV	A,RO	;REPEAT 256 TIMES
00D9	96D4		003450		JNZ	\$1	

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	000B	34F9	003460		CALL	CKSU1	;CONTINUE IN PAGE 1
	000D	94F9	003470		CALL	CKSU4	;CONTINUE IN PAGE 1
	000F	D4F9	003480		CALL	CKSU6	;CONTINUE IN PAGE 1
	00E1		003490	CKSU3:			
	00E1		003500	\$1:			;COMPUTE 3RD PAGE
	00E1	E3	003510		MOVFB	A,5A	;GET A ROM BYTE
	00E2	6A	003520		ADD	A,R2	
	00E3	AA	003530		MOV	R2,A	;SAVE SUM
	00E4	18	003540		INC	RO	
	00E5	F8	003550		MOV	A,RO	;REPEAT 256 TIMES
	00E6	96E1	003560		JNZ	\$1	
	00E8	FA	003570		MOV	A,R2	;TEST RESULT
	00E9	C6EC	003580		JZ	#+3	;GOOD
	00EB	85	003590		CLR	FO	
	00EC	8820	003600		MOV	RO,#STRAPS	;CHANGE GOOD FLAG
			003610			;READ NOW STRAPPING INFO	
			003620				
	00EE	09	003630		IN	A,P1	
	00EF	4308	003640		ORL	A,#8	;SET GOOD FLAG
	00F1	86F5	003650		JFO	#+4	;SKIP
	00F3	53F7	003660		ANL	A,#-9	;CLEAR BIT3 TO BAD STATUS
	00F5	9AFD	003670		ANL	P2,#-3	;SET PORT2 TO LAST COLUMN+1
	00F7	85	003680		CLR	FO	;CLEAR FLAG AGAINST MULTIPLE KEY ENT
	00F8	AD	003690		MOV	5RO,A	
	00F9	F2FE	003700		JB7	\$2	;TEST PLL-CABLE PLUGGED
	00FB	05	003710		EN	I	;ELSE ALLOW IRF INTERRUPTS VIA POLL
	00FC	2404	003720		JMP	#\$	
			003730				
	00FE		003740			\$2:	
	00FE	55	003750		STRT	T	;INITIATE TIMER
	00FF	25	003760		EN	TCNTI	;AND ALLOW TO INTERRUPT
	0100	B81A	003770		MOV	RO,#26	;POINT TO ERRCH IN REG.BANK 1
	0102	B023	003780		MOV	5RO,#35	;SET BEEP MULTIPLIER
	0104		003790			\$3:	
	0104	B824	003800		MOV	RO,#DECPOI	;PRESET TO COMMA
	0106	B02C	003810		MOV	5RO,#2CH	
	0108	17	003820		INC	A	
	0109	37	003830		CPL	A	
	010A	5211	003840		JB2	NEWS	
	010C	07	003850		DEC	A	
	010D	5211	003860		JB2	NEWS	
	010F	B02E	003870		MOV	5RO,#2EH	;CHANGE TO POINT

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
			003890				*****ACTUAL SCAN*****
			003900				;
			003910				;A NEW SCAN STARTS AT NEWS.
			003920				;THE 4 LSB'S OF PORT2 SELECTS A COLUMN OF 8 KEYS.
			003930				;PORT1 INPUTS THEM DATA FROM MATRIX ROWS.
			003940				;KEYS ACTUATED IN COLUMN SELECTED RESPONDS WITH
			003950				;LOGICAL 1. DATA IS SAVED IN REGISTER 7.
			003960				;A 2 BIT HISTORY IS RESERVED FOR EACH KEY IN THE
			003970				;MATRIX.LSB WILL BE SET AS LONG A KEY HAS BEEN FOUND
			003980				;ACTIVE, 2ND BIT INDICATES STATUS KEY HAS GONE FIRST
			003990				;TIME DOWN OR UP. PROPER RECOGNITION OF KEY IS AC-
			004000				;CEPT AFTER FIRST KEY GOING DOWN AND WITH NEXT
			004010				;SCAN CYCLE THIS KEY IS FURTHERON ACTIVE. SUCH WAY
			004020				;NOISE AND DEBOUNCE IS PERFORMED OVER A WHOLE SCAN CYCLE.
			004030				;4 KEYS ARE CONTAINED IN EACH BYTE OF STATUS ARRAY.
			004040				;A TABLE POINTER IS USED FOR GETTING MARKBITS FOR EACH KEY
			004050				;AND ALSO CODE.SYNCHRONIZATION IS FROM TABLEPOINTER TO
			004060				;STATUS POINTER EVERY 4TH KEY AND TO MATRIXCOLUMN EVERY
			004070				;8TH KEY.
			004080				;KEYS ACCEPTED ARE SELECTED IN GROUPS:
			004090				;SHIFT-,CONTROL-,FUNCTION-,ALPHA-,NUMERIC-,SPECIAL-KEYS
			004100				;THE INDICATION WHERE ANY KEY ACCOMPONIES IS CONTAINED IN
			004110				;MARKBIT TABLE.
			004120				;THE SHIFT- AND CONTROL-KEYS ACTIVATED ARE SETTING FLAGB
			004130				;(20H,40H,80H) WHICH ARE STORED AT ADDR.21H.
			004140				;SHIFT CODE 20H IS EXORED WITH UNSHIFT ASCII CODE IF MARK-
			004150				;BIT ENABLES. CONTROL CODE 40H IS EXORED WITH UNSHIFTED
			004160				;ASCII CODE ALSO ENABLED BY MARKBIT.IF BOTH SHIFT AND CTL-
			004170				;FLAG ARE SET CODE OUTPUT IS DISABLED.
			004180				;IF CAPS' LOCK FLAG (80H) IS SET ,THEN UPPER AND LOWER
			004190				;CASE CHARACTERS A..Z ARE EXCHANGED SYNCHRONOUS WITH
			004200				;SHIFT POSITION,E.G. LOWER CASE IN UNSHIFT- AND UPPER CASE
			004210				;LETTERS IN SHIFT MODE.IF LOCK KEY IS RELEASED VICE
			004220				;VERSA.
			004230				;FOR COUNTRY VERSIONS WHICH REQUIRES MODIFIED LAYOUTS
			004240				;OF KEY DISPLACEMENTS THE STRAPPING SWITCHES ARE SET TO
			004250				;A SPECIFIED COUNTRY CODE (0...7).BEFORE TRANSLATIONING
			004260				;THE TABLE OFFSET INTO ASCII-CODES, DEPENDING ON STRAP-
			004270				;INFO A SEARCH ROUTINE IS EXERCISED WHETHER THE ACCORDING
			004280				;KEYPOSITION IS CONTAINED IN THE MODIFY TABLE OR NOT.
			004290				;IF TRUE ,THEN THE INSTANT OFFSET IS REPLACED BY ANOTHER
			004300				;AS SPECIFIED.A ZERO CONTENT IN THESE TABLES MEANS END OF
			004310				;TABLE CONDITION AND POINTER REMAINS UNCHANGED.
			004320				;MOST KEYS ARE AUTO REPEATABLE ELSE ASSOCIATED MARKBIT
			004330				;DETERMINES OTHERWISE. AFTER KEY FIRST GOING DOWN A LONG
			004340				;REPEAT DELAY IS SET(600MS),IF KEY IS HOLD DOWN FURTHERON
			004350				;THEN ACCORDING ASCII CODES ARE ISSUED DEPENDING ON
			004360				;SHIFT AND CONTROL KEY STATUS,WITH AN REPEAT RATE OF
			004370				;APPROX. 12 HZ.
			004380				;CODES TO BE ISSUED TO A MAIN STATION ARE FIRST PUT INTO
			004390				;A BUEUE WHICH WORKS LIKE A FIFO(FIRST IN FIRST OUT)
			004400				;STORAGE BUFFER.
			004410				;THIS USEFUL FOR THE N-KEY ROLL OVER FEATURE WITH FAST
			004420				;KEY DEPRESSIONS AND GIVES THE MAIN STATION SUFFICIENT
			004430				;TIME TO POLL THE DATA FROM KEYBOARD AND INCREASES PRO-

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
			004440				;CESSOR THRUPUT BY NONINTERUPTIVE INTERFACING.
			004450				;IF FIFO GOES OVERFLOW, LAST ENTRY IS LOST.
			004460				;THE KEYS IN THE F-ROW ARE PROVIDED FOR FUNCTION KEY
			004470				;ASSIGNMENTS ITS CODES ARE FOR FREE PROGRAMMING AND
			004480				;ALL SWIFT MODES ARE ALLOWED.
			004490				;
			004500				NEWS:
	0111		004510		MOV	STPTR, #64	;TOP STATUS BYTE
	0113	B940	004520		MOV	TBLPTR, #KEYS	;LAST KEY ADDRESS
	0115	CC	004530		DEC	REPLY	;COUNT DOWN REPEAT DELAY EVERY NEWS
	0116	8A00	004540		ORL	P2, #13	
	0118		004550			SCAN:	
	0118	97	004560		CLR	C	;RESET CARRY FOR POSSIBLE ROTATES TH
	0119	CB	004570		DEC	TBLPTR	
	011A	FB	004580		MOV	A, TBLPTR	
	011B	F211	004590		JB7	NEWS	;TABLE POINTER UNDER FLOW
	011D	37	004600		CPL	A	;TEST BIT0+1 FOR STATUSPOINTER SYNCH
	011E	122E	004610		JBO	\$1	
	0120	322E	004620		JB1	\$1	
	0122	C9	004630		DEC	STPTR	;POINT TO NEXT 4KEY GROUP
	0123	522E	004640		JB2	\$1	;ODD, SKIP MULTIPLEXER UPDATE
	0125	65	004650		STOP	TCNT	;TISR MAY NOT OCCUR IN NEXT ROUTINE
	0126	0A	004660		IN	A, P2	;ELSE DO
	0127	07	004670		DEC	A	
	0128	3A	004680		OUTL	P2, A	
	0129	55	004690		STRT	T	;LET TIMER RUN AGAIN
	012A	A3	004700		MOV	A, \$A	;SPEND A SHORT DELAY
	012B	09	004710		IN	A, P1	;READ NEW MATRIX ROW DATA
	012C	37	004720		CPL	A	
	012D	AF	004730		MOV	R7, A	;SAVE INFO FOR SHIFT AROUND
	012E		004740			\$1:	
	012E	FF	004750		MOV	A, R7	;ROTATE KEY DATA SYNC KEY TO SAME BI
	012F	E7	004760		RL	A	
	0130	AF	004770		MOV	R7, A	
	0131	F1	004780		MOV	A, \$STPTR	;ROTATE ALSO STATUS
	0132	E7	004790		RL	A	
	0133	E7	004800		RL	A	
	0134	A1	004810		MOV	\$STPTR, A	
	0135	3257	004820		JB1	UNLOCK	
	0137	1245	004830		JBO	KYDOWN	
	0139	FF	004840		MOV	A, R7	;TEST KEY IS DOWN
	013A	1218	004850		JBO	SCAN	;NO
	013C	B618	004860		JFO	SCAN	;INHIBIT MULTIPLE KEY ENTRIES
	013E	95	004870		CPL	F0	;ELSE SET INHIBITOR
	013F		004880			LOCKY:	
	013F	F1	004890		MOV	A, \$STPTR	;ELSE SET STATUS BITS
	0140	D303	004900		XRL	A, #3	
	0142	A1	004910		MOV	\$STPTR, A	
	0143	2418	004920		JMP	SCAN	
	0145		004930			KYDOWN:	
	0145	FC	004940		MOV	A, REPLY	;LOCK OUT KEY NEXT 8 SCANS
	0146	53F8	004950		ANL	A, #-8	
	0148	D358	004960		XRL	A, #58H	
	014A	C618	004970		JZ	SCAN	;DON'T CHECK KEY ON ACTIVE
	014C	FF	004980		MOV	A, R7	;TEST KEY HOLD DOWN

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0140	37	004990		CPL	A	
	014E	12AF	005000		JBD	REPCK	;THEN CHECK REPEAT COND. NET
	0150	FB	005010		MOV	A, TBLPTR	;ELSE FIRST KEY UP CONDITION
	0151	E3	005020		MOV3	A, SA	;SELECT SHIFT/CTL KEYS
	0152	F7	005030		RLC	A	
	0153	F6F2	005040		JC	RESHF6	;RESET SHIFT FLAGB
	0155	243F	005050		JMP	LOCKY	;ALL OTHERS KEYUP DEBOUNCE
	0157		005060	UNLOCK:			
	0157	85	005070		CLR	F0	;CLEAR MULTIPLE KEY INHIBITOR
	0158	F1	005080		MOV	A, \$STPTR	;CLEAR DEBOUNCE LOCK
	0159	D302	005090		XRL	A, #2	
	015B	A1	005100		MOV	\$STPTR, A	
	015C	37	005110		CPL	A	
	015D	1218	005120		JBD	SCAN	;WAS KEYUP UNLOCK
	015F	FF	005130		MOV	A, R7	;TEST KEY STILL DOWN
	0160	123F	005140		JBD	LOCKY	;NO
	0162		005150	ACCPTRY:			
	0162	BC60	005160		MOV	REPLY, #LONGDLY	;SET UP 600MS DELAY
	0164		005170	REPEAT:			
	0164	B821	005180		MOV	RO, #FLAGB	
	0166	F0	005190		MOV	A, \$RO	
	0167	37	005200		CPL	A	
	0168	5360	005210		ANL	A, #60H	
	016A	C618	005220		JZ	SCAN	IGNORE CTL/SHIFT COINCIDENCE
	016C	B823	005230		MOV	RO, #LASTKY	;POINT TO LASTKY
	016E	FB	005240		MOV	A, TBLPTR	
	016F	AD	005250		MOV	\$RO, A	
	0170	544A	005260		CALL	NATWLN6	;CHECK NATIVE LANGU. XLAT
	0172	B822	005270		MOV	RO, #LASTCD	;INTERIM SAVE
	0174	AD	005280		MOV	\$RO, A	
	0175	C8	005290		DEC	RO	;POINT TO FLAGB
	0176	F2E1	005300		JB7	SPZEXC	
	0178	E3	005310		MOV3	A, SA	;GET MARKBIT INFO
	0179	AA	005320		MOV	R2, A	;SAVE INTERIM
	017A	F2C0	005330		JB7	FUKY	;SELECT FUNCTION- SHIFT- CTL-LOCK-KE
	017C	531F	005340		ANL	A, #31	;TEST EXCEPTION INDICATION
	017E	96C6	005350		JNZ	EXCEPT	;IF SO TEST LANGUAGE, SHIFT COND.
	0180	FA	005360		MOV	A, R2	;RELOAD MARKS
	0181		005370	NORMLS:			
	0181	D360	005380		XRL	A, #60H	;ALL OTHERS HANDLE NORMALLY
	0183	AA	005390		MOV	R2, A	;INVERT DISABLES TO ENABLES
	0184	F0	005400		MOV	A, \$RO	;PRESAVE
	0185	5A	005410		MOV	A, \$RO	;GET FLAGB AGAIN
	0186		005420	NO:			;COINCIDE WITH MARKS
	0186	AA	005430		MOV	R2, A	;FLAG IS CODE CHANGE OPERAND
	0187	18	005440		INC	RO	;GET LAST KEY AGAIN
	0188	F0	005450		MOV	A, \$RO	
	0189	D368	005460		ADD	A, #ASCITB	;RELATE TO ASCII TABLE
	018B	E3	005470		MOV3	A, SA	;GET UNSHIFTED CODE
	018C	D298	005480		JB6	#1	;SELECT LETTER CODES SKIP SHIFT/CTL
	018E	2A	005490		XCH	A, R2	
	018F	C698	005500		JZ	#1	
	0191	B296	005510		JB5	#1-2	;SELECT SHIFTS
	0193	FA	005520		MOV	A, R2	;TEST CTL FOR 20..3F CODES
	0194	B218	005530		JB5	SCAN	;IF TRUE IGNORE

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0196	2310	005540		MOV	A,#10H	;MAY BE CHANGED TO 10
	0198		005550	\$1:			
	0198	DA	005560		XRL	A,R2	;MAY CHANGE CODE ACCORDING SHIFT MO
	0199		005570	N1:			
	0199	AA	005580		MOV	R2,A	
	019A	F2A9	005590		JB7	M4	;EXTRACT FUNCTIONAL KEYS
	019C	541E	005600		CALL	TXTCHG	;TEST UPPER/LOWER CASE LETTER EXCHAN
	019E		005610	N3:			
	019E	148A	005620		CALL	LDKBQ	;PUT IT NOW INTO FIFO
			005630				
	01A0	FB	005640		MOV	A,TBLPTR	
	01A1	D30B	005650		XRL	A,#DOUBLO	;SELECT DOUBLE ZERO KEY
	01A3	9618	005660		JNZ	SCAN	
	01A5	148A	005670		CALL	LDKBQ	;IF TRUE OUTPUT A 2ND 0
	01A7	2418	005680		JMP	SCAN	;NEXT KEY
			005690				
	01A9		005700	N4:			
	01A9	D2AD	005710		JB6	N5	
	01AB	4408	005720		JMP	COMD	;SELECT DECIMALPOINTCODES
			005730				
	01AD	04A4	005740	N5:	JMP	N5PO	;GO INTO PAGE 0
			005750				
			005760				
			005770				;CHECK ON REPEAT FUNCTION
	01AF		005780	REPCK:			
	01AF	FC	005790		MOV	A,REPLY	;TEST COUNTER ZERO
	01B0	9618	005800		JNZ	SCAN	;NO,IGNORE
	01B2	FB	005810		MOV	A,TBLPTR	
	01B3	E3	005820		MOV3	A,5A	;GET MARKBITS
	01B4	F218	005830		JB7	SCAN	;EXIT IF NON REPEATABLE KEYS
			005840				
	01B6		005850	\$1:			
	01B6	B823	005860		MOV	RO,#LASTKY	;GET LAST KEY
	01B8	FD	005870		MOV	A,5RD	
	01B9	DB	005880		XRL	A,TBLPTR	;IS INSTANT KEY=LAST KEY OUTPUT
	01BA	9618	005890		JNZ	SCAN	
	01BC	BC08	005900		MOV	REPLY,#REPRAT	;SET REPEAT RATE AGAIN
	01BE	2464	005910		JMP	REPEAT	
			005920				
	01C0		005930	FUKY:			
	01C0	F7	005940		RLC	A	
	01C1	96EE	005950		JNZ	STSHFG	;SET SHIFT,OR CTL OR ALPHALOCK FLAG
	01C3	27	005960		CLR	A	
	01C4	2481	005970		JMP	NORMLS	;RECEIVE WITH 0 MARKS
			005980				
	01C6		005990	EXCEPT:			;TEST ON EXCEPTIONAL CONDITIONS
	01C6	FA	006000		MOV	A,R2	;SELECT SPECIAL EXCHANGES
	01C7	D2E0	006010		JB6	SPZEXC-1	
	01C9	C3	006020		DEC	RO	;LOAD LANGUAGE
	01CA	FD	006030		MOV	A,5RD	
	01CB	5307	006040		ANL	A,#7	
	01CD	0300	006050		ADD	A,#EXCTB	;REFER TO TABLE
	01CF	94E8	006060		CALL	TRANS4	;GET OFFSET
	01D1	6A	006070		ADD	A,R2	;ADD REF.
	01D2	AA	006080		MOV	R2,A	

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	01D3	18	006090		INC	RO	;POINT TO FLAGBYTE
	01D4	F0	006100		MOV	A,SRD	
	01D5	D218	006110		JB6	SCAN	;IGNORE CONTROLMODE
	01D7	5320	006120		ANL	A,#32	;MASK OUT SHIFT MODE
	01D9	77	006130		RR	A	;MAKE 16 BYTE OFFSET
	01DA	6A	006140		ADD	A,R2	
	01DB	94E8	006150		CALL	TRANS4	;GET NOW CODE
	01DD	AA	006160		MOV	R2,A	
	01DE	249E	006170		JMP	N3	
			006180				
	01E0	E7	006190		RL	A	
	01E1		006200	SPZEXC:			
	01E1	AA	006210		MOV	R2,A	
	01E2	F0	006220		MOV	A,SRD	;GET FLAGB
	01E3	5360	006230		ANL	A,#60H	
	01E5	77	006240		RR	A	
	01E6	47	006250		SWAP	A	
	01E7	6A	006260		ADD	A,R2	
	01E8	6A	006270		ADD	A,R2	
	01E9	6A	006280		ADD	A,R2	
	01EA	0384	006290		ADD	A,#SPZXC0-128	
	01EC	C400	006300		JMP	TRANS6	
			006310				
	01EE		006320	STSHFG:			
	01EE	40	006330		ORL	A,SRD	;SET UP BIT
	01EF	A0	006340		MOV	SRD,A	;AND STORE
	01F0	2418	006350		JMP	SCAN	
			006360				
	01F2		006370	RESHFG:			
	01F2	B821	006380		MOV	RO,#FLAGB	
	01F4	37	006390		CPL	A	
	01F5	50	006400		ANL	A,SRD	;CLEAR FLAG
	01F6	A0	006410		MOV	SRD,A	;STORE BACK
	01F7	243F	006420		JMP	LOCKY	
	01F9		006430		ORG	1F9H	
	01F9		006440	CKSU1:			
	01F9		006450	\$1:			
	01F9	A3	006460		MOVP	A,SA	;GET A ROM BYTE
	01FA	6A	006470		ADD	A,R2	
	01FB	AA	006480		MOV	R2,A	;SAVE SUM
	01FC	18	006490		INC	RO	
	01FD	F8	006500		MOV	A,RO	;REPEAT 256 TIMES
	01FE	96F9	006510		JNZ	\$1	
	0200		006520	CKSU2:			
	0200		006530	\$1:			
	0200	A3	006540		MOVP	A,SA	
	0201	6A	006550		ADD	A,R2	
	0202	AA	006560		MOV	R2,A	;SAVE SUM
	0203	18	006570		INC	RO	
	0204	F8	006580		MOV	A,RO	;REPEAT 256 TIMES
	0205	9600	006590		JNZ	\$1	
	0207	83	006600		RET		
	0208		006610	COMD:			
	0208	B824	006620		MOV	RO,#DECP0I	;PRESET SOFTADDRESS POINT/CONHA
	020A	D38A	006630		XRL	A,#8AH	;SELECT ORIGINAL CODE OF DEC.POINT

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	020C	9611	006640		JNZ	\$1	;IS NOT ,SKIP
	020E	F0	006650		MOV	A,5RD	;GET REPLACEMENT
	020F	441B	006660		JMP	TRANS+1	
	0211	D310	006670	\$1:	XRL	A,#16	;SELECT CTRL CODE OF DECPD1.
	0213	961C	006680		JNZ	TRANS+2	;IS ANOTHER FUNCTIONCODE
	0215	F0	006690		MOV	A,5RD	;LOAD CODE AND SWAP POINT/COMMA
	0216	D302	006700		XRL	A,#2	
	0218	24EF	006710		JMP	STSHF6+1	
			006720				;
	021A		006730	TRANS:			
	021A	A3	006740		MOVP	A,5A	
	021B	AA	006750		MOV	R2,A	
	021C	249E	006760	\$1:	JMP	N3	
			006770				;

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	021E		006790	TXTCH6:			
	021E	37	006800		CPL	A	
	021F	D227	006810		JB6	UNCH	;NO LETTER CODE
	0221	C8	006820		DEC	RO	
	0222	F0	006830		MOV	A,SRO	;SET WP-FLAG
	0223	37	006840		CPL	A	
	0224	F228	006850		JB7	UNCH+1	;TRUE
	0226	FA	006860		MOV	A,R2	
	0227		006870	UMCH:			
	0227	83	006880		RET		
	0228		006890	\$1:			
	0228	C8	006900		DEC	RO	
	0229	FA	006910		MOV	A,R2	
	022A	531F	006920		ANL	A,#1FH	
	022C	C642	006930		JZ	\$2	
	022E	03E2	006940		ADD	A,#-30	;SELECT ALL) ü
	0230	F642	006950		JC	\$2	
	0232	0303	006960		ADD	A,#3	;SELECT ALL)Z
	0234	E63D	006970		JMC	\$3	
	0236	F0	006980		MOV	A,SRO	;GET LANGUAGE CODE
	0237	37	006990		CPL	A	
	0238	5227	007000		JB2	UNCH	
	023A	17	007010		IMC	A	
	023B	5227	007020		JB2	UNCH	
	023D		007030	\$3:			
	023D	2320	007040		MOV	A,#32	;SET EXCHANGE OPERATOR
	023F	DA	007050		XRL	A,R2	
	0240	AA	007060		MOV	R2,A	
	0241	83	007070		RET		
	0242		007080	\$2:			
	0242	F0	007090		MOV	A,SRO	;LANGUAGE CODE
	0243	37	007100		CPL	A	
	0244	5307	007110		ANL	A,#7	;SELECT YUGOSLAVIA
	0246	9627	007120		JMZ	UNCH	
	0248	443D	007130		JMP	\$3	;INCLUDE ALSO CODES 40H,60H,5EH,7EH
			007140				

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
			007160		;		
			007170		;	LANGUAGE DEPENDENT KEY LOCATION EXCHANGE	
			007180		;		
			007190		;		
			007200		;	CH1(0) CH2(1) Fr(2) Can1(3)Can2(4) SA(5) Po(6) Yu(7)	
			007210		;		
			007220		;	E10^S10	
			007230		;	E11^S05 E11^S4	
			007240		;	E12^S9 E12^S4 E12^E11 E12^S11 E12^D12 E12^C12	
			007250		;	D11^S2 D11^D12 D11^C11 D11^S16 D11^C11	
			007260		;	D12^S3 D12^C12 D12^D11 D12^C10 D12^C12 D12^C10	
			007270		;	C10^S1 C10^B7 C10^C12 C10^D11 C10^D12	
			007280		;	C11^S0 C11^S7 C11^S14 C11^S11 C11^C10 C11^D11	
			007290		;	C12^S8 C12^S15 C12^B0 C12^S11 C12^B10	
			007300		;	C1^D1 C1^D1	
			007310		;	D1^ C1 B0^C10 B00^S13 D1^C1	
			007320		;	B1^D6 B1^D2 B1^D2 B1^D6	
			007330		;	B07^E11	
			007340		;	B10^E10 B10^B00 B10^S17 B10^E12	
			007350		;	D6^B1 D2^B1 D2^B1 D6^B1	
024A			007360			HATVLNG:	
024A	AA		007370		MOV	R2,A ;SAVE TABLE OFFSET	
024B	B820		007380		MOV	RO,#STRAPS	
024D	FD		007390		MOV	A,SRO ;GET LANGUAGE CODE	
024E	5307		007400		ANL	A,#7	
0250	0362		007410		ADD	A,#LNGOFS ;SET OFFSET TO TABLES	
0252	A3		007420		MOV	A,SA	
0253	A8		007430		MOV	RO,A ;SAVE IN REG	
0254			007440		\$1:		
0254	F8		007450		MOV	A,RO	
0255	A3		007460		MOV	A,SA ;ADDRESS INTO PROG.MEMORY	
0256	C626		007470		JZ	UNCH-1 ;END OF TABLE REACHED	
0258	DA		007480		XRL	A,R2 ;COMPARE WITH ORIGINAL DISPLACEMENT	
0259	18		007490		INC	RO	
025A	C65F		007500		JZ	\$2 ;FOUND CONDITION	
025C	18		007510		INC	RO ;SKIP TO NEXT CODE	
025D	4454		007520		JMP	\$1 ;LOOP	
025F			007530		\$2:		
025F	F8		007540		MOV	A,RO ;ADDRESS TO SUBSEQUENT CODE	
0260	A3		007550		MOV	A,SA ;LOAD REPLACE CODE	
0261	83		007560		RET		
0262			007570		LNGOFS:		
0262	6A708CB38FD2		007580		DB	\$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8	
	9E79						
026A			007590		\$1:		
026A	5D8249815980		007600		DB	93,130,73,129,89,128 ;SWISS(GERMAN)	
			007610			D11S2,C10S1,C11S0	
0270			007620		\$2:		
0270	3E893D83452A		007630		DB	62,137,61,131,69,42,42,69,0 ;SWISS 2	
	2A4500						
			007640			E12S9,D12S3,D6B1,B1D6	
0279			007650		\$8:		
0279	452A2A45394A		007660		DB	69,42,42,69,57,74,61,73,62,57 ;YUGOSLAVIA	
	3D493E39						
			007670			D6B1,B1D6,C12B10,D12C10,E12C12	

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0283	5D59595D493D 4A3E00	007680		DB	93,89,89,93,73,61,74,62,0	
			007690 ;			D11C11,C11011,C10012,810E12	
	028C		007700	03:		;FRENCH	
	028C	4A4E525E4952 5987	007710		DB	74,78,82,94,73,82,89,135	
			007720 ;			810E10,87E11,C1087,C1157	
	0294	3988503D4E8A 5E853E84	007730		DB	57,136,93,61,78,138,94,133,62,132	
			007740 ;			C12S8,D11D12,E10S10,E11S5,E12S4	
	029E		007750	07:		;PORTUGUESE	
	029E	2A25252A292D 2D29398B495D	007760		DB	42,37,37,42,41,45,45,41,57,139,73,93	
			007770 ;			81D2,02B1,C1D1,D1C1,C12S11,C10D11	
	02AA	59495D593D39 3E3D00	007780		DB	89,73,93,89,61,57,62,61,0	
			007790 ;			C11C10,D11C11,D12C12,E12D12	
	02B3		007800	04:		;CANADA	
	02B3	0A494A0A598E 398F5D593D5D	007810		DB	10,73,74,10,89,142,57,143,93,89,61,93	
			007820 ;			80C10,810B0,C11S14,C12S15,D11C11,D12D11	
			007830 ;			CANADA(BILINGUAL)	
	02BF	5D9D5E843E5E 49395988390A 0A8D	007840	05:	DB	93,144,94,132,62,94,73,57,89,139,57,10,10,141	
			007850 ;			D11S16,E11S4,E12E11,C10C12,C11S11,C12B0,80,S13	
	02CD	3D494A9100	007860		DB	61,73,74,145,0	
			007870 ;			D12C10,810S17	
	02D2		007880	06:		;AFRIKAANS	
	02D2	3E8800	007890		DB	62,139,0	
			007900 ;			E12S11	

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
			007920				; SPECIAL EXCEPTIONS:
			007930				;
			007940				; OFFS RELOC. LANGU. CODES SYMBOLS POSITION
			007950		00	80 01 78 58 1B	Å § C11
			007960		01	81 01 7C 5C 1C	œ 'e C10
			007970		02	82 01 70 5D 1D	ue e' D11
			007980		03	83 03 7E 27 1E	ß ' D12
			007990		04	84 FF 2D 5F 1F	- - B10,E11
			008000		05	85 04 29 5B 1B) o E11
			008010		06	86 FF 2D F4 F5	- B51
			008020		07	87 04 7C 25 1C	u' Z C11
			008030		08	88 04 60 23 1D	' ð C12
			008040		09	89 03 5E 60 00	^ ' E12
			008050		10	8A 04 40 3D 00	a' 0 E10
			008060		11	8B 40 27 6D 00	' ' C12
			008070		12	46 FF 86 7F 96	CLR D50
			008080		13	8D 18 5D 5B 1D	ü Å D11,80
			008090		14	8E 08 27 22 00	' " C11
			008100		15	8F 08 60 7E 1E	' ß C12
			008110		16	90 10 23 5E 1E	÷ ^ D11
			008120		17	91 10 7B 7D 1B	'e e' B10
			008130				;
			008140				; MARKBIT 6 OR 7 ON MEANS EXCEPT IN ALL MODES
0600			008150			ORG 600H	
0600			008160			TRANS6:	
0600	A3		008170		MOVF	A,5A	;GET BYTE FROM TABLE
0601	AA		008180		MOV	R2,A	
0602	249E		008190		JMP	W3	
			008200				;
0604			008210			SPZXC:	
0604	7B5B1B7C5C1C		008220		DB	7BH,91,27,7CH,92,28,7DH,93,29,7EH,39,30	
	7D5D1D7E271E						
0610	2D5F1F295B1B		008230		DB	45,95,31,41,91,27,45,244,245,124,37,28,96,35,29	
	2DF4F57C251C						
	6D231D						
061F	5E6000403000		008240		DB	94,96,,64,48,,39,96,,134,127,96H,93,91,29	
	276D00867F96						
	5D5B1D						
062E	272200607E1E		008250		DB	39,34,,96,126,30,35,94,30,123,125,27	
	235E1E7B7D1B						
			008260				;

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	0300		008280		ORG	300H	
	0300		008290	HRKTB:			
	0300	A00000000000	008300		DB	160,,,,,4,128	;D13C84D51D53DEF4
		0480					
	0308	60C00D6020A0	008310		DB	96,-64,13,96,32,160,96,128	;A50C80A5253DEF0
		6080					
	0310	460000000000	008320		DB	70,,,,,3,128	;D50C83D5254DEF3
		0380					
	0318	600000000000	008330		DB	96,,,,,5,128	;AC85C5153DEF5
		0580					
	0320	900000002000	008340		DB	144,,,,,32,,2,128	;899C82B5154DEF2
		0280					
	0328	000000000000	008350		DB	0,,,,,1,128	;850C81B5253DEF1
		0180					
	0330	20000F323500	008360		DB	32,,15,50,53,,9,128	;E13C89X1X2DEF9
		0980					
	0338	800C20000000	008370		DB	128,12,32,,,,,128	;F13C1213..DEF12
		0080					
	0340	430000000000	008380		DB	67,,,,,6,128	;C50C86C5254DEF6
		0680					
	0348	800042802000	008390		DB	128,,66,128,20H,,10,128	;F50C810FE52DEF1
		0A80					
	0350	600008020000	008400		DB	96,,128,32,,7,128	;E14C87FE53DEF7
		0780					
	0358	800090802000	008410		DB	128,,144,128,32,,11,128	;F14C811FE51DEF1
		0B80					
	0360	20000E802000	008420		DB	32,,14,128,32,,8,128	;E50C88FE54DEF8
		0880					
	0068		008430	KEYS	EQU	\$_HRKTB	
	0368		008440	ASCITB:			
	0368	004656F7F952	008450		DB	0,'FV',-9,-7,'R4',-28	;D13C84D5153DEF4
		34E4					
	0370	30003C308A00	008460		DB	48,'0',8AH,,27,-32	;A50C80A5253DEF0
		1BE0					
	0378	864443F8FF45	008470		DB	134,'DC',-8,-1,'E3',-29	;D50C83D5254DEF3
		33E3					
	0380	204742F4F654	008480		DB	'6B',-12,-10,'T5',-27	;AC85C5153DEF5
		35E5					
	0388	005358FC8857	008490		DB	'SX',-4,88H,'W2',-30	;899C82B5154DEF2
		32E2					
	0390	FB415AFDFE51	008500		DB	-5,'AZ',-3,-2,'01',-31	;850C81B5253DEF1
		31E1					
	0398	8B4C2E37304F	008510		DB	139,'L.7009',-23	;E13C89X1X2DEF9
		39E9					
	03A0	ED28883D3B5E	008520		DB	-19,43,136,'=;^5',-20	;F13C1213....DEF12
		4DEC					
	03A8	2D484EF5FA59	008530		DB	45,'HN',-11,-6,'Y6',-26	;C50C86C5254DEF6
		36E6					
	03B0	EF5C2DF18350	008540		DB	-17,'0',-15,131,80,48,-22	;F50C810FE52DEF10
		30EA					
	03B8	094A4DF28455	008550		DB	9,'JM',-14,84H,'U7',-25	;E14C87FE53DEF7
		37E7					
	03C0	EE5800F0825D	008560		DB	-18,91,,-16,130,93,47,-21	;F14C811FE51DEF1
		2FEB					
	03C8	81482CF38549	008570		DB	81H,'K',-13,85H,'I8',-24	;E50C88FE54DEF8

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
		38E8					
	0400		008580		ORG	400H	
	0400	080828486888	008590	EXCTB:	DB	\$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8	
		A8C8					
	0408		008600	\$1:			
	0408	003132333435	008610	\$2:	DB	'1234567890!\$(,.,+*%&/()=?*);:'	
		363738393021					
		243C2C2E002B					
		222A4025262F					
		28293D3F233E					
		3B3A					
	0428	002678222728	008620	\$3:	DB	'&ä"',39,'(üü!0=,\$(:;123456789+?*).'/'	
		5D7D215C3D2C					
		243C383A0031					
		323334353637					
		38392B3F2A3E					
		2E2F					
	0448	003132333435	008630	\$4:	DB	'1234567890=:/,.,!\$%&*^&#()=?*()'	
		36373839303D					
		382F2C2E0021					
		402324255E26					
		2A28292B3A3F					
		3C3E					
	0468	003132333435	008640	\$5:	DB	'1234567890=;/\$,.,!/%&*^&#()+B()'	
		36373839303D					
		3B4D2C2E0021					
		222F24253F26					
		2A28292B3A7E					
		3C3E					
	0488	003132333435	008650	\$6:	DB	'1234567890/!(,.,+*%&*()=?*);:'	
		36373839302F					
		213C2C2E002B					
		22234025262A					
		28293D3F243E					
		3B3A					
	04A8	203132333435	008660	\$7:	DB	'1234567890&+,(.,!""%&/()=?*);:'	
		363738393040					
		2B3C2C2E0021					
		22232425262F					
		28293D3F2A3E					
		3B3A					
	04C8	203132333435	008670	\$8:	DB	'1234567890',39,'+(,.,!""%&/()=?*);:'	
		363738393027					
		2B3C2C2E0021					
		22232425262F					
		28293D3F2A3E					
		3B3A					
			008680				
	04E8		008690	TRANS4:			
	04E8	A3	008700		MOV	A,5A	
	04E9	B3	008710		RET		
			008720				

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CMD	OPERAND	COMMENT
	04F9		008740		ORG	4F9H	
	04F9		008750	CKSU4:			
	04F9		008760	\$1:			
	04F9	A3	008770		MOVP	A,5A	;GET A ROM BYTE
	04FA	6A	008780		ADD	A,R2	
	04FB	AA	008790		MOV	R2,A	;SAVE SUM
	04FC	18	008800		INC	RO	
	04FD	F8	008810		MOV	A,RO	;REPEAT 256 TIMES
	04FE	96F9	008820		JNZ	\$1	
	0500		008830	CKSU5:			
	0500		008840	\$1:			
	0500	A3	008850		MOVP	A,5A	
	0501	6A	008860		ADD	A,R2	
	0502	AA	008870		MOV	R2,A	;SAVE SUM
	0503	18	008880		INC	RO	
	0504	F8	008890		MOV	A,RO	;REPEAT 256 TIMES
	0505	9600	008900		JNZ	\$1	
	0507	83	008910		RET		
			008920 ;				
	06F9		008930		ORG	6F9H	
	06F9		008940	CKSU6:			
	06F9		008950	\$1:			
	06F9	A3	008960		MOVP	A,5A	;GET A ROM BYTE
	06FA	6A	008970		ADD	A,R2	
	06FB	AA	008980		MOV	R2,A	;SAVE SUM
	06FC	18	008990		INC	RO	
	06FD	F8	009000		MOV	A,RO	;REPEAT 256 TIMES
	06FE	96F9	009010		JNZ	\$1	
	0700		009020	CKSU7:			
	0700		009030	\$1:			
	0700	A3	009040		MOVP	A,5A	
	0701	6A	009050		ADD	A,R2	
	0702	AA	009060		MOV	R2,A	;SAVE SUM
	0703	18	009070		INC	RO	
	0704	F8	009080		MOV	A,RO	;REPEAT 256 TIMES
	0705	9600	009090		JNZ	\$1	
	0707	83	009100		RET		

ERR-C ADDR OBJ-CD

PG-LIN LABEL CHG OPERAND

COMMENT

```

009120 ;
009130 ;CROSS REFERENCE WIRING/KEY LOCATIONS
009140 ;
009150 ; COLUMN/ROW RD R1 R2 R3 R4 R5 R6 R7
009160 ;
009170 ; 0 COL 0 D13 C4 B4 D51 D53 D4 E4 F4
009180 ;
009190 ; 1 COL 1 A50 C0 B00 A52 +A53 D0 E0 F0
009200 ;
009210 ; 2 COL 2 +D50 C3 B3 D52 D54 D3 B3 F3
009220 ;
009230 ; 3 COL 3 A5 C5 B5 C51 C53 D5 E5 F5
009240 ;
009250 ; 4 COL 4 B99 C2 B2 B51 +B54 B02 B2 F2
009260 ;
009270 ; 5 COL 5 B50 B01 B01 B52 B53 D1 E1 F1
009280 ;
009290 ; 6 COL 6 +E13 C9 B09 X1 X2 D9 B09 F9
009300 ;
009310 ; 7 COL 7 F13 B012 +C13 X3 X4 B012 B012 F12
009320 ;
009330 ; 8 COL 8 C50 C6 B6 C52 C54 B06 B06 F6
009340 ;
009350 ; 9 COL 9 F50 B010 B010 F52 +E52 D10 B010 F10
009360 ;
009370 ; A COL 10 E14 C7 B7 F53 +E53 D7 B07 F7
009380 ;
009390 ; B COL 11 F14 B011 B11 F51 +E51 B011 B011 F11
009400 ;
009410 ; C COL 12 +E50 C8 B08 F54 +E54 D8 B08 F8
009420 ;
009430 ;PSEUDO 12 X5 X6 X7
009440 ;
009450 ; KEY LOCATIONS WITH B ARE RELOCATED LANGUAGE DEPENDENT
009460 ; KEY LOCATIONS WITH + TRANSLATABLE FOR 7900,7901 USAGE

```

ERR-C	ADDR	OBJ-CO	PG-LIN	LABEL	CMD	OPERAND	COMMENT
			009480				;NUMERIC KEY PAD PART
			009490				;EXPLANATION: 1ST LINE KEYLOCATION 2ND MATRIXPOSITION
			009500				; 3RD LINE MARKBITS 4TH SYMBOLS
			009510				; 5TH LINE CTL-CODE, 6TH SHIFT-CODE, 7TH UNSHIFT
			009520				-----MATRIXCODE COMBINED:
			009530				!F50!F51!F52!F53!F54!
			009540				! 90! 83! 93! A3! C3!
			009550				! 80! 80! 80! 80! 80!
			009560				!F16!F17!F18!F19!F20!
			009570				! AF! 80! B1! B2! B3!
			009580				! CF! D0! D1! D2! D3!
			009590				! EF! F0! F1! F2! F3!
			009600				!-----!
			009610				!E50!E51!E52!E53!E54!
			009620				! C0! B4! 94! A4! C4!
			009630				! 20! 20! 20! 20! 20!
			009640				Home! (--!Dun! Up!--)!
			009650				! 91! 92! 93! 94! 95!
			009660				! 81! 82! 83! 84! 85!
			009670				! 81! 82! 83! 84! 85!
			009680				!-----!
			009690				!D50!D51!D52!D53!D54!
			009700				! 20! 03! 23! 04! 24!
			009710				! 46! 00! 00! 00! 00!
			009720				!CLR!7 !8 !9 !/ !
			009730				! 96! 87! 88! 89! 8F!
			009740				! 7F! D7! D8! D9! DF!
			009750				! 86! 37! 38! 39! 2F!
			009760				!-----!
			009770				!C50!C51!C52!C53!C54!
			009780				! 80! 33! 83! 34! 84!
			009790				! 43! 00! 00! 00! 00!
			009800				!- !4 !5 !6 !* !
			009810				! F5! B4! B5! B6! BA!
			009820				! F4! D4! D5! D6! DA!
			009830				! 20! 34! 35! 36! 2A!
			009840				!-----!
			009850				!B50!B51!B52!B53!B54!
			009860				! 50! 43! 53! 54! 44!
			009870				! 00! 00! 00! 00! 20!
			009880				!+ !1 !2 !3 !F! !
			009890				! B8! BC! BD! BE! 98!
			009900				! DB! DC! DD! DE! 88!
			009910				! 28! 31! 32! 33! 88!
			009920				!-----!
			009930				!A50 !A52!A53! !
			009940				! 10 ! 13! 14! !
			009950				! 60 ! 60! 20! !
			009960				! 0 !00 !, ! !
			009970				! 30 !*30! 9A!(-----USED TO SWAP FM . TO ,
			009980				! 30 !*30! 8A! ! AND BACK
			009990				! 30 !*30! 8A!(-----OUTPUT: 2E(.) or 2C(+)
			010000				-----

ERR-C	ADDR	OBJ-CD	PG-LIN	LABEL	CHD	OPERAND	COMMENT
	010560						SWISS-ROMANDE
	010570						EXPLANATION: 1ST LINE KEYLOCATION 2ND MATRIXPOSITION
	010580						3RD LINE MARKBITS 4TH SYMBOLS
	010590						5TH LINE CTL-CODE, 6TH SHIFT-CODE, 7TH UNSHIFT
	010600						-----
	010610						F00!F01!F02!F03!F04!F05!F06!F07!F08!F09!F10!F11!F12!F13!F14!
	010620						17! 57! 47! 27! 07! 37! 87! A7! C7! 67! 97! B7! 77! 70! 80!
	010630						80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80!
	010640						F1! F2! F3! F4! F5! F6! F7! F8! F9!F10!F11!F12!F13!F14!F15!
	010650						A0! A1! A2! A3! A4! A5! A6! A7! A8! A9! AA! AB! AC! AD! AE!
	010660						C0! C1! C2! C3! C4! C5! C6! C7! C8! C9! CA! CB! CC! CD! CE!
	010670						E0! E1! E2! E3! E4! E5! E6! E7! E8! E9! EA! EB! EC! ED! EE!
	010680						-----
	010690						E00!E01!E02!E03!E04!E05!E06!E07!E08!E09!E10!E11!E12!E13!E14!
	010700						16! 56! 46! 26! 06! 36! 86! 46! C6! 66! 96! B6! 76! 60! A0!
	010710						60! 01! 02! 03! 04! 05! 06! 07! 08! 09! 0A! 0B! 0C! 0D! 0E!
	010720						ESC!1+ !2* !3# !4c;!5Z !6\$!7/ !8(!9) !0= !!? !^ !BSP!TAB!
	010730						1B! 1C! 1D! 1E! 1F! 1G! 1H! 1I! 1J! 1K! 1L! 1M! 1N! 1O!
	010740						1B! 2B! 22! 2A! 40! 25! 26! 2F! 2B! 29! 30! 3F! 60! 8B! 09!
	010750						1B! 31! 32! 33! 34! 35! 36! 37! 38! 39! 30! 21! 5E! 8B! 09!
	010760						-----
	010770						000 !001!002!003!004!005!006!007!008!009!D10!D11!D12! D13 !
	010780						15! 55! 45! 25! 05! 35! 85! A5! C5! 65! 95! B5! 75! 00 !
	010790						A0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! S3! A0 !
	010800						CTRL!q0 !wW !eE !rR !tT !yY !uU !iI !oO !pP !eE!B! !CTRL !
	010810						! 11! 17! 05! 12! 14! 1A! 15! 09! 0F! 10! 1D! 1E! !
	010820						! 51! 57! 45! 52! 54! 5A! 55! 49! 4F! 70! 7D! 27! !
	010830						! 71! 77! 65! 72! 74! 7A! 75! 69! 6F! 50! 5D! 7E! !
	010840						-----
	010850						C00 !C01!C02!C03!C04!C05!C06!C07!C08!C09!C10!C11!C12!C13!
	010860						! 11 ! 51! 41! 21! 01! 31! 81! A1! C1! 61! 91! B1! 71! 72!
	010870						! C0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 0C! 20!
	010880						!LOCK^!aA !sS !dD !fF !gG !hH !jJ !kK !L !eE!a!s!# ! !
	010890						! 01! 13! 04! 06! 07! 08! 0A! 0B! 0C! 1C! 1B! ! 98!
	010900						! 41! 53! 44! 46! 47! 48! 4A! 4B! 4C! 7C! 7B! 23! 8B!
	010910						! 61! 73! 64! 66! 67! 68! 6A! 6B! 6C! 5C! 5B! 24! 8B!
	010920						-----
	010930						B99!B00!B01!B02!B03!B04!B05!B06!B07!B08!B09!B10! B11 !
	010940						! 40! 12! 52! 42! 22! 02! 32! 82! A2! C2! 62! 92! B2 ! !
	010950						! 90! 00! 00! 00! 00! 00! 00! 00! 00! 0E! 0F! 0A! 00! ! !
	010960						!SHIFT(! !yY !xX !cC !vV !bB !nN !mM !; ; ! : ! _ !SHIFT^!
	010970						! 19! 18! 03! 16! 02! 0E! 00! ! ! 1F! ! !
	010980						! 3E! 59! 58! 43! 56! 42! 4E! 4D! 3B! 3A! 5F! ! !
	010990						! 3C! 79! 78! 63! 76! 62! 6E! 6D! 2C! 2E! 2D! ! !
	011000						-----
	011010						A05
	011020						30
	011030						60
	011040						
	011050						20
	011060						20
	011070						20
	011080						-----


```

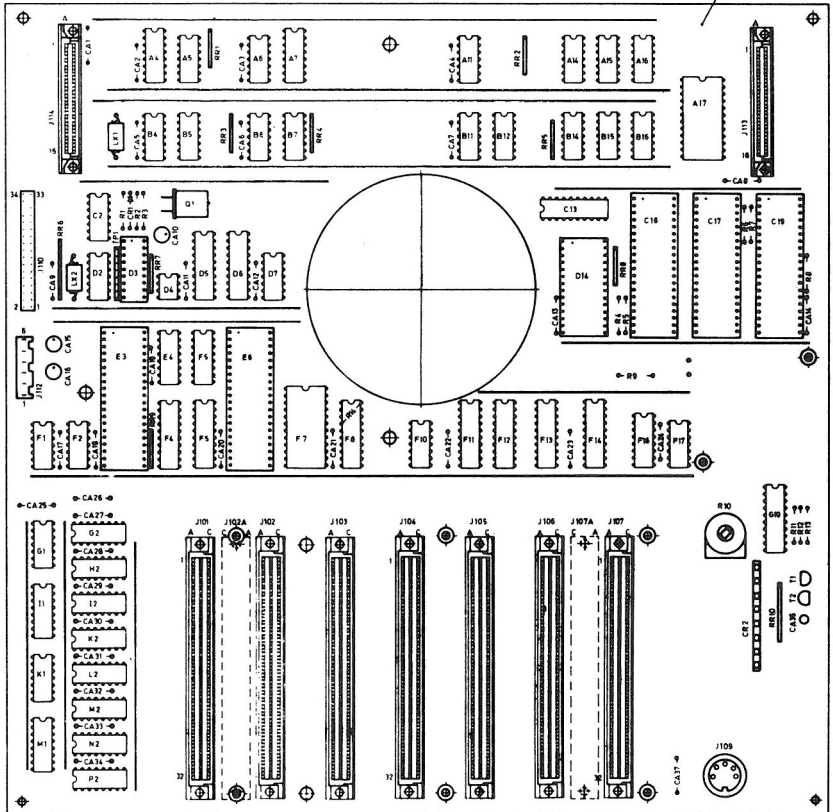
013260 ;PORTUGUESE
013270 ;EXPLANATION: 1ST LINE KEYLOCATION 2ND MATRIXPOSITION
013280 ;      3RD LINE MARKBITS      4TH SYMBOLS
013290 ;      5TH LINE CTL-CODE, 6TH SHIFT-CODE, 7TH UNSHIFT
013300 ;-----
013310 ;F00!F01!F02!F03!F04!F05!F06!F07!F08!F09!F10!F11!F12!F13!F14!
013320 ; 17! 57! 47! 27! 07! 37! 87! A7! C7! 67! 97! B7! 77! 70! B0!
013330 ; 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80! 80!
013340 ; F1! F2! F3! F4! F5! F6! F7! F8! F9!F10!F11!F12!F13!F14!F15!
013350 ; A0! A1! A2! A3! A4! A5! A6! A7! A8! A9! AA! AB! AC! AD! AE!
013360 ; C0! C1! C2! C3! C4! C5! C6! C7! C8! C9! CA! CB! CC! CD! CE!
013370 ; E0! E1! E2! E3! E4! E5! E6! E7! E8! E9! EA! EB! EC! ED! EE!
013380 ;-----+-----+-----+-----+-----+-----+-----+-----+
013390 ;E00!E01!E02!E03!E04!E05!E06!E07!E08!E09!E10!E11!E12!E13!E14!
013400 ; 16! 56! 46! 26! 06! 36! 86! 46! C6! 66! 96! B6! 76! 60! A0!
013410 ; 60! 01! 02! 03! 04! 05! 06! 07! 08! 09! 0A! 0B! 00! 20! 60!
013420 ;ESC!1! 12! 13! 14! 15! 16! 17! 18! 19! 10! 15! 1A! 1BSP!TAB!
013430 ; 18! 1! 1! 1! 1! 1! 1! 1! 1! 1! 1! 1! 1! 1! 1! 1! 1! 1!
013440 ; 18! 21! 22! 23! 24! 25! 26! 2F! 28! 29! 30! 3F! 7E! 8B! 09!
013450 ; 18! 31! 32! 33! 34! 35! 36! 37! 38! 39! 30! 40! 5E! 8B! 09!
013460 ;-----
013470 ; 000 !001!002!003!004!005!006!007!008!009!D10!D11!D12! D13 !
013480 ; 15 ! 55! 45! 25! 05! 35! 85! A5! C5! 65! 95! B5! 75! 00 !
013490 ; A0 !00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! A0 !
013500 ; CTRL!aA !zZ !eE !rR !tT !yY !uU !iI !oO !pP !aBA!+* ! CTRL!
013510 ; ! 01! 1A! 05! 12! 14! 19! 15! 09! 0F! 10! 18! ! !
013520 ; ! 41! 5A! 45! 52! 54! 59! 55! 49! 4F! 50! 5B! 2A! ! !
013530 ; ! 61! 7A! 65! 72! 74! 79! 75! 69! 6F! 70! 7B! 2B! ! !
013540 ;-----
013550 ;!C00 !C01!C02!C03!C04!C05!C06!C07!C08!C09!C10!C11!C12!C13!
013560 ;! 11 ! 51! 41! 21! 01! 31! 81! A1! C1! 61! 91! B1! 71! 72!
013570 ;! C0 ! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 00! 20!
013580 ;!LOCK!qQ !sS !dD !fF !gG !hH !jJ !kK !lL !c,C!oO!P! ! !
013590 ;! ! 11! 13! 04! 06! 07! 08! 0A! 0B! 0C! 10! 16! 00! 9B!
013600 ;! ! 51! 53! 44! 46! 47! 48! 4A! 4B! 4C! 5D! 5C! 6D! 8B!
013610 ;! ! 71! 73! 64! 66! 67! 68! 6A! 6B! 6C! 7D! 7C! 27! 8B!
013620 ;-----FT!
013630 ;!B99!B00!B01!B02!B03!B04!B05!B06!B07!B08!B09!B10! B11 ! !
013640 ;! 40! 12! 52! 42! 22! 02! 32! 82! A2! C2! 62! 92! B2 ! !
013650 ;! 90! 0D! 00! 00! 00! 00! 00! 00! 00! 0E! 0F! 42! 90 ! !
013660 ;!SHF!() !wW !xX !cC !vV !bB !nN !mM !,; !. : _ !SHFT!
013670 ;! ! 17! 18! 03! 16! 02! 0E! 00! ! ! 1F! ! !
013680 ;! ! 3E! 57! 58! 43! 56! 42! 4E! 4D! 3B! 3A! 5F! ! !
013690 ;! ! 3C! 77! 78! 63! 76! 62! 6E! 6D! 2C! 2E! 2D! ! !
013700 ;-----
013710 ; ! ! ! A05 ! !
013720 ; ! ! ! 30 ! !
013730 ; ! ! ! 60 ! !
013740 ; ! ! ! ! !
013750 ; ! ! ! 20 ! !
013760 ; ! ! ! 20 ! !
013770 ; ! ! ! 20 ! !
013780 ;-----

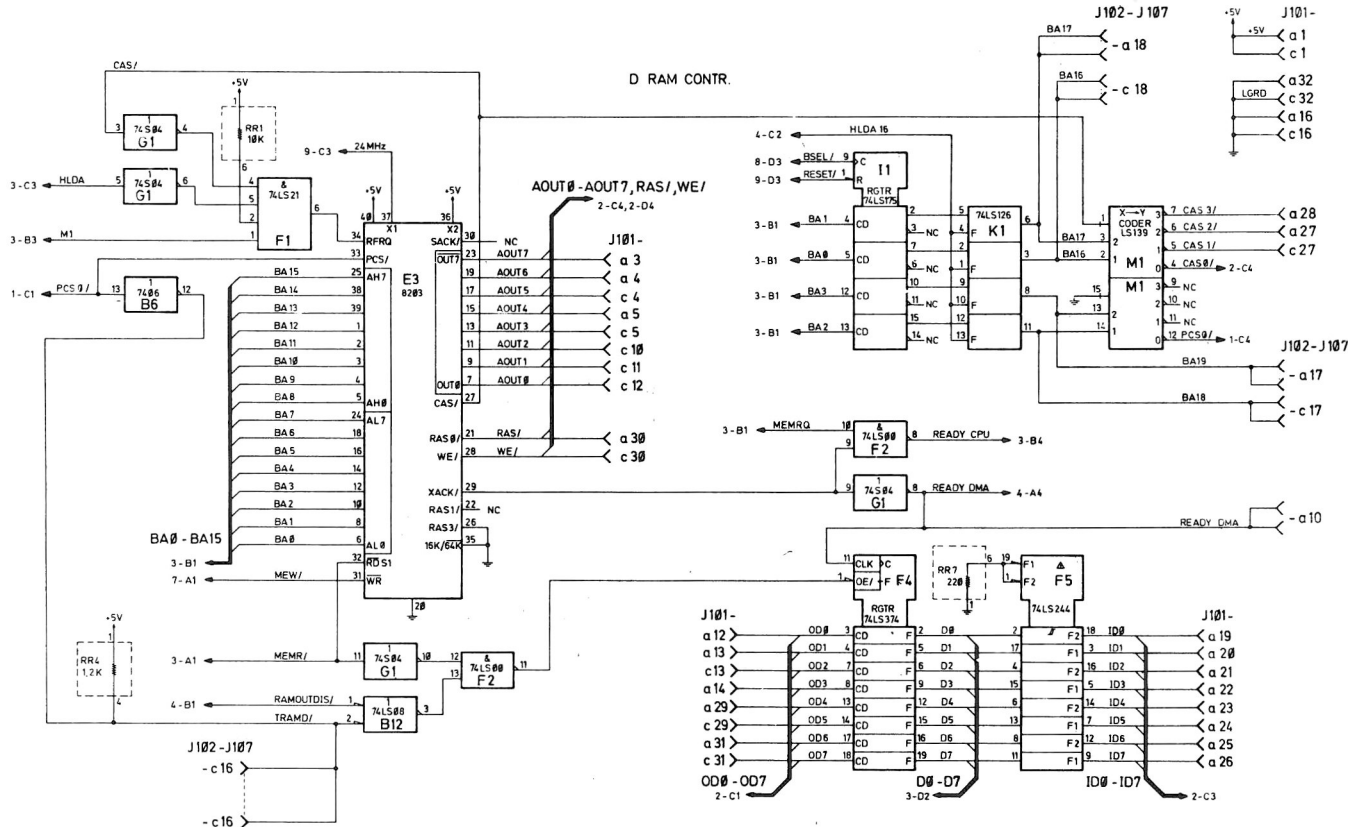
```

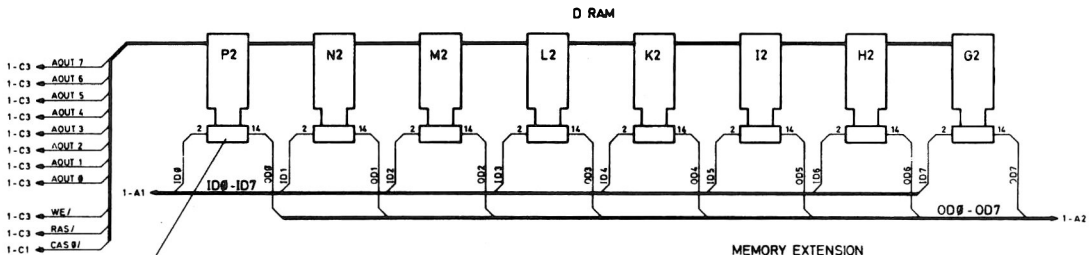

TAG-LIST

ACCFTK	LABL	0162H	005150				
ASCITB	LABL	0368H	008440	005460			
ATEMP	EQU	0007H	001110				
BDCLK	EQU	0004H	001080				
BITCTR	EQU	0003H	001070				
BITIME	LABL	0081H	002630	002520			
BUS	EQU	0000H	000000				
C	EQU	0070H	000000				
CAFLG	EQU	007EH	001160				
CHARBF	EQU	0005H	001090				
CKSU1	LABL	01F9H	006440	003460			
CKSU2	LABL	0200H	006520				
CKSU3	LABL	00E1H	003490				
CKSU4	LABL	04F9H	008750	003470			
CKSU5	LABL	0500H	008830				
CKSU6	LABL	06F9H	008940	003480			
CKSU7	LABL	0700H	009020				
CLEAR	LABL	00BFH	003160	003290			
CNT	EQU	0000H	000000				
CNTDOWN	LABL	0087H	002690	002450	002650		
COMD	LABL	0208H	006610	005720			
DBB	EQU	001AH	000000				
DECP01	EQU	0024H	001270	003800	006620		
DMJ	EQU	00E0H	000000				
DOUBLO	EQU	000BH	001200	005650			
ERRCNT	EQU	0002H	001060				
EXCEPT	LABL	01C6H	005990	005350			
EXCTB	LABL	0400H	008590	006050			
EXINT	LABL	0064H	002370	001740	001840	002160	002590
FG	EQU	005EH	000000				
F1	EQU	007EH	000000	001160			
FIFO	EQU	0010H	001250	002770	002780	003340	003350
FIFOEN	EQU	0017H	001260				
FLAGB	EQU	0021H	001220	005180	006380		
FLAGS	EQU	00F0H	000000				
FUKY	LABL	01C0H	005930	005330			
I	EQU	0000H	000000				
INTRPT	LABL	0029H	001820	001400			
INTRUP	LABL	0003H	001370				
KEYS	EQU	0068H	008430	004520			
KYDOWN	LABL	0145H	004930	004830			
KYDWN	EQU	0007H	001000				
LASTCD	EQU	0022H	001230	005270			
LASTKY	EQU	0023H	001240	005230	005860		
LDBQ	LABL	008AH	002730	005620	005670		
LNQOFS	LABL	0262H	007570	007410			
LOCKY	LABL	013FH	004880	005050	005140	006420	
LONGDL	EQU	0060H	001170	005160			
MBO	EQU	0020H	000000				
MB1	EQU	0030H	000000				
MRKTBL	LABL	0300H	008290	008430			
MTKBQ	LABL	0042H	002050	001760	001900		
NO	LABL	0186H	005420				
N1	LABL	0199H	005570				
N3	LABL	019EH	005610	003040	006170	006760	008190

CONTROLLER BOARD



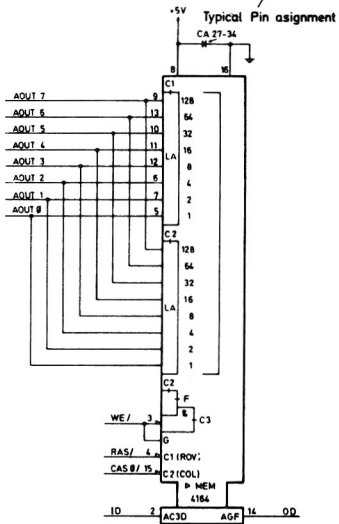


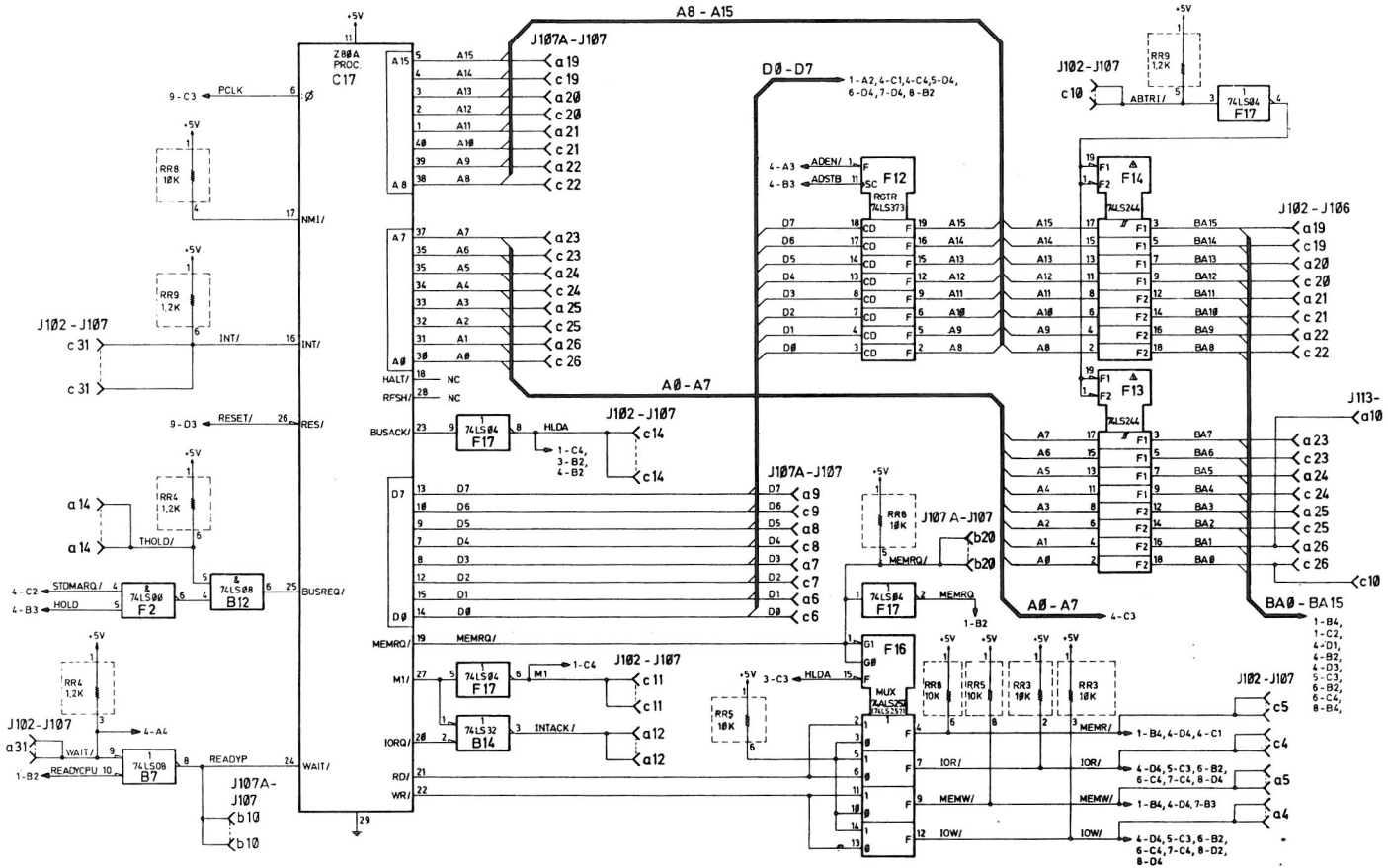


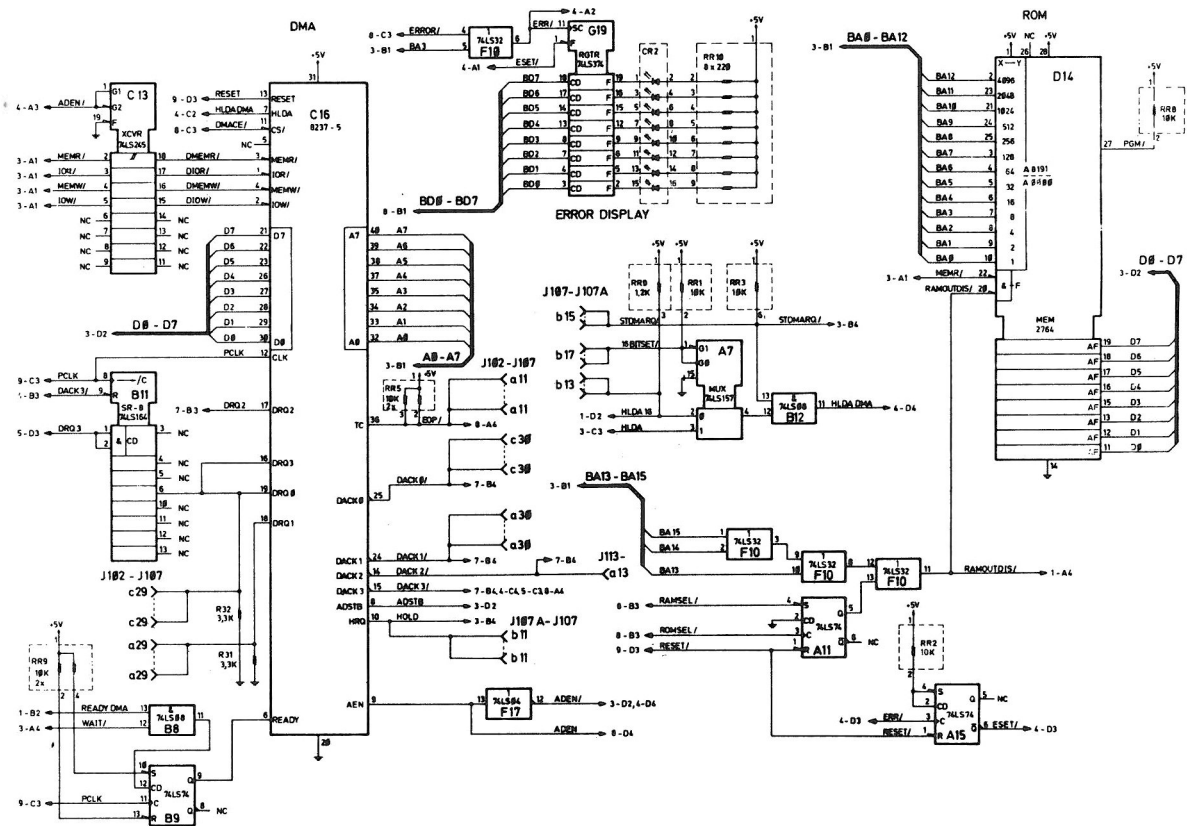
MEMORY EXTENSION

J101

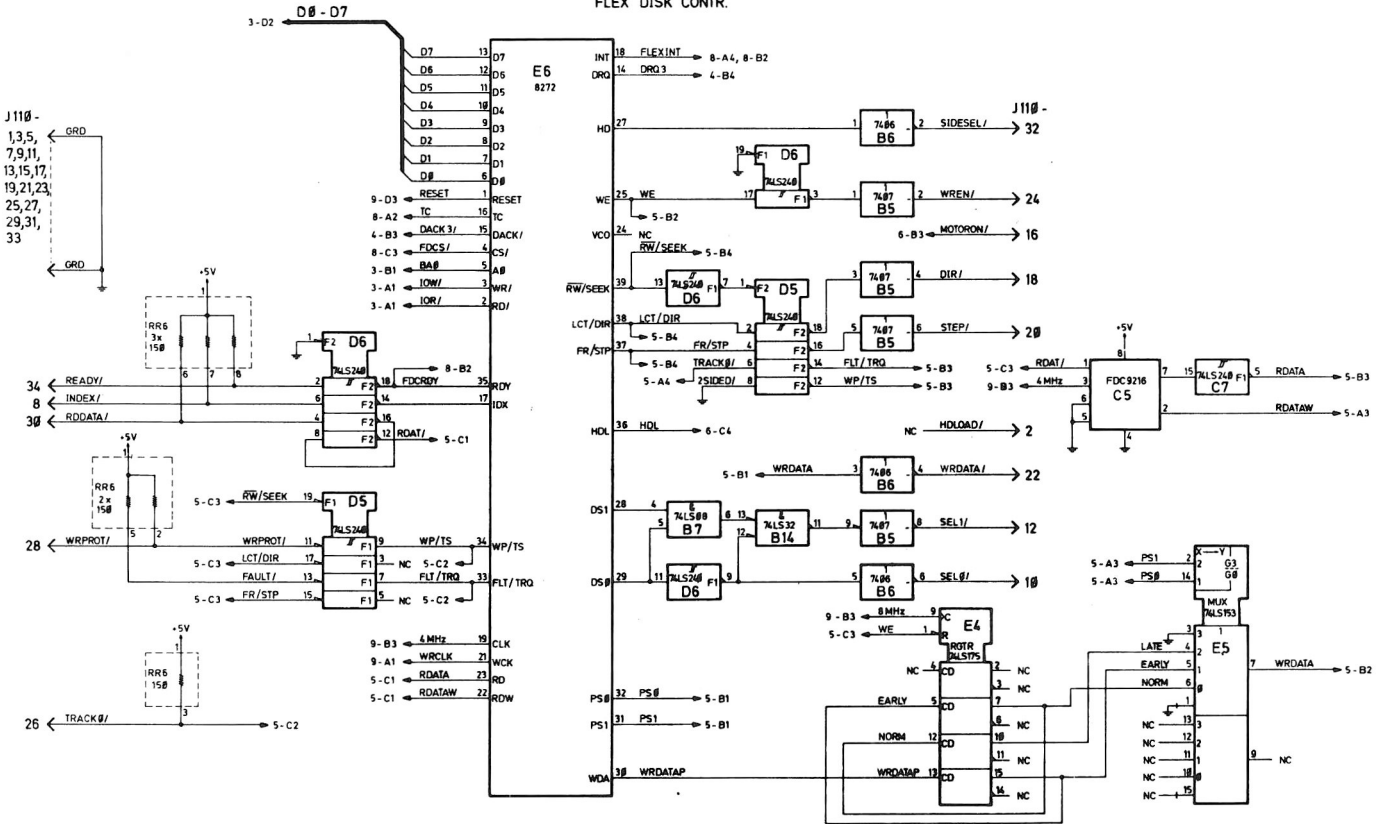
0		c
1	-5V	1 -5V
2	NC	2 NC
3	AOUT 7	3 NC
4	AOUT 6	4 AOUT 5
5	AOUT 4	5 AOUT 3
6	NC	6 NC
7	NC	7 NC
8	NC	8 NC
9	NC	9 NC
10	NC	10 AOUT 2
11	NC	11 AOUT 1
12	OD0	12 AOUT 0
13	OD1	13 OD2
14	OD3	14 NC
15	NC	15 NC
16	LGRD	16 LGRD
17	NC	17 NC
18	NC	18 NC
19	ID0	19 NC
20	ID1	20 NC
21	ID2	21 NC
22	ID3	22 NC
23	ID4	23 NC
24	ID5	24 NC
25	ID6	25 NC
26	ID7	26 NC
27	CAS 2/	27 CAS 1/
28	CAS 3/	28 NC
29	OD4	29 OD5
30	RAS/	30 WE/
31	OD6	31 OD7
32	LGRD	32 LGRD

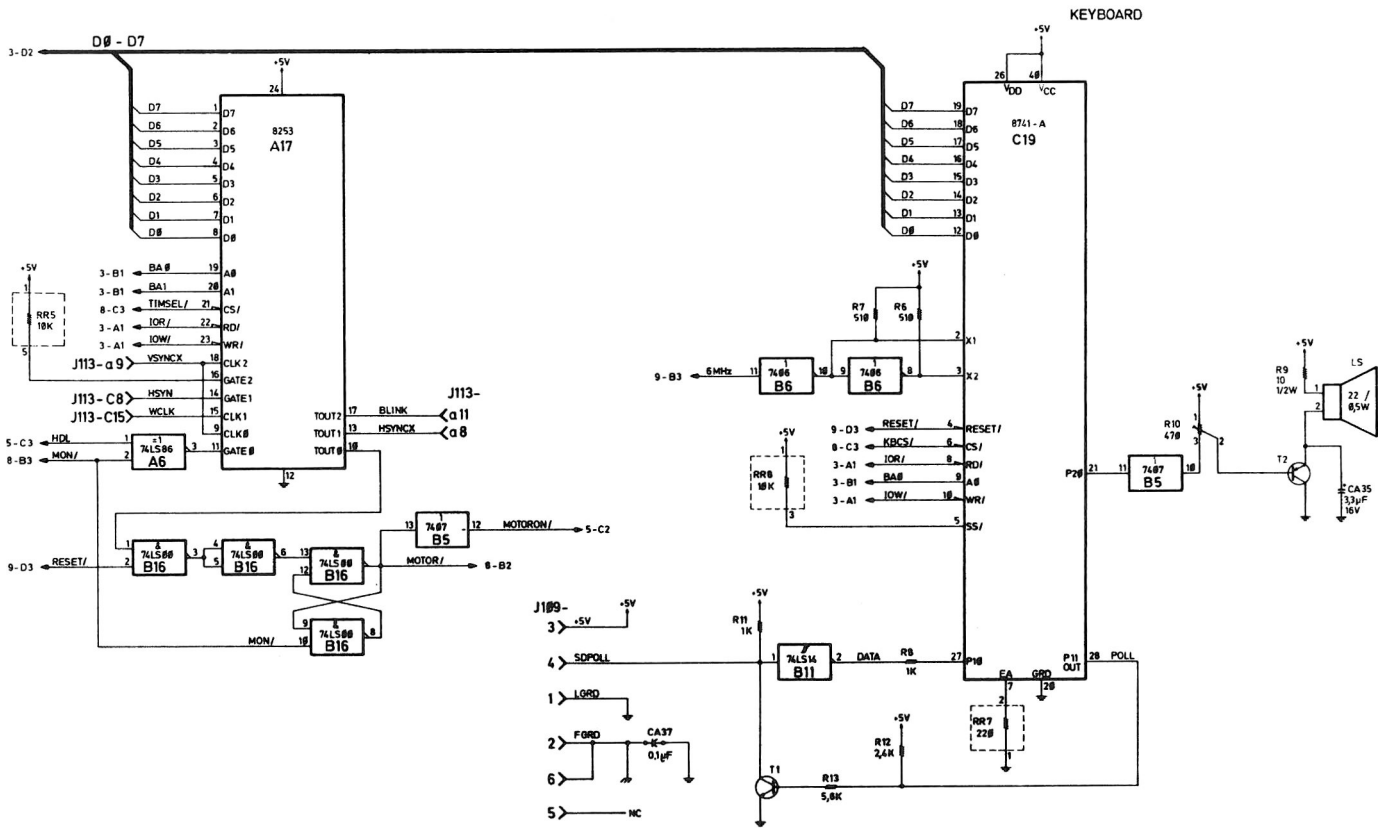


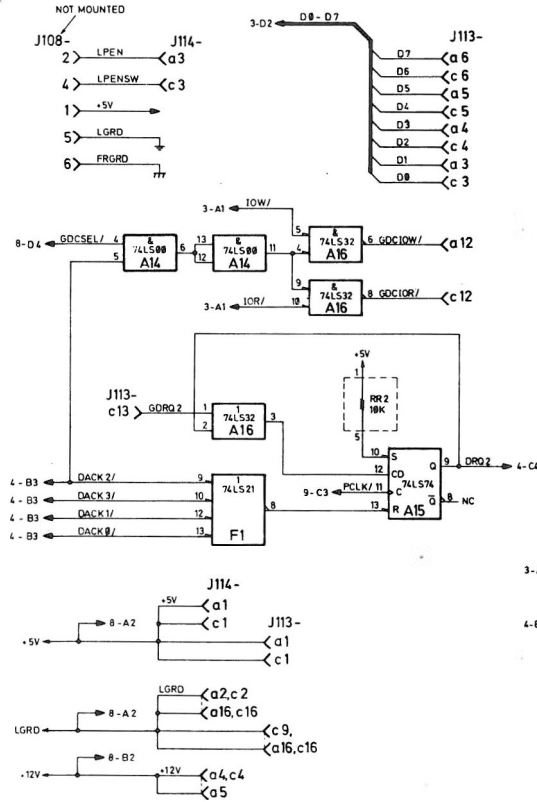




FLEX DISK CONTR.







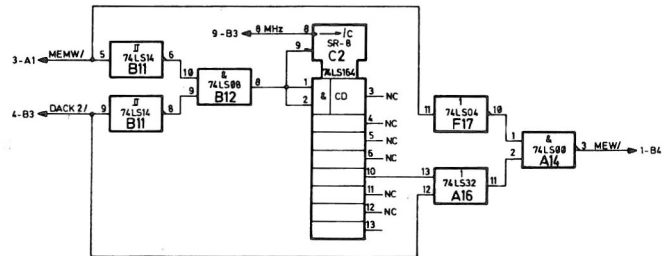
TO GRAPHIC BOARD

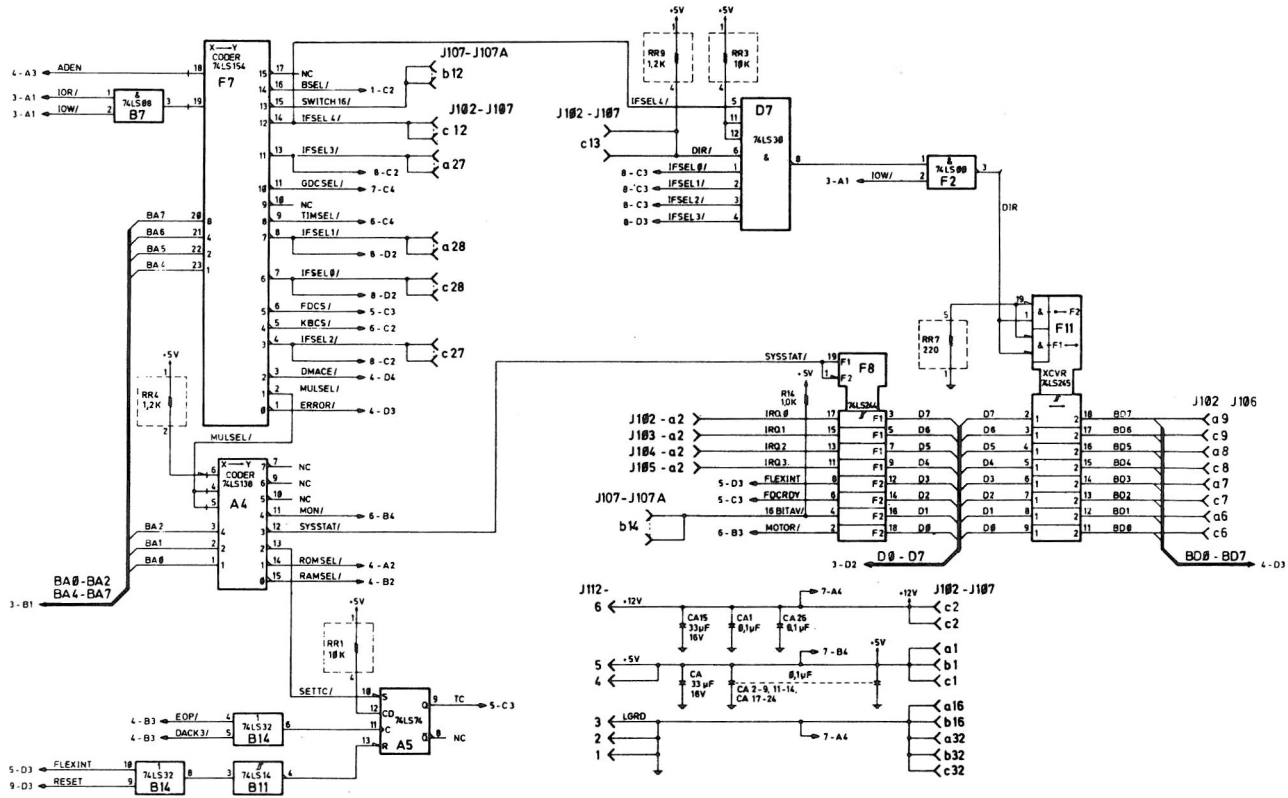
J114

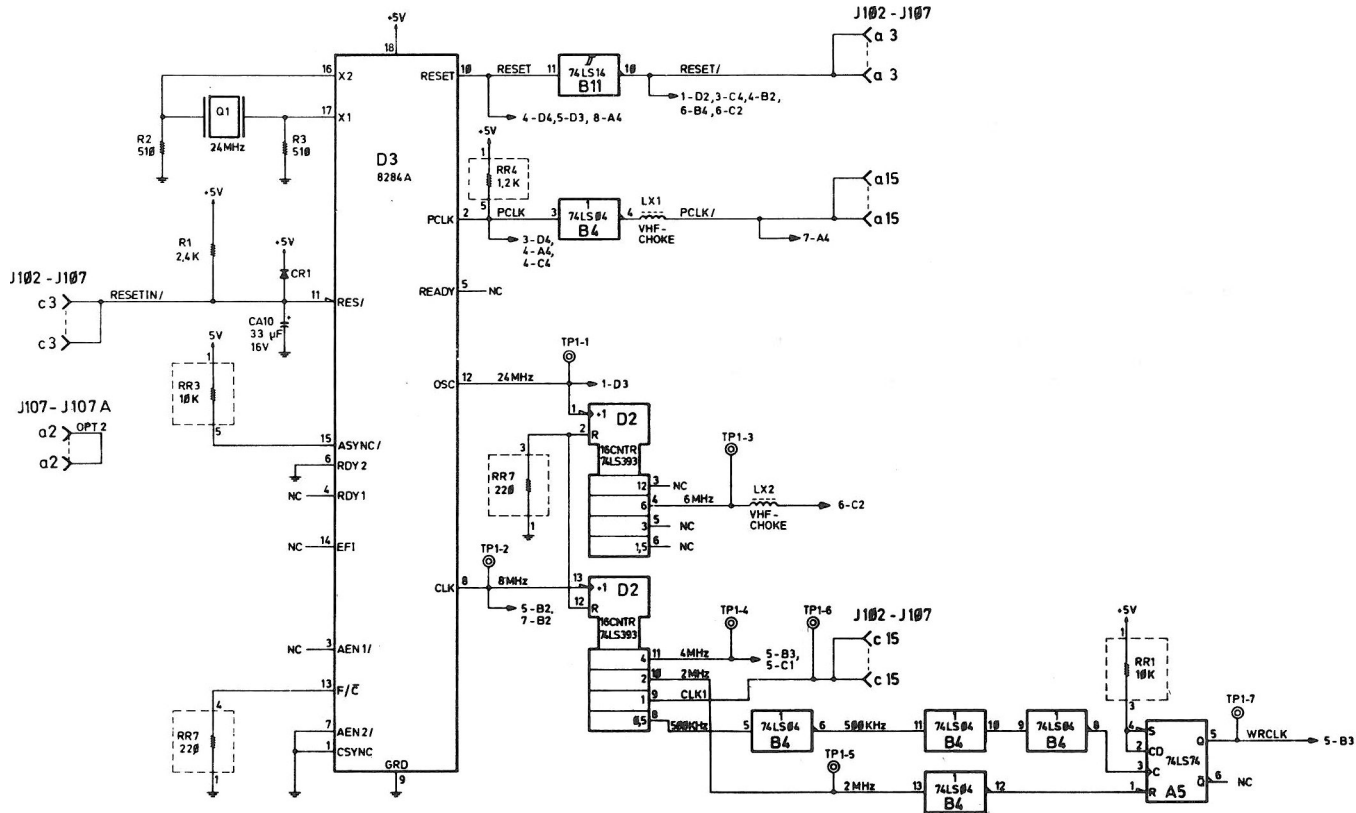
a	b	c
+5V	1	+5V
LG RD	2	LG RD
LP EN	3	LP EN SW
+12V	4	+12V
-12V	5	NC
NC	6	NC
NC	7	NC
NC	8	NC
NC	9	NC
NC	10	NC
NC	11	NC
NC	12	NC
NC	13	NC
NC	14	NC
NC	15	LG RD
LG RD	16	LG RD

J113

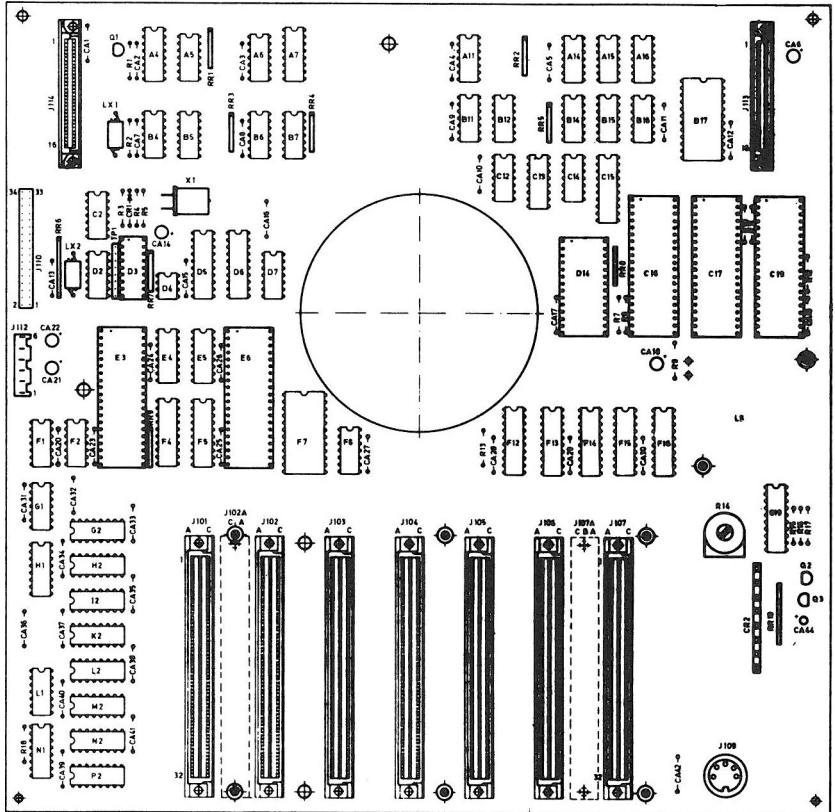
a	b	c
+5V	1	+5V
NC	2	NC
D1	3	D#
D3	4	D2
D5	5	D4
D7	6	D5
NC	7	NC
H SYNC	8	H SYN
V SYNC	9	LG RD
NC	10	BA#
BLNK	11	NC
G DCK 0/	12	G DCK 0/
D ACK 2/	13	G DRQ 2
NC	14	NC
NC	15	W ELK
LG RD	16	LG RD



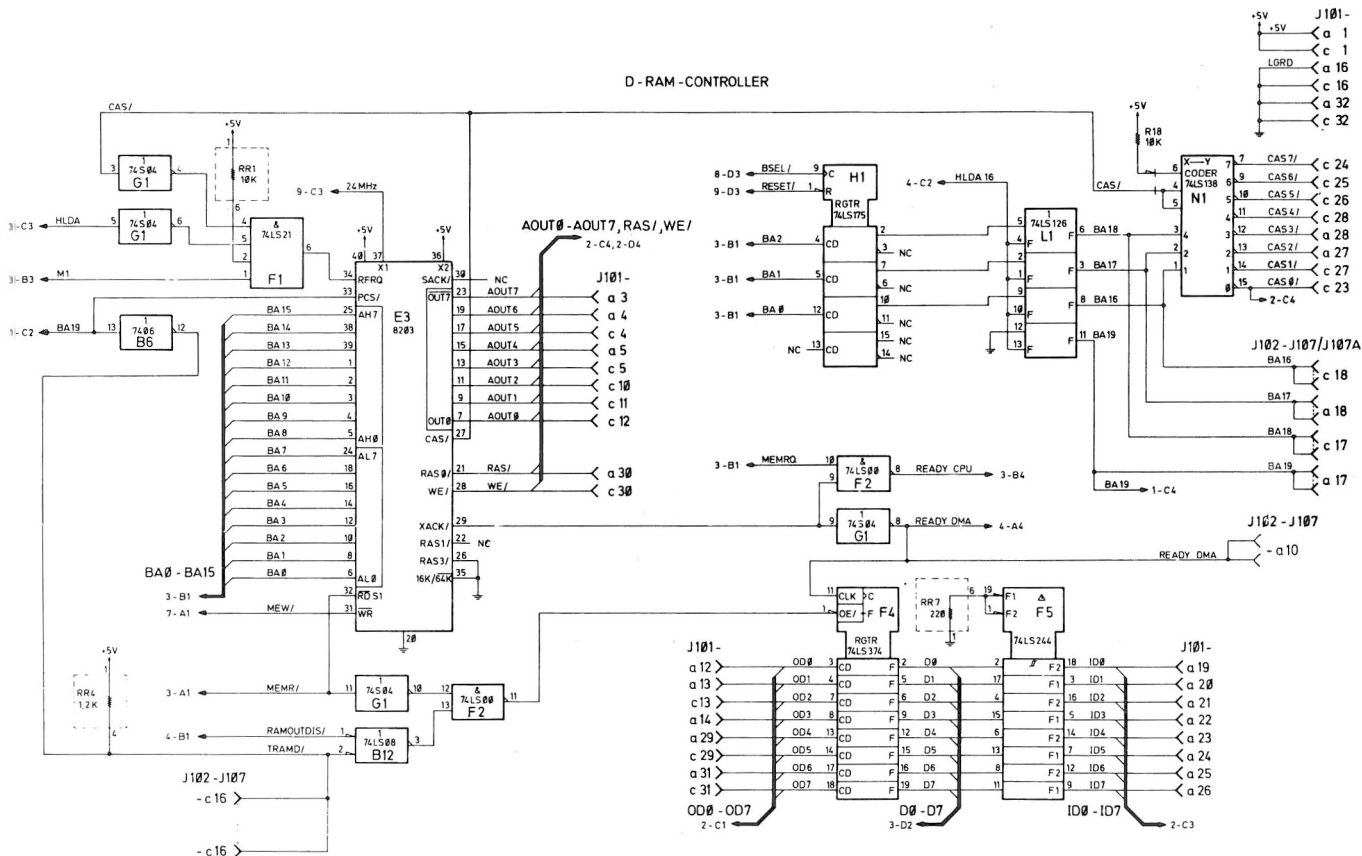


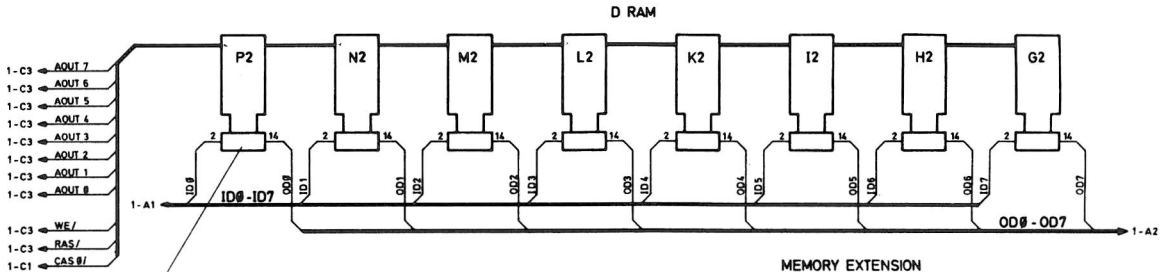


CONTROLLER BOARD (MULTI-LAYER)



D - RAM - CONTROLLER

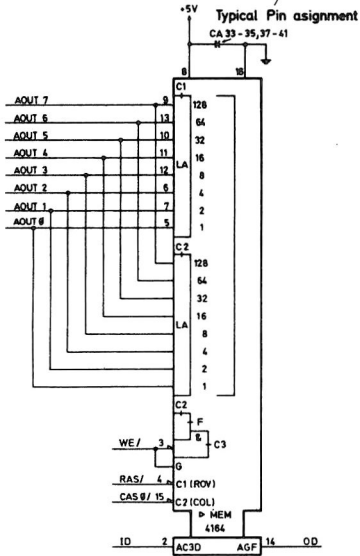


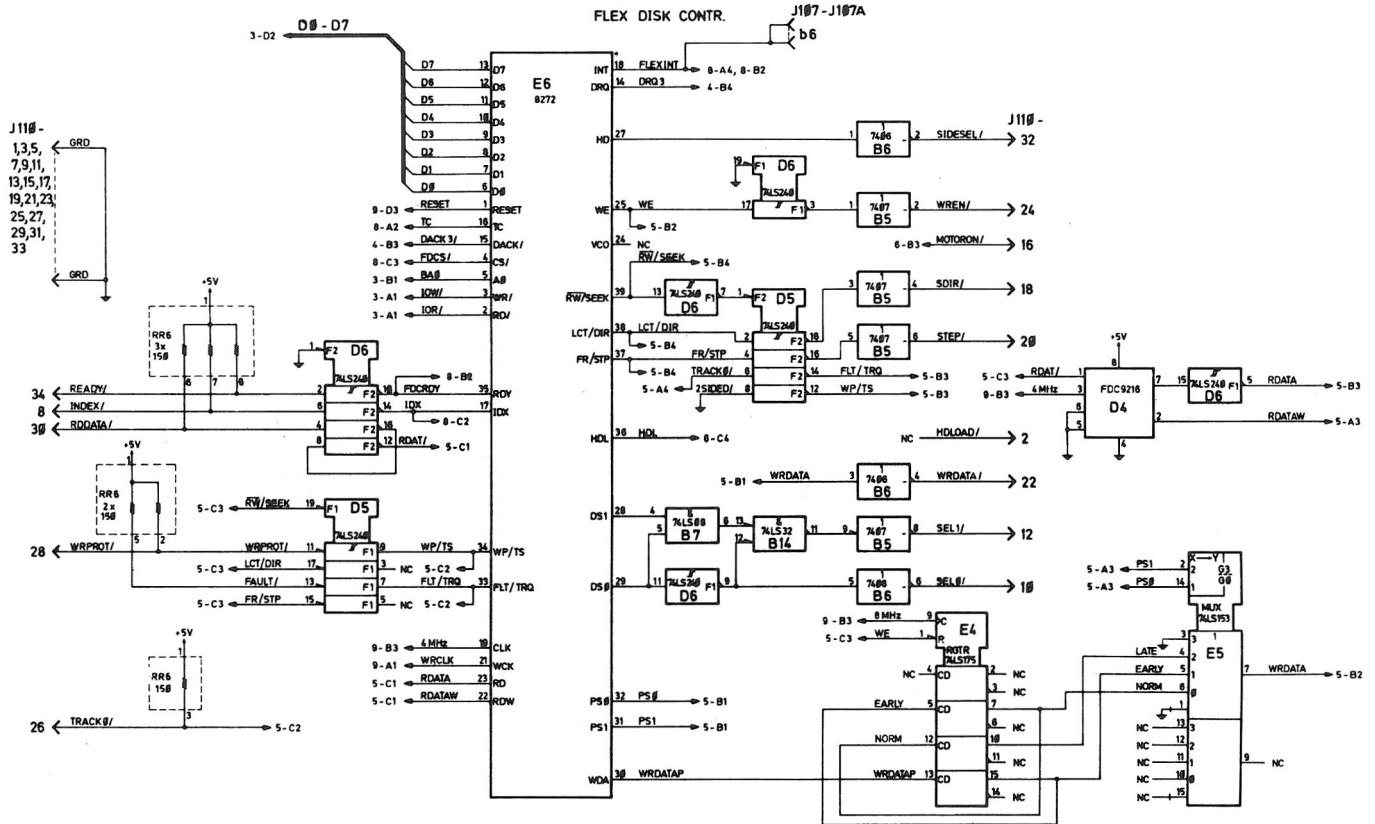


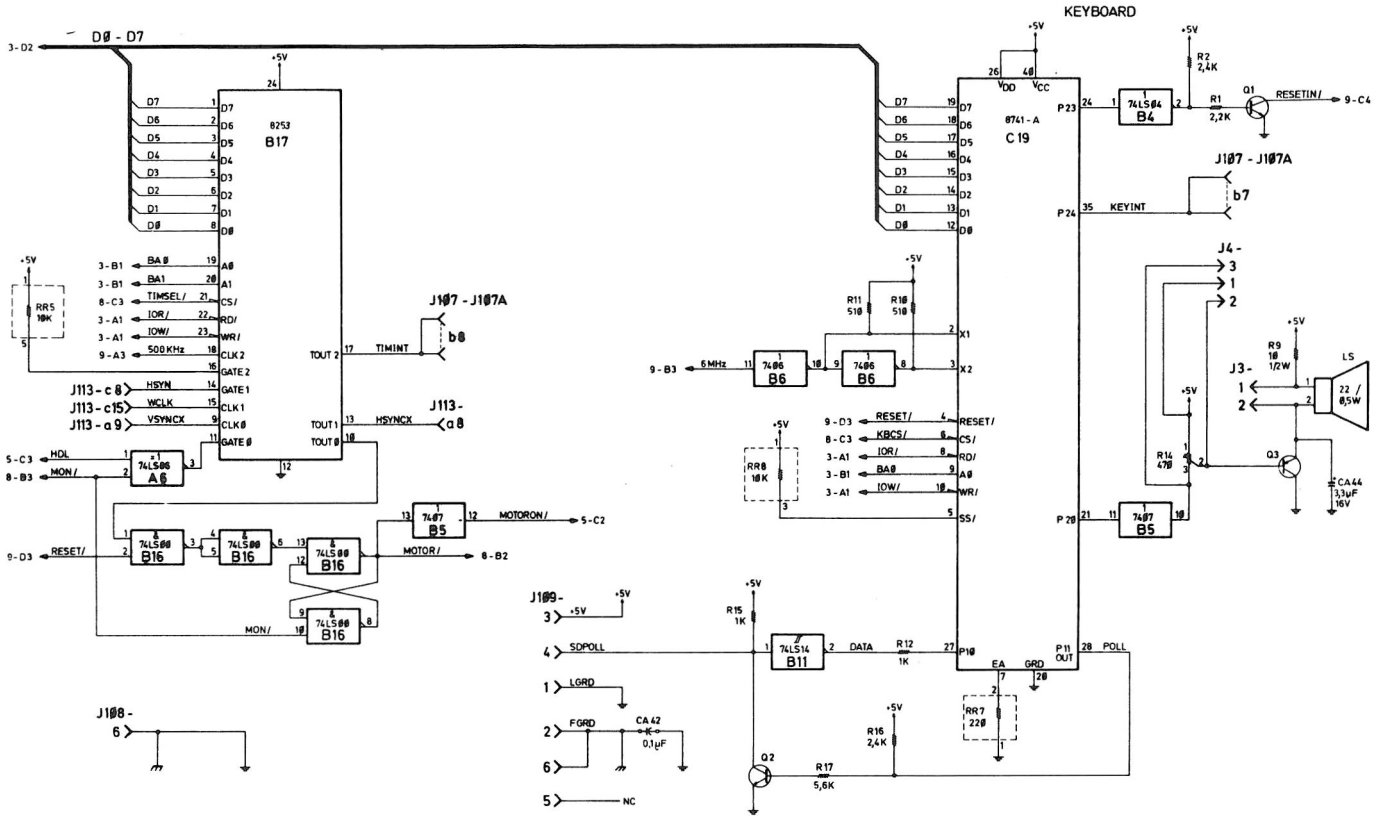
MEMORY EXTENSION

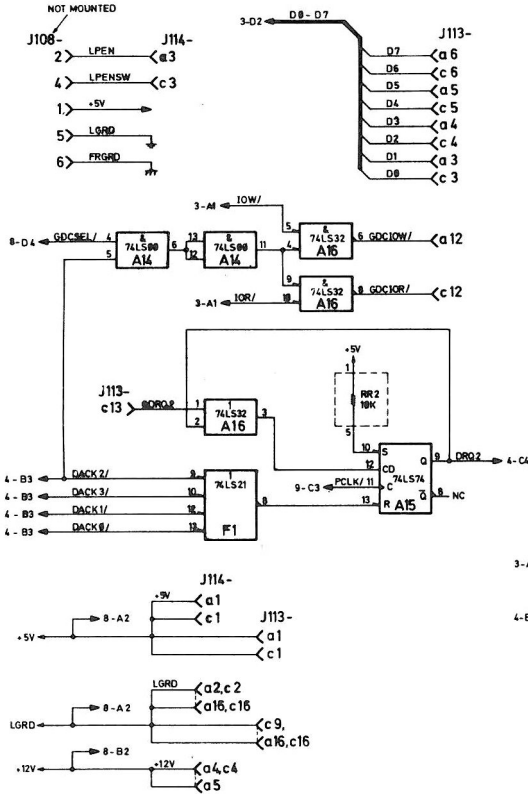
J101

a	1	6SV	1	6SV
	2	NC	2	NC
	3	AOUT 7	3	NC
	4	AOUT 6	4	AOUT 5
	5	AOUT 4	5	AOUT 3
	6	NC	6	NC
	7	NC	7	NC
	8	NC	8	NC
	9	NC	9	NC
	10	NC	10	AOUT 2
	11	NC	11	AOUT 1
	12	OD 0	12	AOUT 0
	13	OD 1	13	OD 2
	14	OD 3	14	NC
	15	NC	15	NC
	16	LGRD	16	LGRD
	17	NC	17	NC
	18	NC	18	NC
	19	ID 0	19	NC
	20	ID 1	20	NC
	21	ID 2	21	NC
	22	ID 3	22	NC
	23	ID 4	23	CAS #/
	24	ID 5	24	CAS 7/
	25	ID 6	25	CAS 8/
	26	ID 7	26	CAS 5/
	27	CAS 3/	27	CAS 1/
	28	CAS 3/	28	CAS 4/
	29	OD 4	29	OD 5
	30	RAS /	30	WE /
	31	OD 6	31	OD 7
	32	LGRD	32	LGRD









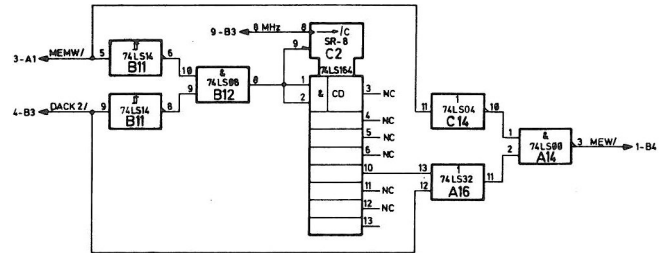
TO GRAPHIC BOARD

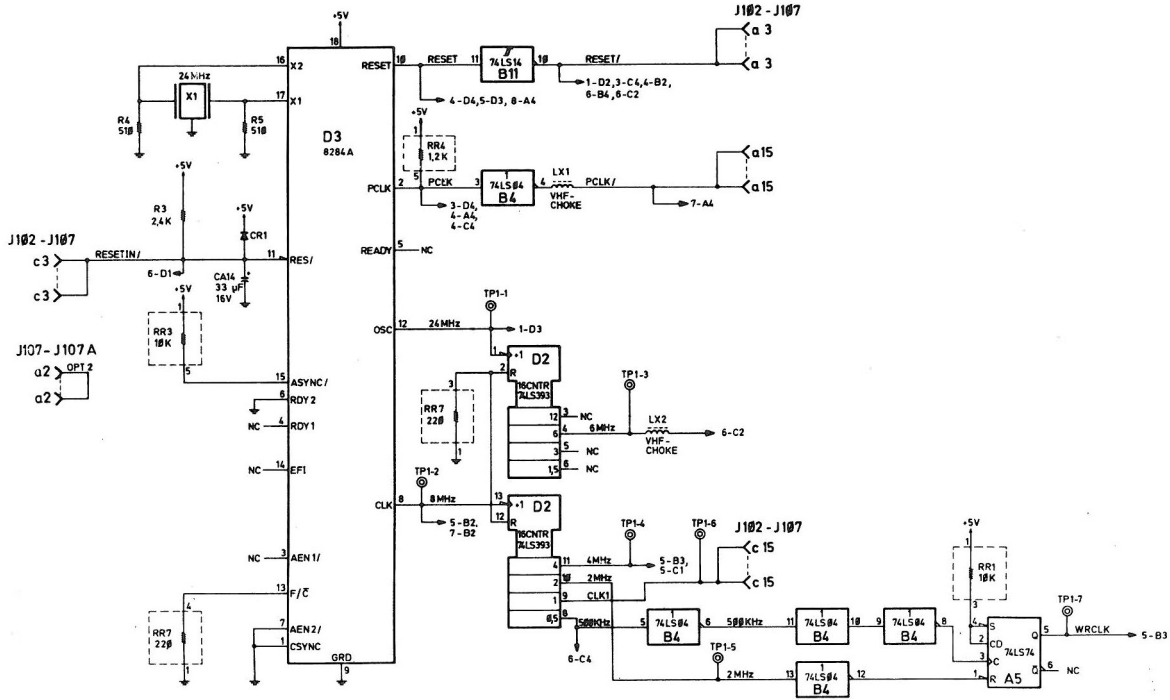
J114

a	1	c
+5V	1	+5V
LG RD	2	LG RD
LP EN	3	LP EN SW
+12V	4	+12V
-12V	5	NC
NC	6	NC
NC	7	NC
NC	8	NC
NC	9	NC
NC	10	NC
NC	11	NC
NC	12	NC
NC	13	NC
NC	14	NC
NC	15	LG RD
LG RD	16	LG RD

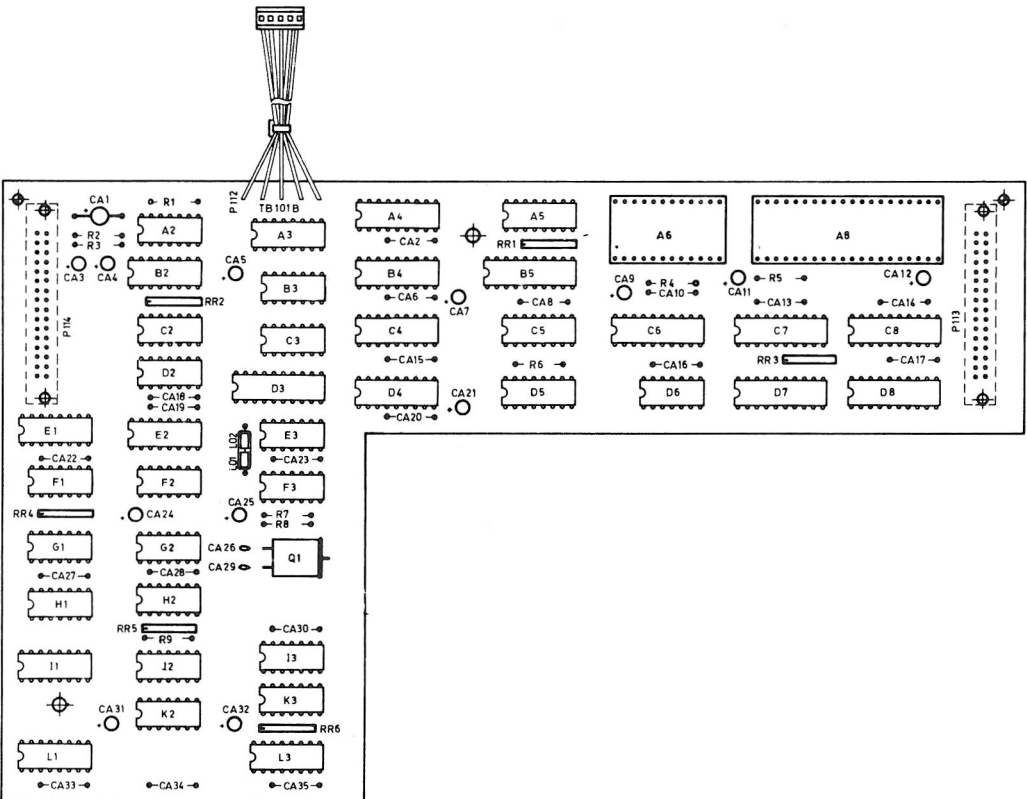
J113

a	1	c
+5V	1	+5V
NC	2	NC
D1	3	D9
D3	4	D2
D5	5	D4
D7	6	D6
NC	7	NC
H5YACK	8	H5YH
V5YACK	9	LG RD
BA1	10	BA8
BLNK	11	NC
GDCIOW/	12	GDCIOR/
DACK2/	13	DRQ2
NC	14	NC
NC	15	WCLK
LG RD	16	LG RD

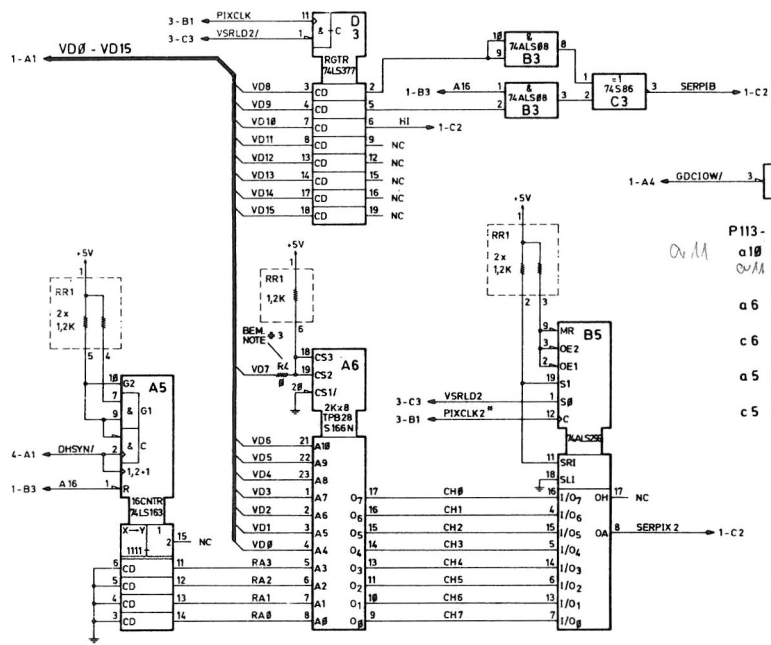




GRAPHICS CONTROLLER (MONOCHROME)



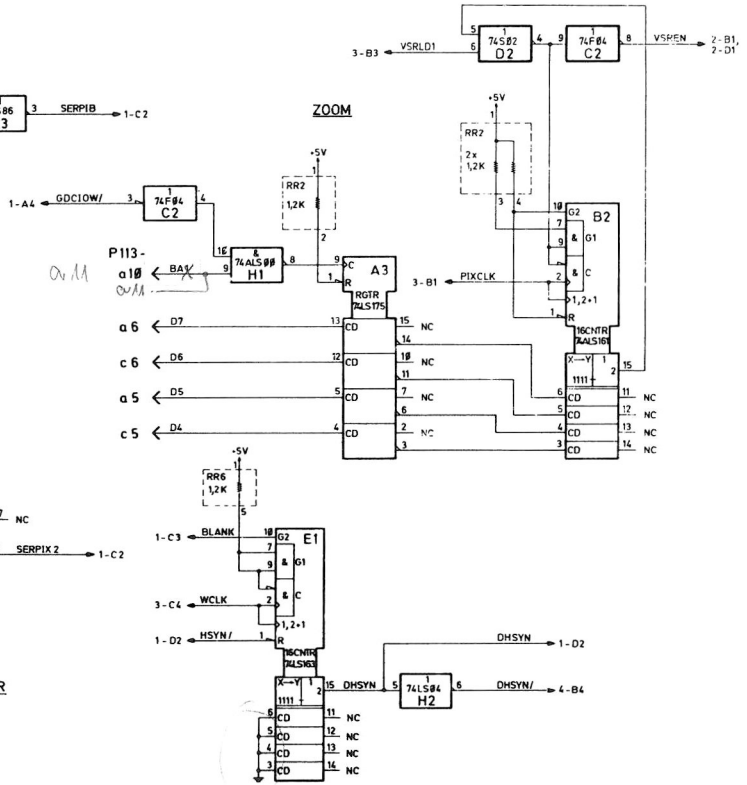
ATTRIBUT - REGISTER



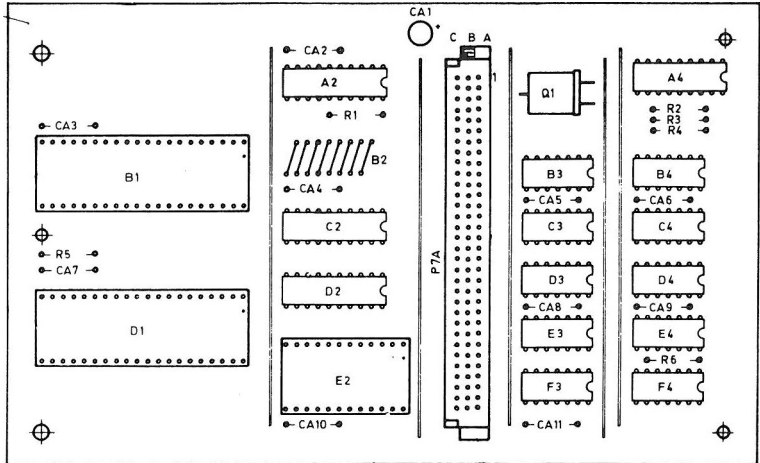
CHARACTER - GENERATOR

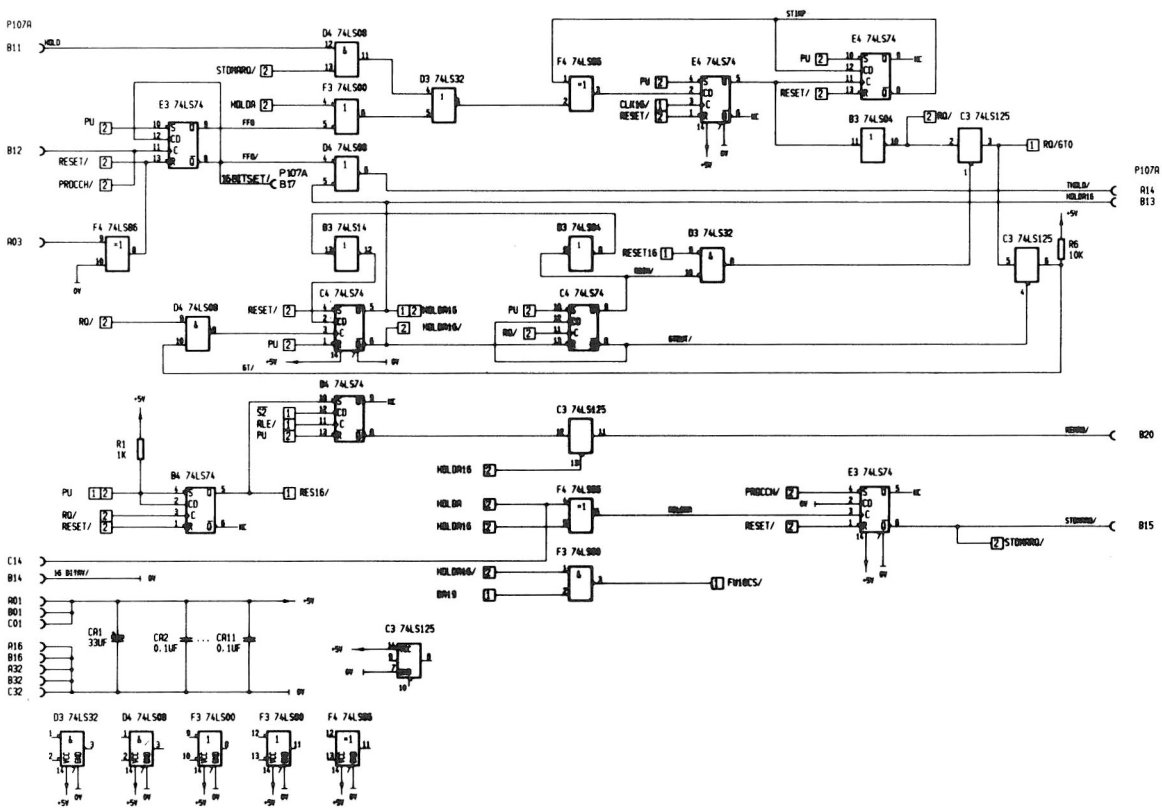
SHIFT - REGISTER

ZOOM

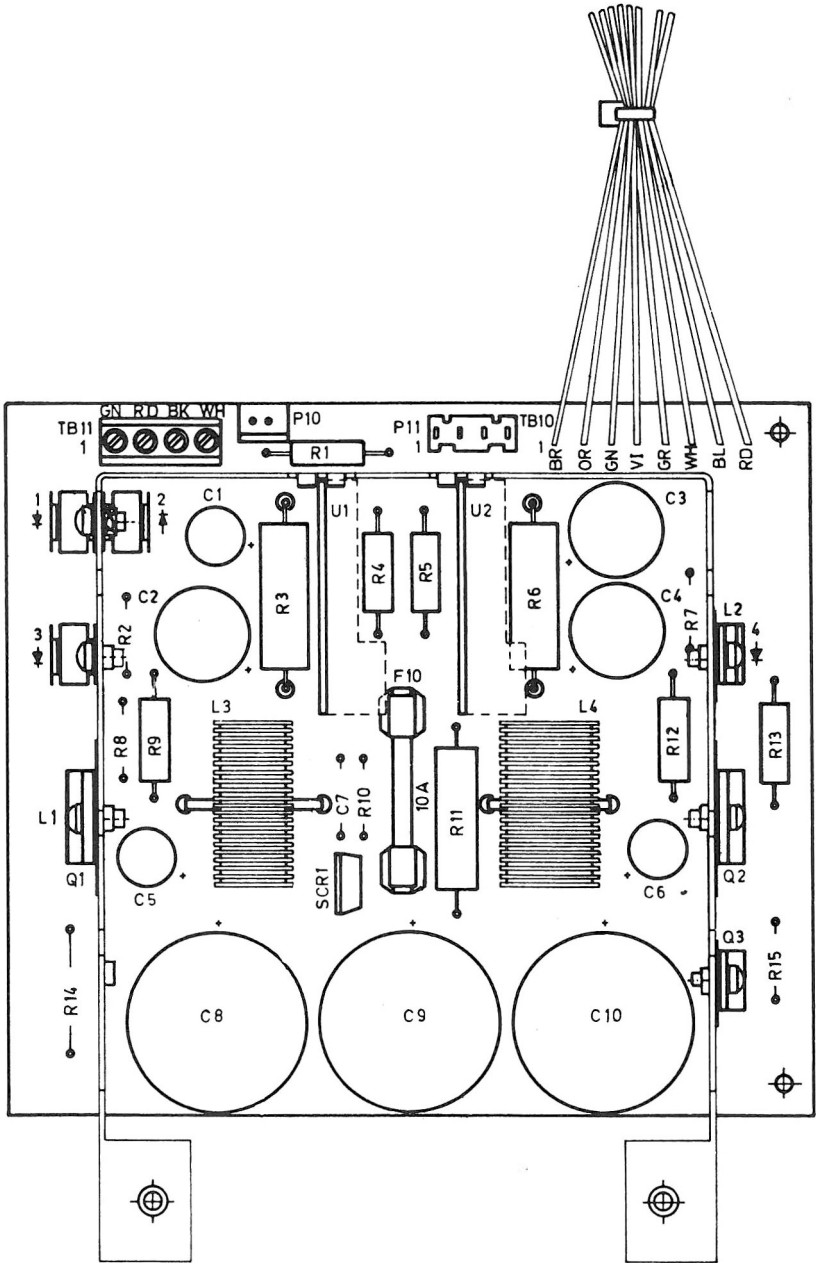


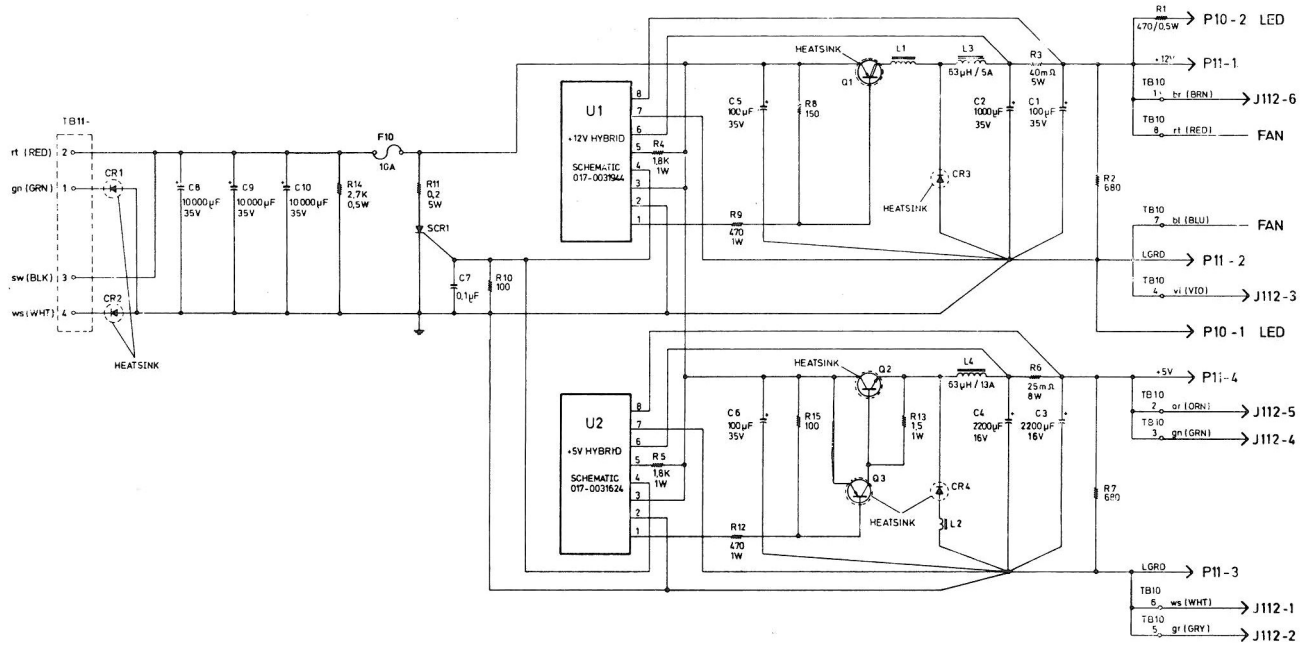
16-BIT PROCESSOR



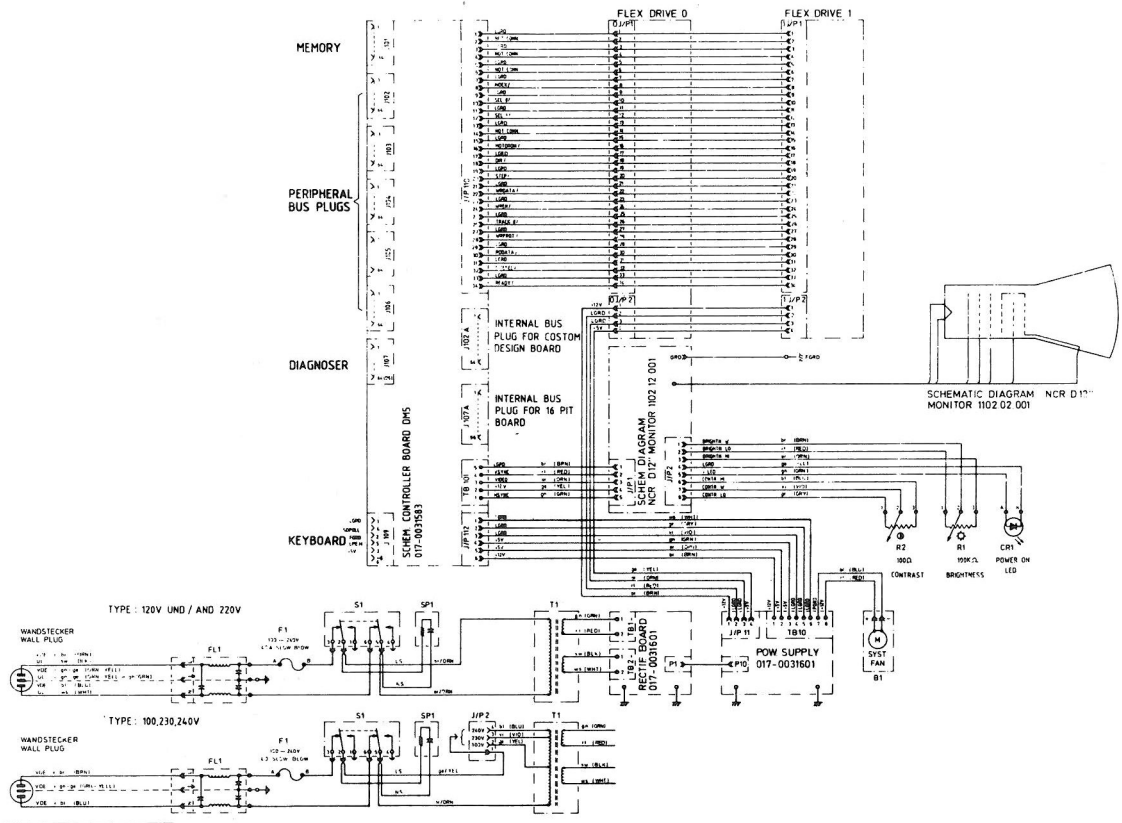


POWER SUPPLY

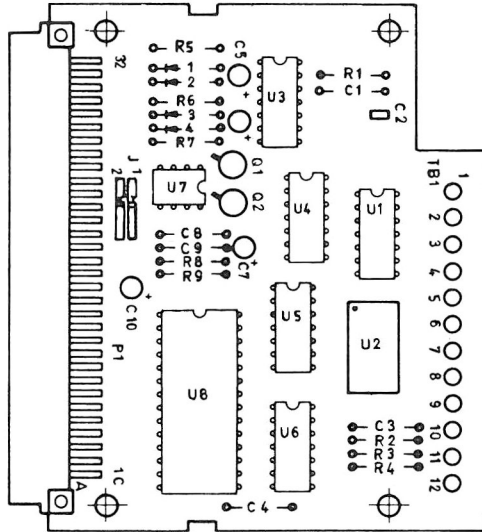


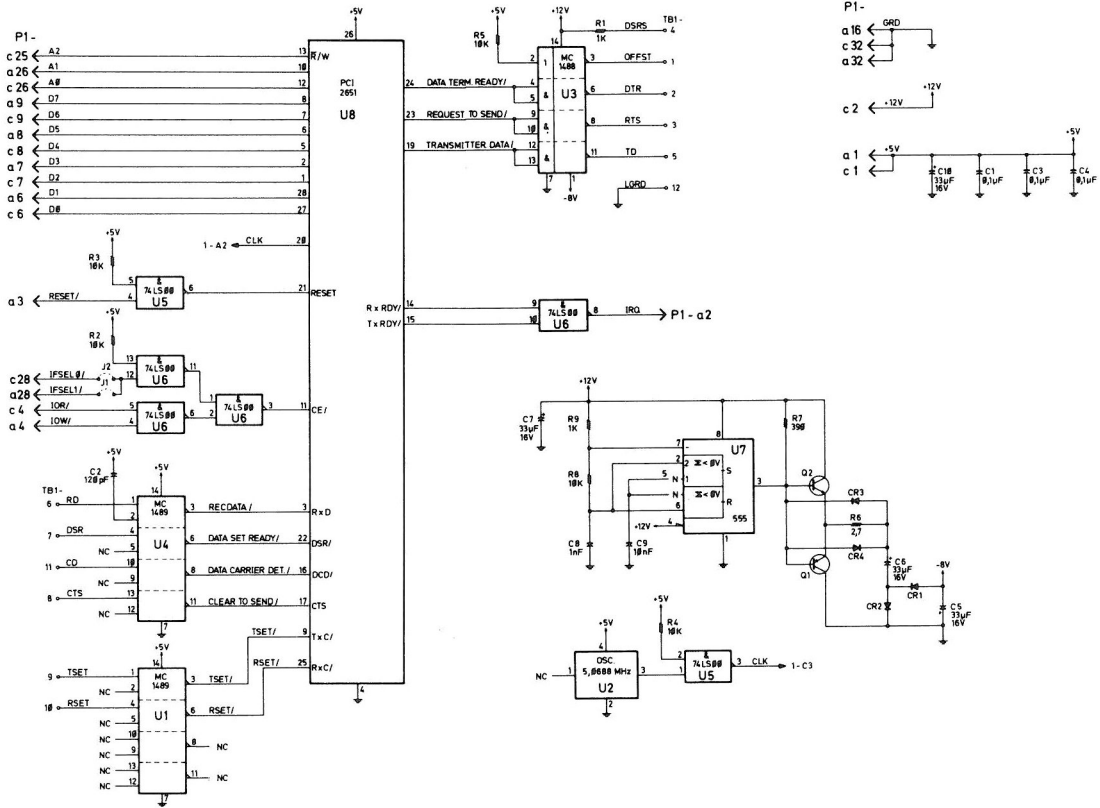


AC/DC WIRING

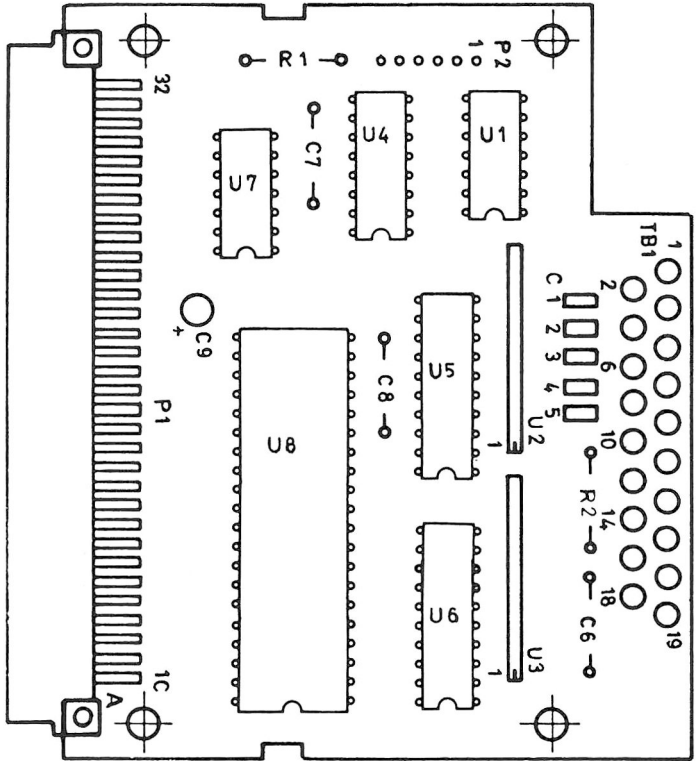


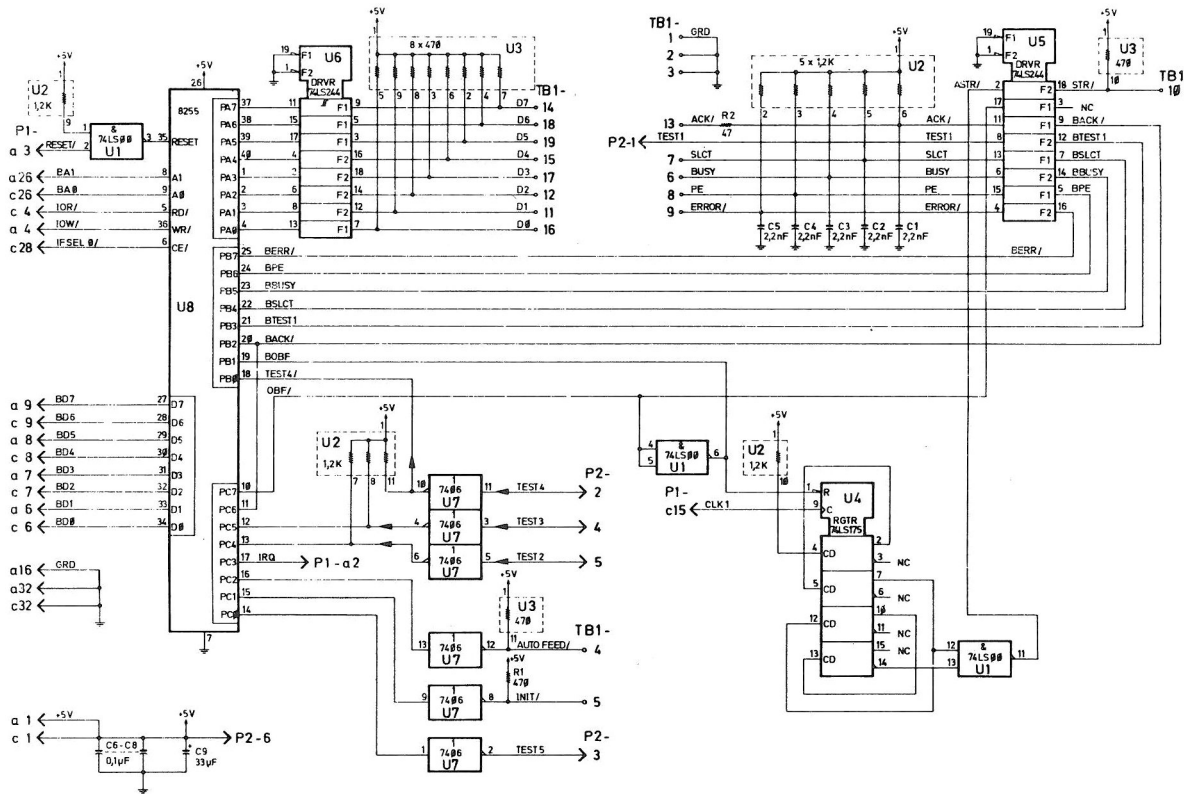
RS-232C ADAPTER



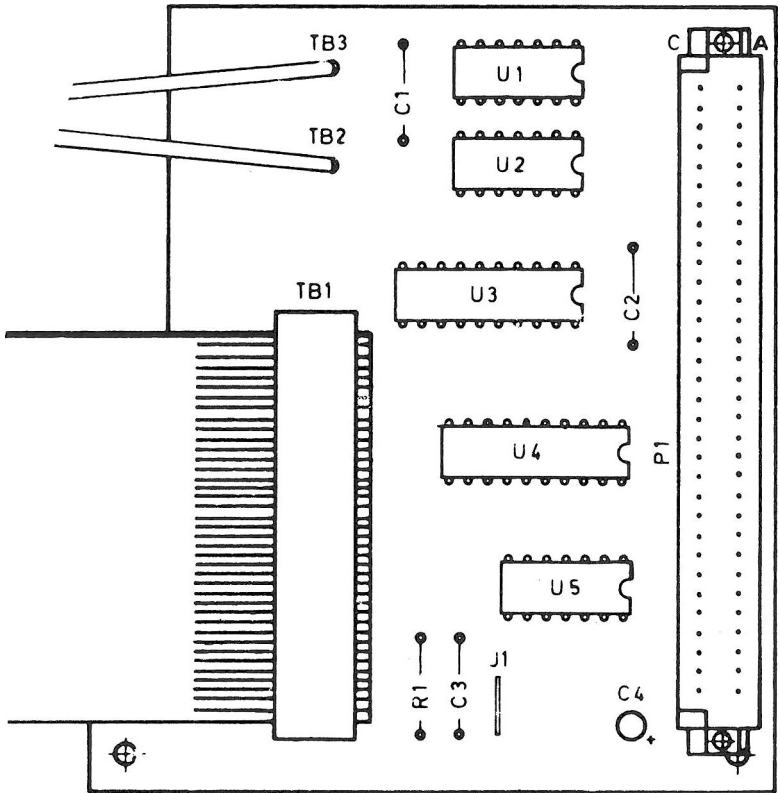


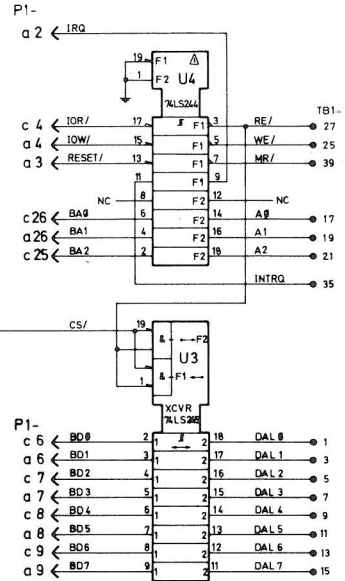
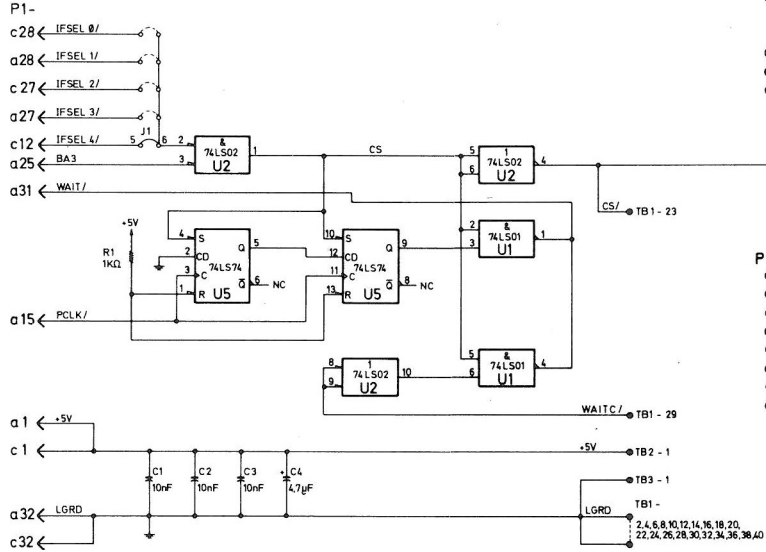
CENTRONICS ADAPTER



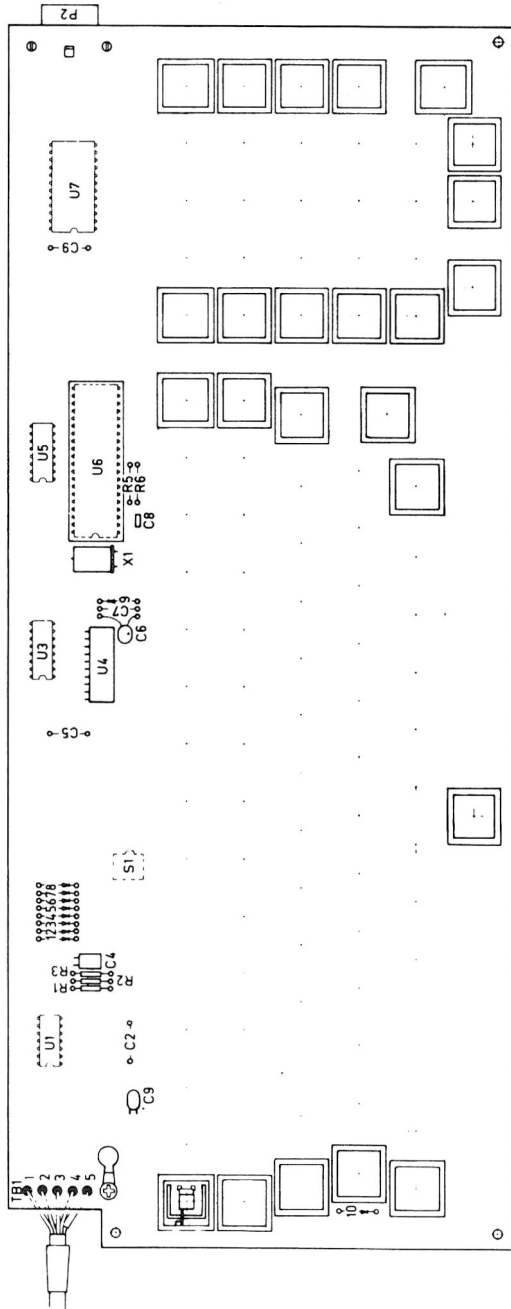


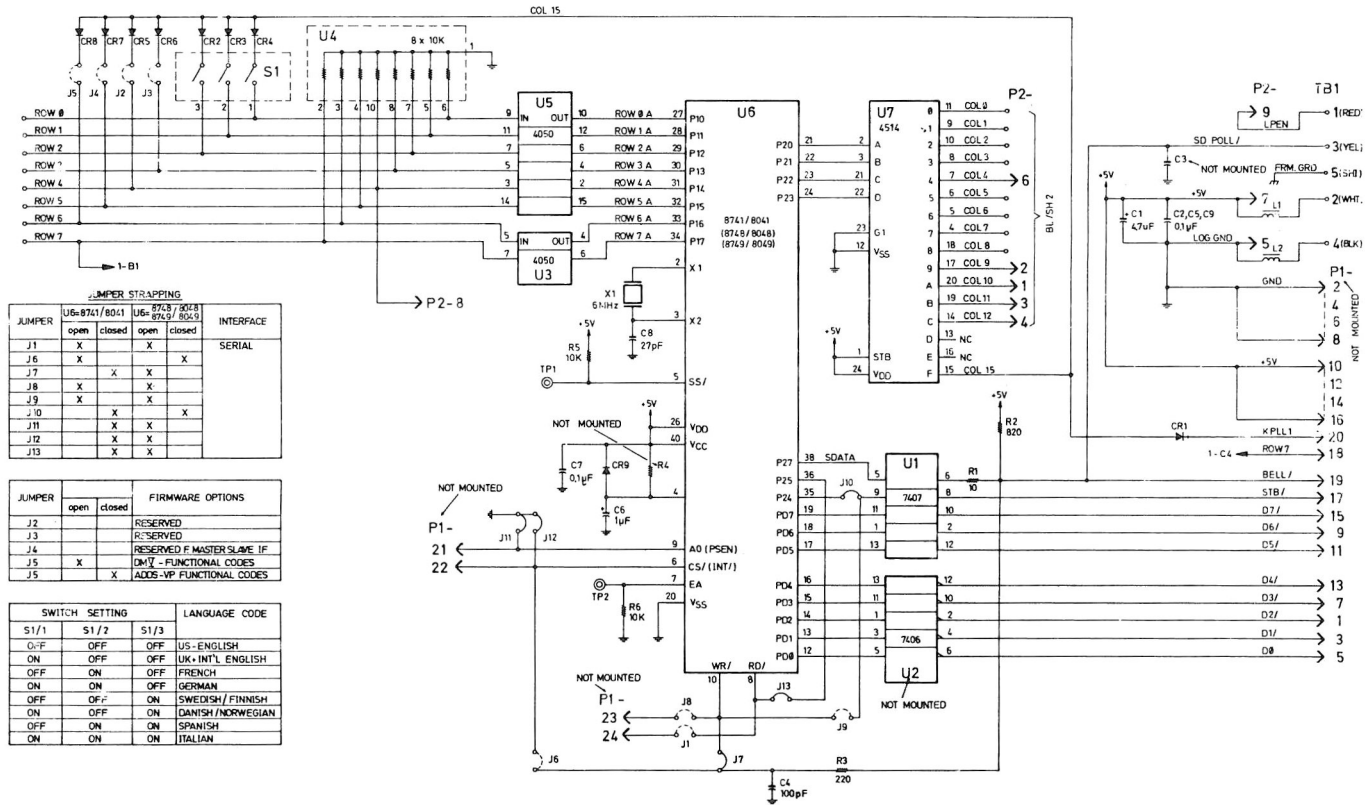
FIXED DISK INTERFACE





KEYBOARD





JUMPER STRAPPING

JUMPER	U5-8741/8041		U6-8748/8048		INTERFACE
	open	closed	open	closed	
J1	X		X		SERIAL
J6	X		X	X	
J7		X	X		
J8	X		X		
J9	X	X	X	X	
J10		X	X		
J11		X	X		
J12		X	X		
J13		X	X		

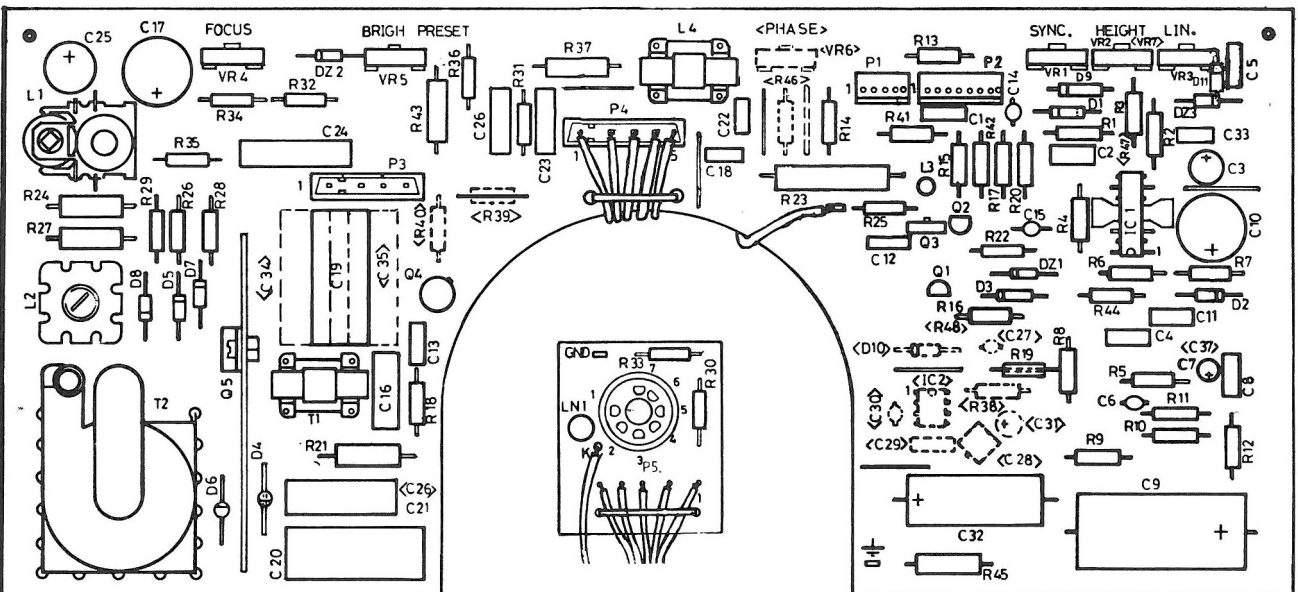
JUMPER

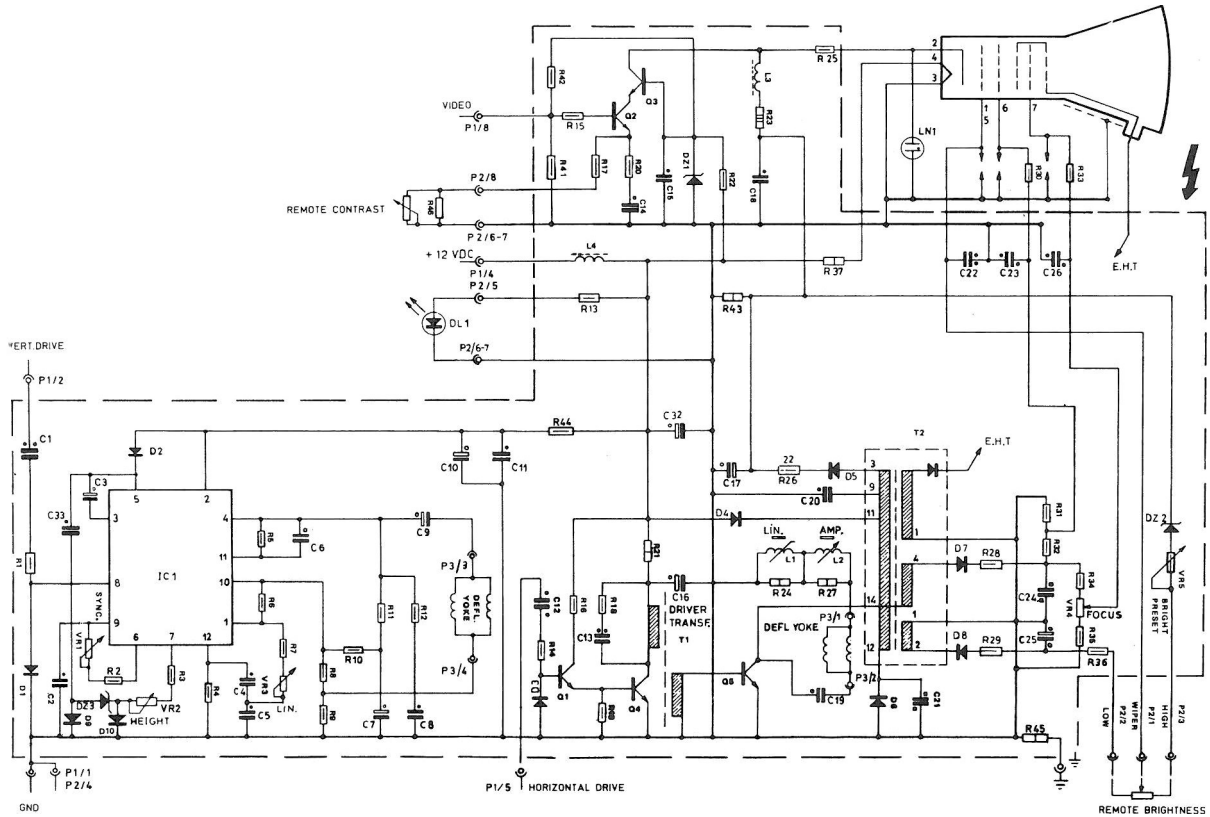
JUMPER	open	closed	FIRMWARE OPTIONS
J2			RESERVED
J3			RESERVED
J4			RESERVED F MASTER SLAVE IF
J5	X		ONLY - FUNCTIONAL CODES
J5		X	ADD5-VIP FUNCTIONAL CODES

SWITCH SETTING

S1/1	S1/2	S1/3	LANGUAGE CODE
OFF	OFF	OFF	US-ENGLISH
ON	OFF	OFF	UK+INTL ENGLISH
OFF	ON	OFF	FRENCH
ON	ON	OFF	GERMAN
OFF	OFF	ON	SWEDISH/FINNISH
ON	OFF	ON	DANISH/NORWEGIAN
OFF	ON	ON	SPANISH
ON	ON	ON	ITALIAN

MONOCHROME CRT



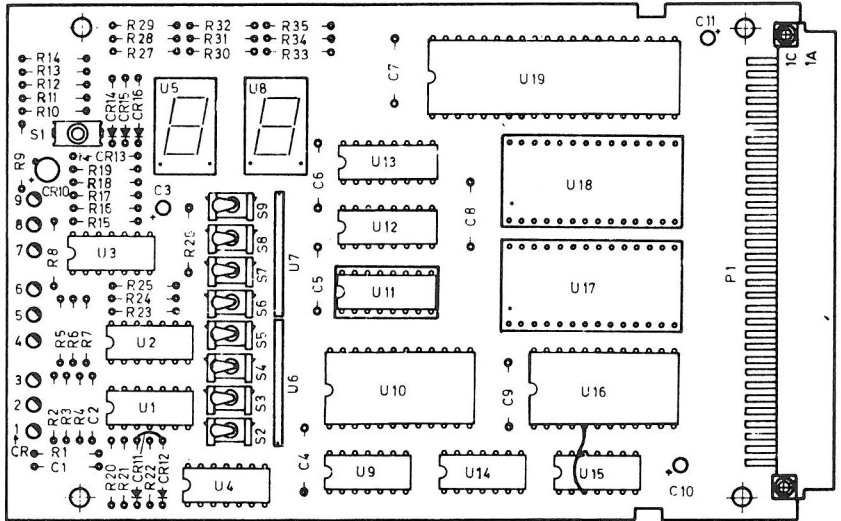


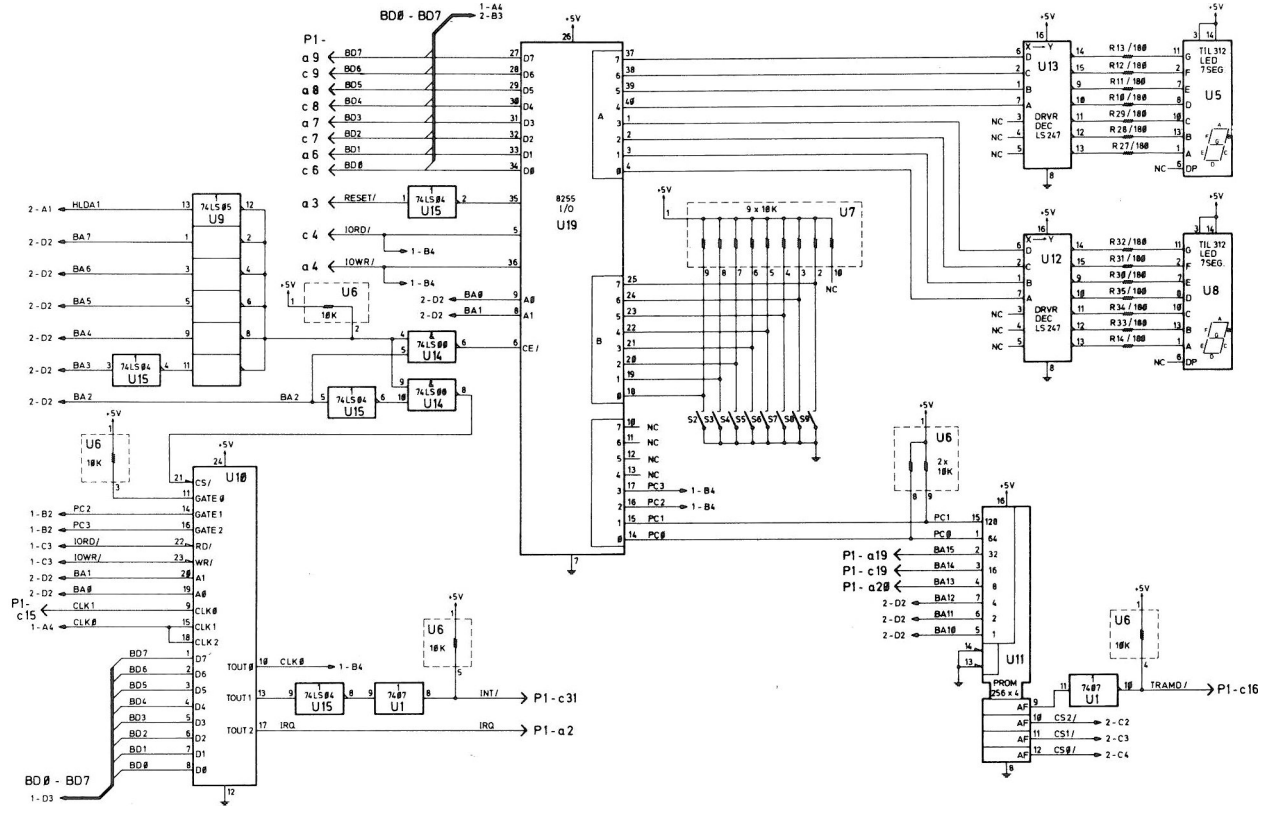
GND

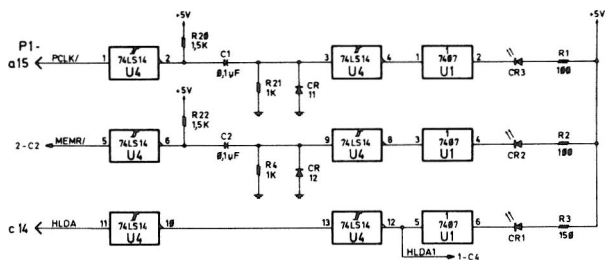
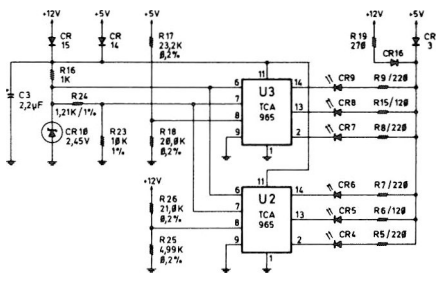
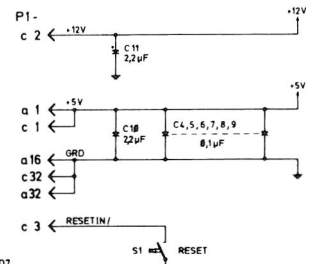
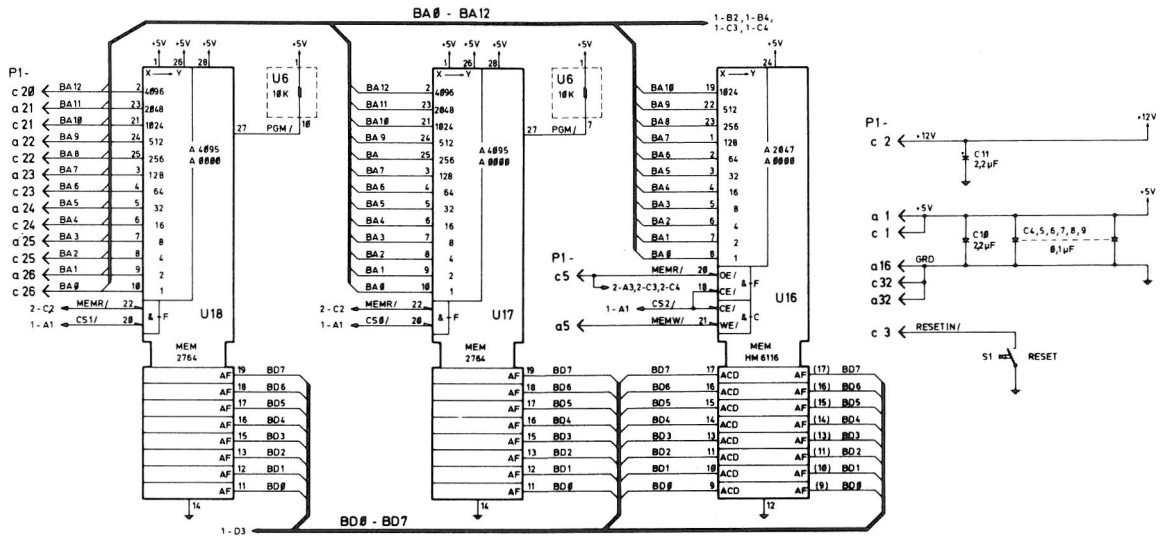
P1/5 HORIZONTAL DRIVE

REMOTE BRIGHTNESS

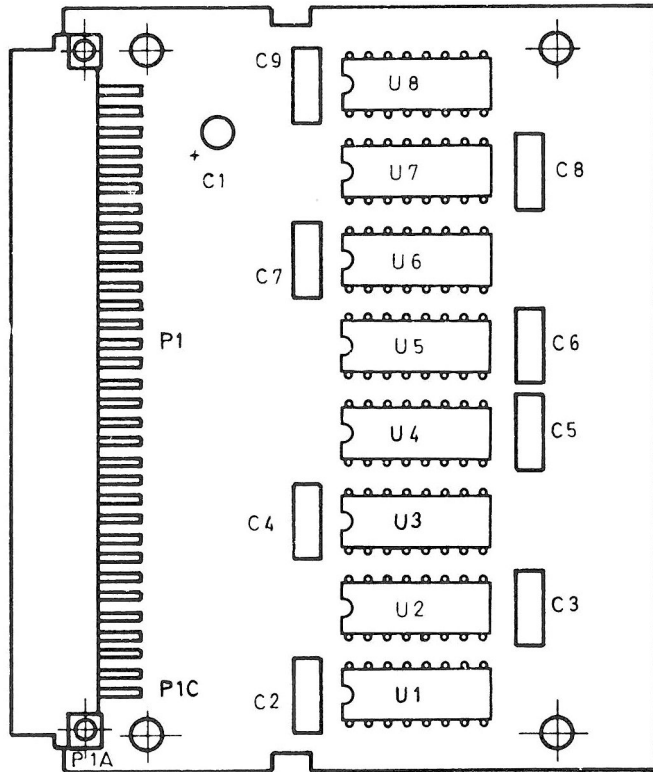
DIAGNOSTIC MODULE



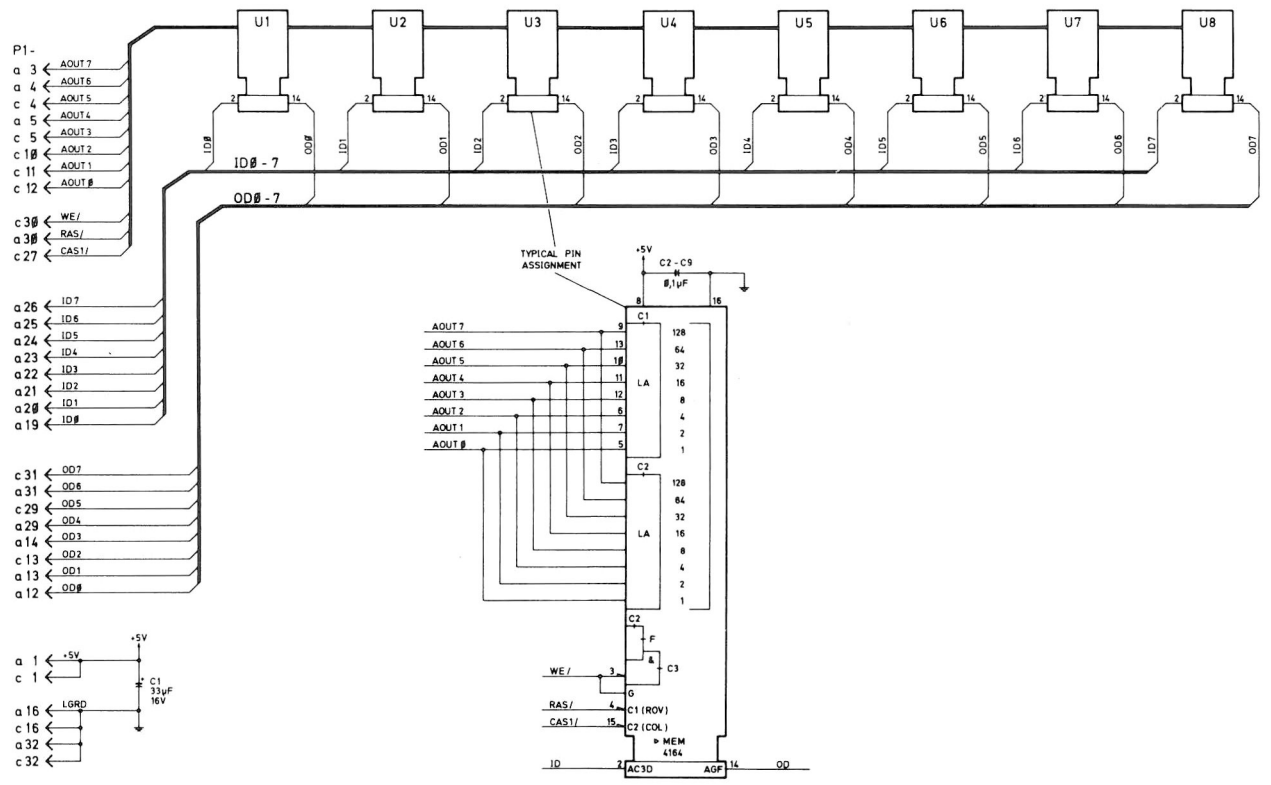




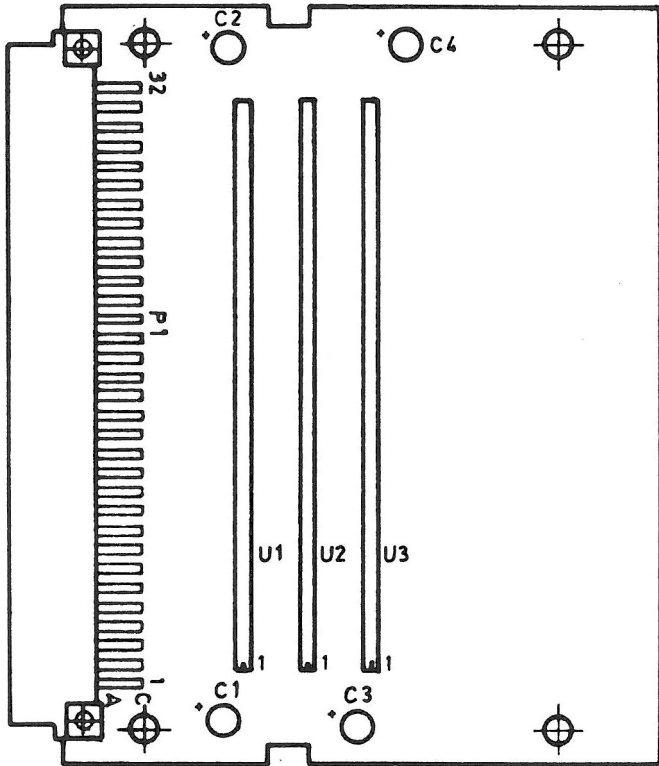
MEMORY EXTENSION 64K

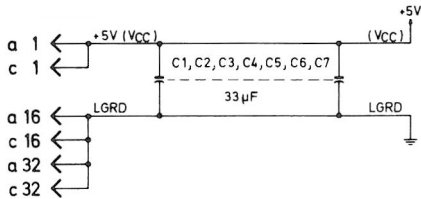
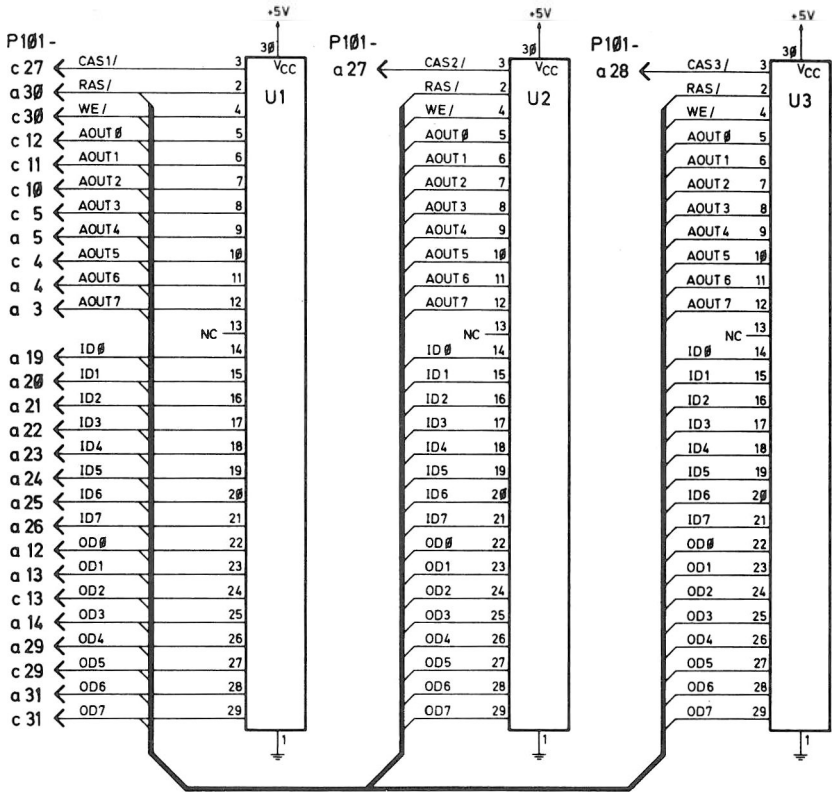


D - RAM

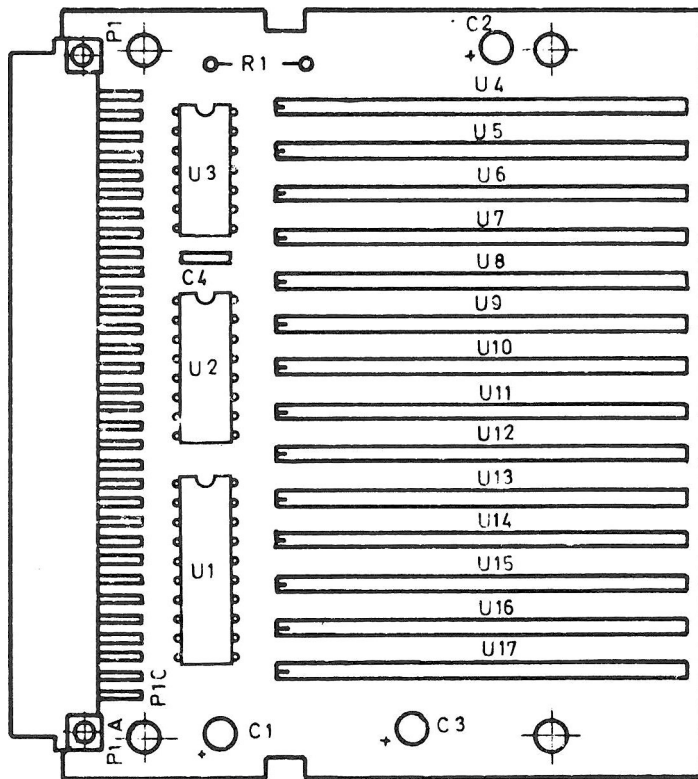


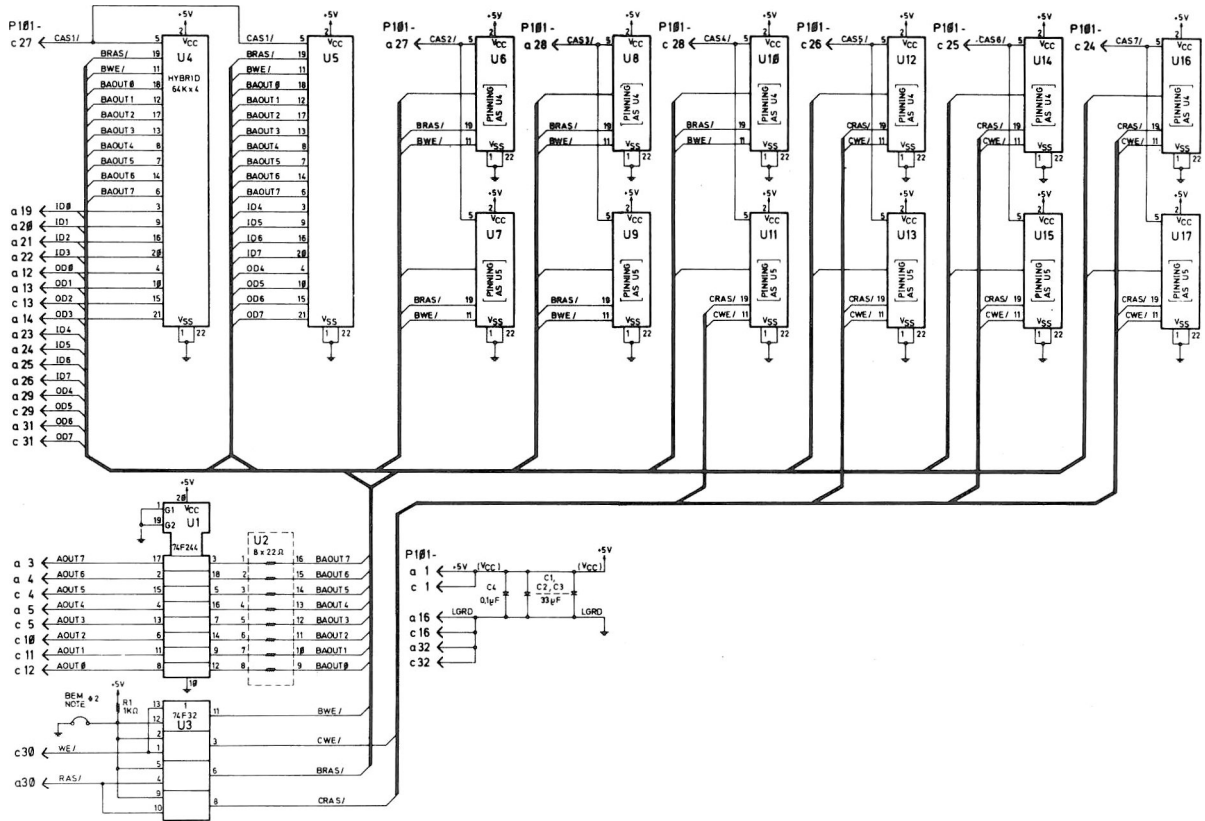
MEMORY EXTENSION 192K



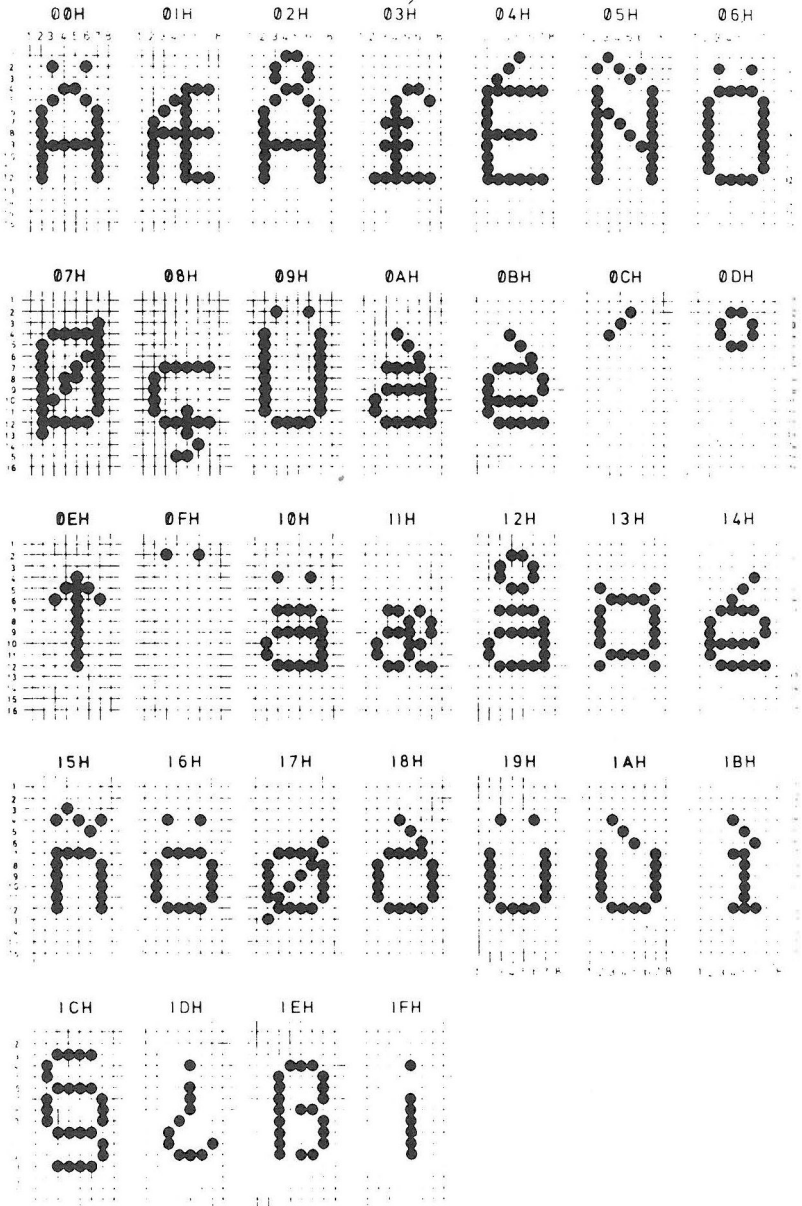


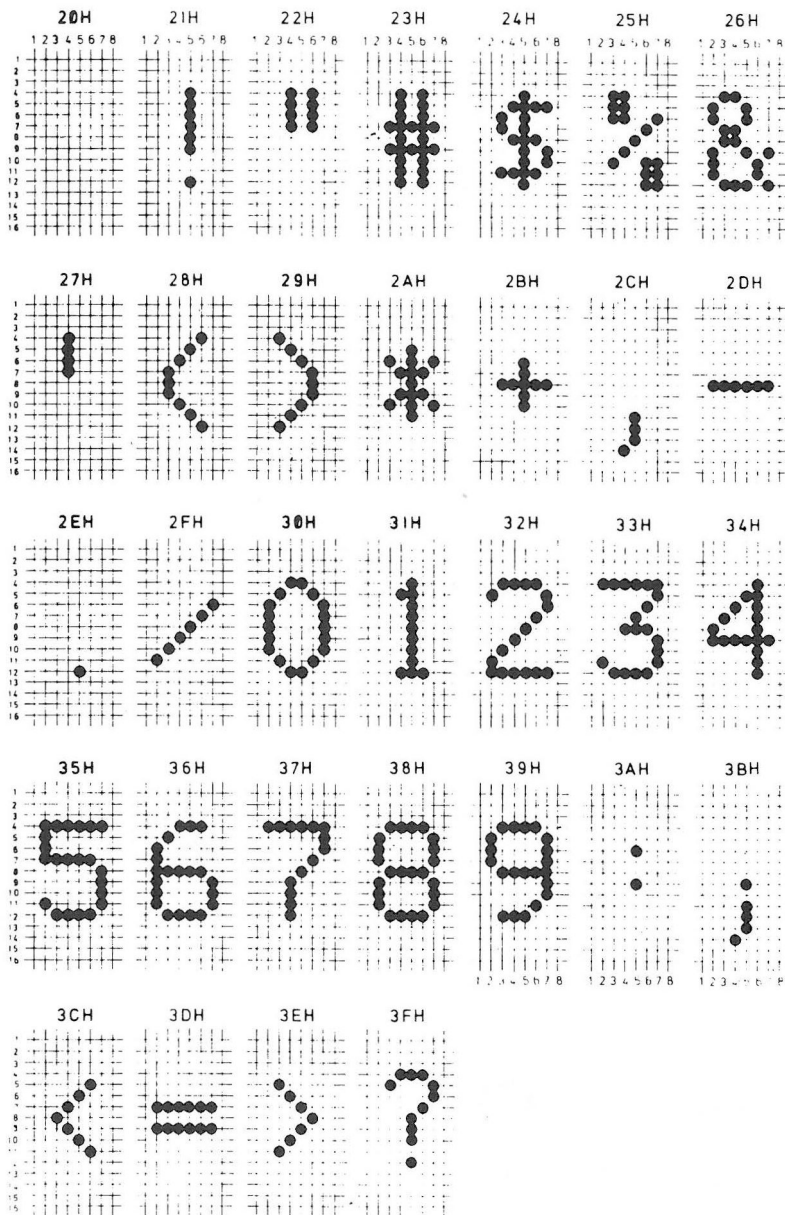
MEMORY EXTENSION 448K

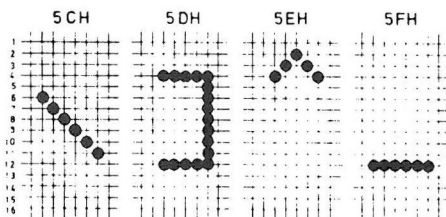
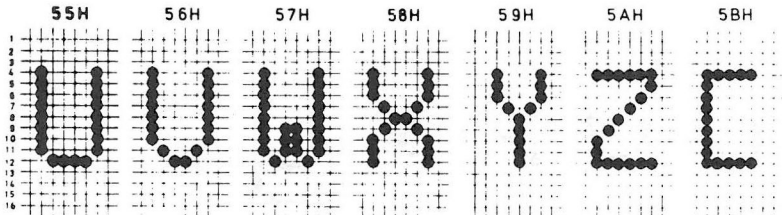
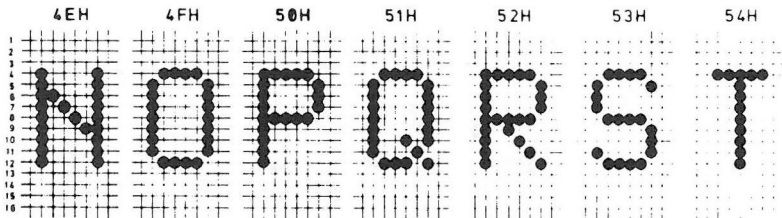
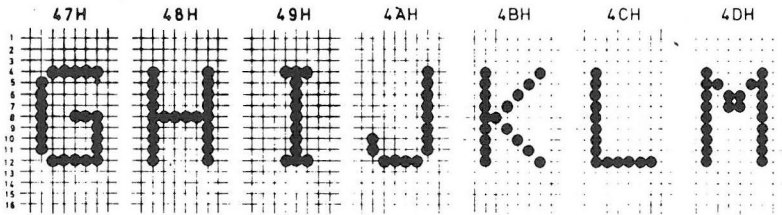
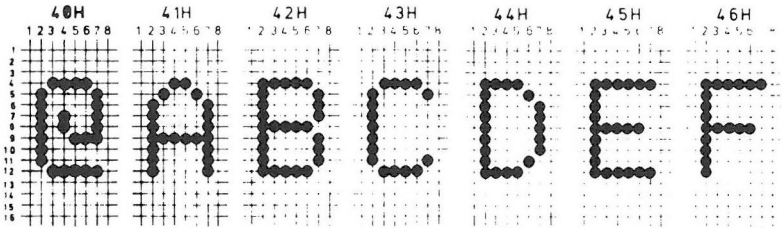


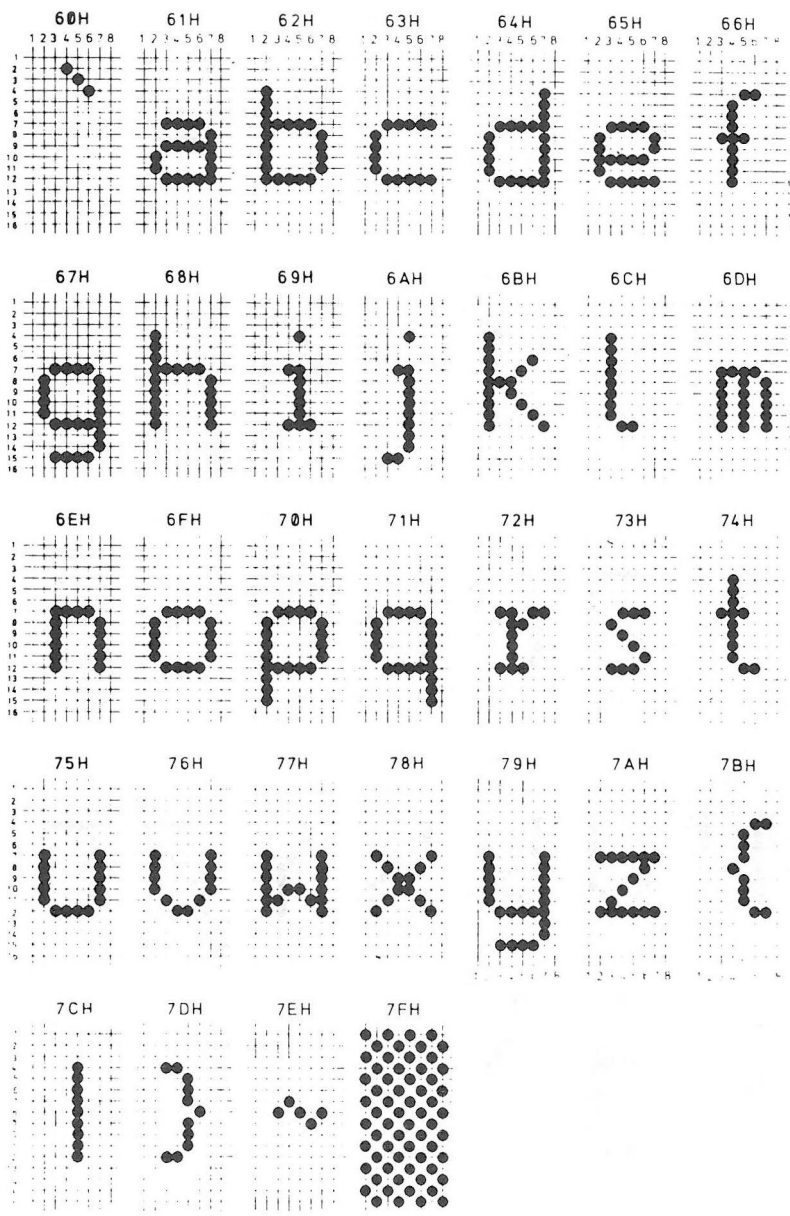


MATRIX CHARACTER SET










CHARACTER GENERATOR TABLE

Langu grp. II Extension

	00	10	20	30	40	50	60	70	80	90
0	Ä ä		0	@	P	`	p	Ã	ã	
1	Æ æ	!	1	A	Q	a	q	Õ	õ	
2	Å å	"	2	B	R	b	r	Ë	ë	
3	£	¤	#	3	C	S	c	s	Ê	ê
4	¢	é	\$	4	D	T	d	t	'N	'n
5	Ñ	ñ	%	5	E	U	e	u	Ç	±
6	Ö	ö	&	6	F	V	f	v	«	â
7	Ø	ø	'	7	G	W	g	w	»	ô
8	ç	ò	(8	H	X	h	x	Ð	đ
9	Ü	ü)	9	I	Y	i	y	Ć	ć
A	à	ù	*	:	J	Z	j	z	Č	č
B	è	ì	+	;	K	Ĉ	k	{	Š	š
C	´	§	,	<	L	\	l		Ž	ž
D	°	¿	-	=	M	Ĵ	m	}	²	û
E	↑	ß	.	>	N	^	n	~	³	î
F	¨	ı	/	?	O	_	o	■	½	ï

KEYBOARD OUTPUT ASSIGNMENTS

	00	10	20	30	40	50	60	70	80	90
0	NUL	DLE		0	@	P	`	p		
1	SOH	DC1	!	1	A	Q	a	q		
2	STX	DC2	"	2	B	R	b	r		
3	ETX	DC3	#	3	C	S	c	s		
4	EOT	DC4	\$	4	D	T	d	t		
5	ENQ	NAK	%	5	E	U	e	u		
6	ACK	SYN	&	6	F	V	f	v	CLR	CLR
7	BEL	ETB	'	7	G	W	g	w		
8	BS	CAN	(8	H	X	h	x	NEW LINE	
9	HT	EM)	9	I	Y	i	y		
A	LF	SUB	*	:	J	Z	j	z		
B	VT	ESC	+	;	K	[k	{	BSp	
C	FF	FS	,	<	L	\	l			
D	CR	GS	-	=	M]	m	}		
E	SO	RS	.	>	N	^	n	~		
F	SI	US	/	?	O	_	o	DEL		

 such marked fields are assigned to language dependent symbol extensions

PROGRAMMABLE COMMUNICATIONS INTERFACE DATA

MODE REGISTER 1 (MR1)

MR17	MR16	MR15	MR14	MR13	MR12	MR11	MR10
		Parity Type	Parity Control	Character Length		Mode and Baud Rate Factor	
ASYNCH: STOP BIT LENGTH 00 = Invalid 01 = 1 Stop bit 10 = 1 1/2 Stop bits 11 = 2 Stop bits		0 = Odd 1 = Even	0 = Disabled 1 = Enabled	00 = 5 Bits 01 = 6 Bits 10 = 7 Bits 11 = 8 Bits	00 = Synchronous 1X rate 01 = Asynchronous 1X rate 10 = Asynchronous 16X rate 11 = Asynchronous 64X rate		
SYNCH: NUMBER OF SYN CHAR 0 = Double syn 1 = Single syn	SYNCH: TRANSPARENCY CONTROL 0 = Normal 1 = Transparent						
Note Baud rate factor in asynchronous applies only if external clock is selected. Factor is 16X if internal clock is selected.							

MODE REGISTER 2 (MR2)

MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20
		Transmitter Clock	Receiver Clock	Baud Rate Selection			
Always zero in NCR DMV		0 = External 1 = Internal	0 = External 1 = Internal	0000 = 50 Baud 0001 = 75 0010 = 110 0011 = 134.5 0100 = 150 0101 = 300 0110 = 600 0111 = 1200	1000 = 1800 Baud 1001 = 2000 1010 = 2400 1011 = 3600 1100 = 4800 1101 = 7200 1110 = 9600 1111 = 19.200		

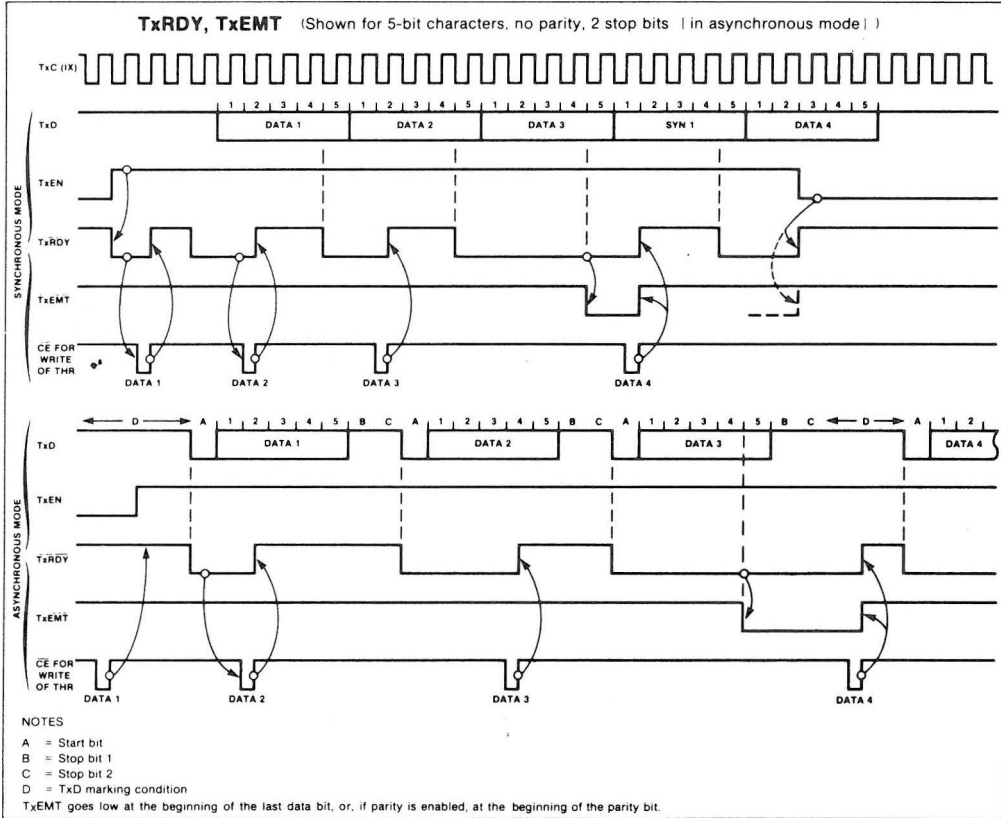
COMMAND REGISTER (CR)

CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
Operating Mode		Request to Send	Reset Error		Receive Control (RxEN)	Data Terminal Ready	Transmit Control (TxEN)
00 = Normal operation 01 = Asynch: automatic echo mode Synch: SYN and /or DLE stripping mode 10 = Local loop back 11 = Remote loop back		0 = Force $\overline{\text{RTS}}$ Output High 1 = Force $\overline{\text{RTS}}$ Output low	0 = Normal 1 = Reset error flag in status reg (FE, OE, PE/DLE detect)	ASYNCH: FORCE BREAK 0 = Normal 1 = Force break SYNCH: SEND DLE 0 = Normal 1 = Send DLE	0 = Disable 1 = Enable	0 = Force $\overline{\text{DTR}}$ Output High 1 = Force $\overline{\text{DTR}}$ Output low	0 = Disable 1 = Enable

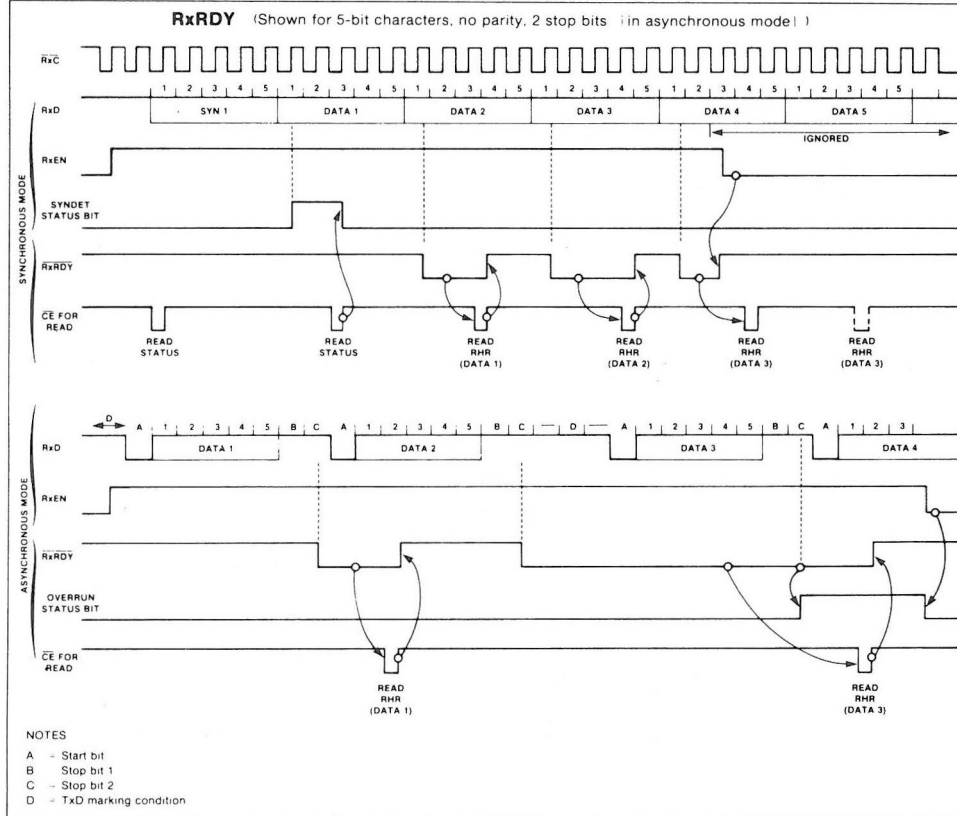
STATUS REGISTER (SR)

SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
Data Set Ready	Data Carrier Detect	FE/SYN Detect	Overrun	PE/DLE Detect	TxE $\overline{\text{MT}}$ /D $\overline{\text{SCHG}}$	RxRDY	TxRDY
0 = $\overline{\text{DSR}}$ input is high 1 = $\overline{\text{DSR}}$ input is low	0 = $\overline{\text{DCD}}$ input is high 1 = $\overline{\text{DCD}}$ input is low	ASYNCH: 0 = Normal 1 = Framing error SYNCH: 0 = Normal 1 = SYN char detected	0 = Normal 1 = Overrun error	ASYNCH: 0 = Normal 1 = Parity error SYNCH: 0 = Normal 1 = Parity error or DLE char received	0 = Normal 1 = Change in $\overline{\text{DSR}}$ or $\overline{\text{DCD}}$, or transmit shift register is empty	0 = Receive holding reg empty 1 = Receive holding reg has data	0 = Transmit holding reg busy 1 = Transmit holding reg empty

TIMING DIAGRAMS



TIMING DIAGRAMS (Cont'd)



2651 REGISTER ADDRESSING

K212/K213	K211	\overline{CE}	A1	A0	$\overline{R/W} = A2$	FUNCTION
—	—	1	X	X	X	Tri-state data bus
60H	70H	0	0	0	0	Read receive holding register
64H	74H	0	0	0	1	Write transmit holding register
61H	71H	0	0	1	0	Read status register
65H	75H	0	0	1	1	Write SYN1/SYN2/DLE registers
62H	72H	0	1	0	0	Read mode registers 1/2
66H	76H	0	1	0	1	Write mode registers 1/2
63H	73H	0	1	1	0	Read command register
67H	77H	0	1	1	1	Write command register

Use IN; OUT
opcodes by
Z80, 8088

For more information on the 2651 Programmable Communications Interface, see the manufacturers data sheets.

16-BIT FIRMWARE

```

LOC OBJ          LINE  SOURCE
-----
1 +1 $PL(68) XR SR DR EP HNGE
2
3
4 NAME BOOT_LOADER_LEVEL0
5
6
7 ;*****
8 ;*****
9 ;**
10 ;**          DMS 16 BIT EXTENSION FIRMWARE
11 ;**
12 ;**
13 ;**
14 ;**          VERSION:          1.0
15 ;**
16 ;**
17 ;**
18 ;**
19 ;**
20 ;**          COPYRIGHT WCR CORP. 1983
21 ;**
22 ;*****
23 ;*****
24
25
26
27
28 DATA SEGMENT WORD COMMON 'DATA'
29
30 BUFFER EQU THIS WORD
31 M LABEL BYTE
32
33 ;
34 ORG OF800H
35 ;
36 CURSX DB ? ;CURSOR - X - POSITION
37 CURSY DB ? ;CURSOR - Y - POSITION
38 ;
39
40 ; *** COMMAND MAIL BLOCK OF LANGUAGE AND IO BOARD ***
41 ; *****
42
43 ORG OFE01H
44
45 FE01 ?? SB8 DB ? ;STATUS BLOCK FROM Z80
46
47 FE04 ORG OFE04H
48
49 FE04 ?? CB16 DB ? ;COMMAND BLOCK TO 16 BIT
50 FE05 ?? SB16 DB ? ;STATUS BLOCK RETURN OF 16 BIT
51 FE06 ?? MEM16 DB ? ;AVAILABLE MEMORY SIZE
52
53
54 DATA ENDS
55
56
57 $ EJECT

```

BOOT_LOADER.LEVEL0

LOC	OBJ	LINE	SOURCE
		59	
		60	PUBLIC SWITCH,EXECUTE_SOFTWARE,SWITCH_TEST
		61	
		62	EXTRM MONITOR:NEAR,STARTS:NEAR,DISPL:NEAR
		63	EXTRM KEYINP:NEAR,L_13:NEAR,DISPLAY:NEAR
		64	
		65	
----		66	
		67	BOOTER SEGMENT WORD COMMON 'CODE'
		68	
		69	
		70	ASSUME DS:DATA,CS:BOOTER,SS:DATA,ES:DATA
		71	
		72	
		73	!DEFINE (JUMPER)(DW MONITOR DW STARTS)
		74	
		75	; *** "OUT" EQUATES ***
		76	; *****
		77	
0010		78	RAMSEL EQU 10H ;RAMS SELECT
0011		79	ROMSEL EQU 11H ;ROM SELECT
0000		80	PROCCH EQU 000H ;PROCESSOR CHANGE
		81	
		82	;*** ASCII EQUATES ***
		83	;*****
		84	
0088		85	NEWL EQU 88H ;NEW LINE COMMAND (CR+LF)
		86	
		87	
		88	; *** JUMP TABLE ***
		89	; *****
		90	
0000 E90000	E	91	JMP DISPLAY
		92	
		93	
		94	; *** COMMAND PHASE VERTEILER ***
		95	; *****
		96	
0003 B8----	R	97	BEGIN: MOV AX,DATA
0006 B8E0		98	MOV SS,AX
0008 B8E0		99	MOV SP,AX
000A B8E8		100	MOV DS,AX
000C BEC0		101	MOV ES,AX
000E EB00		102	JMP SHORT LOOK_MAIL_BOX
		103	
		104	
0010		105	COMMAND EQU THIS WORD
		106	!JUMPER
0014 3900	E	110	DW EXECUTE_SOFTWARE ;PHASE 2
0016 4800	R	111	DW LEVEL0 ;PHASE 3
		112	
		113	
0018 E611		114	SWITCH: OUT ROMSEL,AL
001A E600		115	OUT PROCCH,AL ;RETURN TO 8 BIT
001C 90		116	MOP
		117	
0010		118	LOOK_MAIL_BOX:
0010 E610		119	OUT RAMSEL,AL ;ALLWAYS RAM SELECTION
001F A004FE	R	120	MOV AL,CB16
0022 250F00		121	AND AX,0FH

BOOT LOADER LEVEL 0

LOC	OBJ	LINE	SOURCE		
0025	3C04	122	CMP	AL,04	
0027	73EF	123	JAE	SHORT SWITCH	
0029	D1E0	124	SHL	AX,1	
002B	8BF0	125	MOV	SI,AX	
002D	2EFA41000	R 126	JMP	COMMANDSTJ	: JUMP TO COMMAND
		127			
0032		128	SWITCH TEST:		
0032	C60605FE00	R 129	MOV	SB16,0	:00=GOOD STATUS
0037	EBDF	130	JMP	SWITCH	
		131			
0039		132	EXECUTE SOFTWARE:		
0039	33C0	133	XOR	AX,AX	
003B	8ED8	134	MOV	DS,AX	
003D	8EC0	135	MOV	ES,AX	
003F	8ED0	136	MOV	SS,AX	
0041	8BE0	137	MOV	SP,AX	
		138			
0043	EA0000----	R 139	JMP	FAR PTR CPMS6	: JUMP IN TO CPMS6 OPERATING SOFTWARE
		140			
		141	;	FFC7	

BOOT_LOADER_LEVEL0

LOC	OBJ	LINE	SOURCE
		142	
		143	
		144	
		145	
		146	*****
		147	
		148	; POWER UP POUTINE
		149	
		150	*****
		151	
		152	
0048		153	LEVEL0:
0048	33C0	154	XOR AX,AX ;SET AX AND START ADDRESS TO 0
004A	8BF0	155	MOV SI,AX
004C	890010	156	MOV CX,1000H ;ROM-TEST OVER 4K
004F	2EAC	157	SUM1: LODS CS:M
0051	02E0	158	ADD AH,AL
0053	E2FA	159	LOOP SUM1
0055	80C4D0	160	ADD AH,D
0058	7407	161	JZ SHORT RAMTST ;ROM-TEST O.K.
		162	
005A		163	LEVEL0_FAIL:
005A	C60605FEFF	164	MOV SB16,OFFH
005F	EBB7	165	JMP SWITCH ;LEVEL 0 FAILED
		166	
		167	
		168	; *** RAM-TEST ***
		169	; *****
		170	
0061		171	RAMTST:
0061	C60601F817	172	MOV CURSY,20
0066	C6060DF800	173	MOV CURSX,D
0068	8D1EE001	174	LEA BX,MESS1 ;MESSAGE MEMORY TEST OF BANK 0
006F	E80000	175	CALL DISP1
0072	8D1E1102	176	LEA BX,MESS4
0076	E80000	177	CALL DISP1
		178	
		179	
0079	89AA0A	180	MOV CX,2000H/3 ;WRITE A UNEVEN BYTE MUSTER IN MEMORY
007C	33FF	181	XOR DI,D1
007E		182	RAMTST1A:
007E	B8AA55	183	MOV AX,55AAH
0081	AB	184	STOSW
0082	8000	185	MOV AL,D0
0084	AA	186	STOSB
0085	E2F7	187	LOOP RAMTST1A
0087	B8AA55	188	MOV AX,55AAH
008A	AB	189	STOSW
		190	
008B	89AA0A	191	MOV CX,2000H/3
008E	33FF	192	XOR DI,D1
0090		193	RAMTST1B:
0090	B8AA55	194	MOV AX,55AAH
0093	AF	195	SCASW
0094	7542	196	JNZ ERROR0
0096	8000	197	MOV AL,D0
0098	AE	198	SCASB
0099	753D	199	JNZ ERROR0
009B	E2F3	200	LOOP RAMTST1B
009D	B8AA55	201	MOV AX,55AAH
00A0	AF	202	SCASW
00A1	7535	203	JNZ ERROR0
		204	

BOOT_LOADER_LEVEL0

LOC	OBJ	LINE	SOURCE
		323	
016B	86E0	324	XCHG AH,AL
0160	FECF	325	DEC BH
016F	75EC	326	JNZ MEM_TEST
		327	
0171		328	GOOD_BANK:
0171	FEC3	329	INC BL ;GOOD BANK INC BANK INDICATOR
0173	8B0C	330	MOV CX,BP
0175	E2B1	331	LOOP NEXT_BANK ;ALL BANKS ?
0177	EB1B	332	JMP SHORT_MEMORY_END
		333	
0179	33C0	334	ERROR1: XOR AX,AX
017B	8ED8	335	MOV DS,AX
017D	8EC0	336	MOV ES,AX ;SET SEGMENT TO ZERO
017F	53	337	PUSH BX
0180	C60600F81E	R 338	MOV CURSX,30
0185	8D1E0502	R 339	LEA BX,MESS3 ;ERROR DETECTED WRITE FAILED TO SCREEN
0189	E80000	E 340	CALL DISPL
018C	E80000	E 341	ERR11: CALL KEYIMP
018F	3C88	342	CMF AL,NEUL
0191	75F9	343	JNZ ERR11
0193	58	344	POP BX
		345	
0194		346	MEMORY_END:
0194	33C0	347	XOR AX,AX
0196	8ED8	348	MOV DS,AX
0198	8EC0	349	MOV ES,AX ;SET SEGMENT TO ZERO
019A	C60601F80A	R 350	MOV CURSY,10
019F	881E06FE	R 351	MOV MEM16,BL ;SET COUNT OF MEM BANKS IN MEM16
01A3	C60605FE00	R 352	MOV SB16,0
01A8	E96DFE	353	JMP SWITCH ;JUMP TO 8 BIT
		354	
01AB		355	BANK_DIS:
01AB	50	356	PUSH AX
01AC	33C0	357	XOR AX,AX
01AE	8ED8	358	MOV DS,AX
01B0	8EC0	359	MOV ES,AX ;SET SEGMENT TO ZERO BANK
01B2	8D05	360	MOV AL,5
01B4	F6E3	361	MUL BL
01B6	53	362	PUSH BX
01B7	51	363	PUSH CX
01B8	50	364	PUSH AX
01B9	C60601F817	R 365	MOV CURSY,23 ;SET CURSOR FOR MESSAGES
01BE	C60600F800	R 366	MOV CURSX,0
01C3	8D1EED01	R 367	LEA BX,MESS1 ;WRITE MEMORY TEST ON SCREEN
01C7	E80000	E 368	CALL DISPL
01CA	58	369	POP AX
		370	
01CB	8B08	371	MOV BX,AX
01CD	8D061A02	R 372	LEA AX,BANKS
01D1	03D8	373	ADD BX,AX
01D3	85D9	374	MOV CH,9
01D5	E80100	E 375	CALL L_13-1 ;WRITE THE TESTED BANK TO SCREEN
01D8	59	376	POP CY
01D9	58	377	POP BX
01DA	58	378	POP AX
01DB	8EDA	379	MOV DS,DX
01DD	8EC2	380	MOV ES,DX
01DF	C3	381	RET
		382	
		383	
01ED	11	384	MESS1 DB 17,'TEST MEMORY BANK '
01E1	54455354204045		

BOOT_LOADER_LEVEL0

LOC	OBJ	LINE	SOURCE
	404F5259204241 4E4820		
01F2	12	385	MESS2 DB 18, 'FATAL MEMORY ERROR'
01F3	464154414C2040 45404F52592045 52524F52		
0205	08	386	MESS3 DB 11, 'FAILED (CR)'
0206	4641494C454420 28435229		
0211	08	387	MESS4 DB 8, ' 0-64K'
0212	20202030203634 48		
021A	20363448203132 38482031393248 20323536482033 32304820333834 48203434384820 35313248	388	BANKS DB ' 64K-128K-192K-256K-320K-384K-448K-512K'
		389	
		390	
		391	
----		392	BOOTER ENDS
		393	
		394	
----		395	SOFTWARE SEGMENT PARA PUBLIC 'CODE'
		396	
0000		397	CPM86 EQU THIS FAR
		398	
----		399	SOFTWARE ENDS
		400	
		401	
		402	END BEGIN

BOOT LOADER LEVEL 0

LOC	OBJ	LINE	SOURCE	
00A3	BA0010	205	MOV	DX,1000H
00A6	B80201	206	MOV	AX,0102H
00A9		207	RAMTST1:	;TEST WITH A ROTATING ONE
00A9	33FF	208	XOR	DI,DI
00AB	B8CA	209	MOV	CX,DX
00AD	F3	210	REP	STOSW
00AE	AB			
00AF	33FF	211	XOR	DI,DI
00B1	B8CA	212	MOV	CX,DX
00B3	F3	213	REPZ	SCASW
00B4	AF			
00B5	7521	214	JNZ	ERROR0
00B7	D0C0	215	ROL	AL,1
00B9	D0C4	216	ROL	AH,1
00BB	73EC	217	JMC	RAMTST1
		218		
00BD	B702	219	MOV	BH,2
00BF	B855AA	220	MOV	AX,0AA55H
00C2		221	RAMTST2:	
00C2	B8CA	222	MOV	CX,DX
00C4	33FF	223	XOR	DI,DI
00C6	F3	224	REP	STOSW ;STORE AA55
00C7	AB			
		225		
00C8	33FF	226	XOR	DI,DI
00CA	B8CA	227	MOV	CX,DX
00CC	F3	228	REPZ	SCASW
00CD	AF			
		229		
00CE	75D8	230	JNZ	ERROR0
		231		
00D0	B6E0	232	XCHG	AH,AL
00D2	FECF	233	DEC	BH
00D4	75EC	234	JNZ	RAMTST2
00D6	EBDE	235	JMP	SHORT MEM_BANK_TEST
		236		
		237		
00D8	C6060DF81E	R 238	ERROR0: MOV	CURSX,30 ;WRITE 'FATAL MEMORY ERROR'
00D0	8D1EF201	R 239	LEA	BX,MESS2 ;AND STOP PROCESSING
00E1	E80000	E 240	CALL	DISPL
00E4	EBFE	241	JMP	SHORT \$
		242		
		243		
		244	;	MEMORY EXTENSION TEST
		245	;	=====
		246		
		247		
00E6		248	MEM_BANK_TEST:	
00E6	B90300	249	MOV	CX,3 ;SET BANK COUNTER TO MAX BANKS=256K (512K)
00E9	803E01FE09	R 250	CMPL	SBB,09 ;VERSION NUMBER) 10 ?
00EE	7203	251	JB	UNDER
00FD	83C104	252	ADD	CX,4 ;SET BANK COUNTER TILL 512K
00F3	BA0000	253	UNDER: MOV	DX,0
00F6	B300	254	MOV	BL,0 ;SET BANK INDICATOR
		255		
00F8		256	NEXT_BANK:	
00F8	B8E9	257	MOV	BP,CX ;SAVE BANK COUNTER
00FA	B1C20010	258	ADD	DX,1000H ;SET SEGMENT FOR NEXT BANK
00FE	BEDA	259	MOV	DS,DX
0100	BEC2	260	MOV	ES,DX
		261		
0102	B855AA	262	MOV	AX,0AA55H
0105	A30000	R 263	MOV	BUFFER,AX ;TEST THE FIRST TWO BYTES OF A BANK

BOOT_LOADER_LEVEL0

LOC	OBJ	LINE	SOURCE	
0108	3B060000	R 264	CMP AX,BUFFER	:FOR A PRESENT BANK
010C	7403	265	JZ L_200	
010E	E98300	266	JMP MEMORY_END	
0111	E89700	267	L_200: CALL BANK_DIS	
		268		
0114	B95555	269	MOV CX,OFFFH/3	:WRITE AN UNEVEN BYTE PATTERN IN MEMORY
0117	33FF	270	XOR DI,DI	
		271		
0119		272	ADDTSTA:	
0119	B8AA55	273	MOV AX,55AAH	
011C	AB	274	STOSW	
011D	8000	275	MOV AL,00	
011F	AA	276	STOSB	
0120	E2F7	277	LOOP ADDTSTA	
0122	8055	278	MOV AL,55H	
0124	AA	279	STOSB	
		280		
0125	B95555	281	MOV CX,OFFFH/3	
0128	33FF	282	XOR DI,DI	
012A		283	ADDTSTB:	
012A	B8AA55	284	MOV AX,55AAH	
012D	AF	285	SCASW	
012E	7549	286	JNZ ERROR1	
0130	8000	287	MOV AL,0	
0132	AE	288	SCASB	
0133	7544	289	JNZ ERROR1	
0135	E2F3	290	LOOP ADDTSTB	
0137	8055	291	MOV AL,55H	
0139	AE	292	SCASB	
013A	753D	293	JNZ ERROR1	
		294		
013C	33FF	295	XOR DI,DI	
013E	BE0080	296	MOV SI,8000H	:COUNTER OF A ENTIRE BANK 64 K
		297		
0141	B80201	298	MOV AX,0102H	
0144		299	ROT_ONE:	
0144	33FF	300	XOR DI,DI	
0146	8BCE	301	MOV CX,SI	
0148	F3	302	REP STOSW	
0149	AB			
014A	33FF	303	XOR DI,DI	
014C	8BCE	304	MOV CX,SI	
014E	F3	305	REPZ SCASW	
014F	AF			
0150	7527	306	JNZ ERROR1	
0152	DOC0	307	ROL AL,1	
0154	DOC4	308	ROL AH,1	
0156	73EC	309	JMC ROT_ONE	
		310		
0158	8702	311	MOV BH,2	:COUNT OF BIT PATTERN
015A	B855AA	312	MOV AX,DAA55H	
015D		313	MEM_TEST:	
015D	8BCE	314	MOV CX,SI	:SET COUNTER FOR A ENTIRE BANK = 8000 WORDS
015F	33FF	315	XOR DI,DI	
0161	F3	316	REP STOSW	:STORE AX INTO MEMORY
0162	AB			
		317		
0163	33FF	318	XOR DI,DI	
0165	8BCE	319	MOV CX,SI	
0167	F3	320	REPZ SCASW	:CHECK MEMORY WITH ACCU
0168	AF			
		321		
0169	750E	322	JNZ ERROR1	

BOOT_LOADER_LEVEL0

XREF SYMBOL TABLE LISTING

NAME	TYPE	VALUE	ATTRIBUTES, XREFS
??SEG	SEGMENT		SIZE=000H PARA PUBLIC
ADDTSTA	L NEAR	0119H	BOOTER 272# 277
ADDTSTB	L NEAR	012AH	BOOTER 283# 290
BANK_DIS	L NEAR	01ABH	BOOTER 267 355#
BANKS	V BYTE	021AH	BOOTER 372 388#
BEGIN	L NEAR	0003H	BOOTER 97# 402 402
BOOTER	SEGMENT		SIZE=0241H WORD COMMON 'CODE' 67# 70 392
BUFFER	V WORD	0000H	DATA 30# 263 264
CB16	V BYTE	FED4H	DATA 49# 120
COMMAND	V WORD	0010H	BOOTER 105# 126
CPMB6	L FAR	0000H	SOFTWARE 139 397#
CURSX	V BYTE	F800H	DATA 36# 173 238 338 366
CURSY	V BYTE	F801H	DATA 37# 172 350 365
DATA	SEGMENT		SIZE=FED7H WORD COMMON 'DATA' 28# 54 70 70 97
DISPL	L NEAR	0000H	EXTRN 62# 175 177 240 340 368
DISPLAY	L NEAR	0000H	EXTRN 63# 91
ERR11	L NEAR	018CH	BOOTER 341# 343
ERROR0	L NEAR	0008H	BOOTER 196 199 203 214 230 238#
ERROR1	L NEAR	0179H	BOOTER 28# 289 293 306 322 334#
EXECUTE_SOFTWARE	L NEAR	0039H	BOOTER PUBLIC 60 110 132#
GOOD_BANK	L NEAR	0171H	BOOTER 328#
KEYINP	L NEAR	0000H	EXTRN 63# 341
L_13	L NEAR	0000H	EXTRN 63# 375
L_200	L NEAR	0111H	BOOTER 265 267#
LEVELO	L NEAR	0048H	BOOTER 111 153#
LEVELO_FAIL	L NEAR	005AH	BOOTER 163#
LOOK_MAIL_BOX	L NEAR	0010H	BOOTER 102 118#
M	V BYTE	0000H	DATA 31# 157
MEM_BANK_TEST	L NEAR	0DE6H	BOOTER 235 248#
MEM_TEST	L NEAR	015DH	BOOTER 313# 326
MEM16	V BYTE	FED6H	DATA 51# 351
MEMORY_END	L NEAR	0194H	BOOTER 266 332 346#
NESS1	V BYTE	01E0H	BOOTER 174 367 384#
NESS2	V BYTE	01F2H	BOOTER 239 385#
NESS3	V BYTE	0205H	BOOTER 339 386#
NESS4	V BYTE	0211H	BOOTER 176 387#
MONITOR	L NEAR	0000H	EXTRN 62# 107
NEWL	NUMBER	0088H	85# 342
NEXT_BANK	L NEAR	00F8H	BOOTER 256# 331
PROCCH	NUMBER	0000H	80# 115
RAMSEL	NUMBER	0010H	78# 119
RAMTST	L NEAR	0061H	BOOTER 161 171#
RAMTST1	L NEAR	00A9H	BOOTER 207# 217
RAMTST1A	L NEAR	007EH	BOOTER 182# 187
RAMTST1B	L NEAR	0090H	BOOTER 193# 200
RAMTST2	L NEAR	00C2H	BOOTER 221# 234
ROMSEL	NUMBER	0011H	79# 114
ROT_ONE	L NEAR	0144H	BOOTER 299# 309
SB16	V BYTE	FED5H	DATA 50# 129 164 352
SBB	V BYTE	FED1H	DATA 45# 250
SOFTWARE	SEGMENT		SIZE=0000H PARA PUBLIC 'CODE' 395# 399
STARTS	L NEAR	0000H	EXTRN 62# 108
SUM1	L NEAR	004FH	BOOTER 157# 159
SWITCH	L NEAR	0018H	BOOTER PUBLIC 60 114# 123 130 165 353
SWITCH_TEST	L NEAR	0032H	BOOTER PUBLIC 60 128#
UNDER	L NEAR	00F3H	BOOTER 251 253#

END OF SYMBOL TABLE LISTING

BOOTER_IO_DRIVER

SERIES-111 8086/8087/8088 MACRO ASSEMBLER V1.1 ASSEMBLY OF MODULE BOOTER_IO_DRIVER
 OBJECT MODULE PLACED IN :F1:DH1610.OBJ
 ASSEMBLER INVOKED BY: AB6.86 :F1:DH1610.SRC

LOC	OBJ	LINE	SOURCE
		1 +1	\$PAGELNGTH(68) XREF SYMBOLS DEBUG EP NOMR
		2	
		3	
		4	NAME BOOTER_IO_DRIVER
		5	
		6	
		7	*****
		8	*****
		9	***
		10	*** DHS 16 BIT EXTENSION FIRMWARE ***
		11	***
		12	*** IDENTIFIER: AHF-DH16FI 0101-00 ***
		13	***
		14	*** VERSION: 1.0 ***
		15	***
		16	*** DATE: 24.02.83 ***
		17	***
		18	*** ORIG. PLANT: MCR AUGSBURG ***
		19	***
		20	*** COPYRIGHT MCR CORP. 1983 ***
		21	***
		22	*****
		23	*****
		24	
		25	
		26	
		27	
		28	DATA _SEGMENT WORD COMMON 'DATA'
		29	
	0000	30	BUFFER EQU THIS WORD
	0000	31	H LABEL BYTE
		32	
	F800	33	ORG DF800H
		34	
	F800 ??	35	CURSX DB ? ;CURSOR - X - POSITION
	F801 ??	36	CURSY DB ? ;CURSOR - Y - POSITION
		37	
	F830	38	ORG DF830H
		39	
	F830 ??	40	BUFSTA DB ? ;CRT-BUFFER START ADR.
		41	
	F900	42	ORG DF900H
		43	
	F900 ??	44	CEADL DB ? ;LOW BYTE OF CURSOR
	F901 ??	45	CEADH DB ? ;HIGH BYTE OF CURSOR
	F902 ??	46	CDAD DB ? ;POS OF PIXEL
	F903 ????	47	SP1 DW ? ;START OF PAGE 1
	F905 ??	48	LP11 DB ? ;LENGTH OF PAGE 1 LOW
	F906 ??	49	LP12 DB ? ; " " " 1 HIGH
	F907 ????	50	SP2 DW ? ;START OF PAGE 2
	F909 ??	51	LP21 DB ? ;LENGTH OF PAGE 2 LOW
	F90A ??	52	LP22 DB ? ; " " " 2 HIGH
	F90B ??	53	INWFL6 DB ?
		54	
	----	55	DATA ENDS
		56	
		57 +1	* EJECT

BOOTER_TO_DRIVER

LOC	OBJ	LINE	SOURCE
		58	
		59	
		60	
		61	PUBLIC KEYIMP,KYLOOK,MODCRT,SETCU,CALCHR,DISPL,L_13,DISPL1
		62	PUBLIC DISPLAY
---		63	
		64	BOOTER SEGMENT BYTE COMMON 'CODE'
		65	
		66	
		67	ASSUME DS:DATA,CS:BOOTER,SS:DATA,ES:DATA
0250		68	
		69	ORG 250H
		70	
		71	
		72	; *** "OUT" EQUATES ***
		73	; *****
		74	
0010		75	RANSEL EQU 10H ;RANS SELECT
0011		76	RONSEL EQU 11H ;RON SELECT
		77	
		78	
		79	
		80	;*****
		81	;
		82	; DRIVER FOR KEYBOARD-CONTROLLER 8741
		83	;
		84	;*****
		85	
		86	
		87	; *** "IN/OUT" EQUATES ***
		88	; *****
		89	
0040		90	KEYDAT EQU 40H ;READ KEYBOARD DATA
0041		91	KEYSTA EQU 41H ;READ STATUS
		92	
		93	
0250 E441		94	KEYIMP: IN AL,KEYSTA
0252 D0D8		95	RCR AL,1
0254 73FA		96	JNC KEYIMP
0256 E440		97	IN AL,KEYDAT ;READ KEYBOARD-CHARACTER
0258 C3		98	RET
		99	
0259 E441		100	KYLOOK: IN AL,KEYSTA
025B 24D1		101	AND AL,1
025D 7501		102	JNZ SHORT L_1
025F C3		103	RET
0260 E440		104	L_1: IN AL,KEYDAT
0262 C3		105	RET
		106	
		107	
		108	;*****
		109	;
		110	; CRT-ROUTINES
		111	;
		112	;*****
		113	
		114	
		115	
		116	
		117	; *** CONTROL CHARACTER EQUATES ***
		118	; *****
		119	
0000		120	NOMOV EBX 00H ;NO CURSOR MOVE

BOOTER_IO_DRIVER

LOC	OBJ	LINE	SOURCE	
002A		121	PROMPT EQU	2AH
0020		122	SPACE EQU	20H
0088		123	MEML EQU	8BH
0001		124	HOME EQU	01H
000C		125	SCRER EQU	0CH
008B		126	RUBOUT EQU	8BH
		127		
		128		
		129		
		130	;*****	
		131	;	
		132	; GDC-EQUATES	
		133	;	
		134	;*****	
		135		
		136		
0050		137	CHARA EQU	80 ;80 CHARACTER/LINE
		138		
		139		
		140	; *** GDC PORT EQUATES ***	
		141	; *****	
		142		
00A1		143	GDCOM EQU	0A1H ;WRITE COMMAND
00A0		144	GDCSTA EQU	0A0H ;READ STATUS
00A0		145	GDCPAR EQU	0A0H
		146		
		147	; *** GDC STATUS EQUATES ***	
		148	; *****	
		149		
0002		150	FIFULL EQU	02H ;FIFO IS FULL
		151		
		152	; *** GDC COMMAND EQUATES ***	
		153	; *****	
		154		
0068		155	START EQU	6BH ;START DISPLAY/END IDLE-NODE
0049		156	CURS EQU	49H ;SPECIFY CURSOR-POSITION
004A		157	MASK EQU	4AH ;LOAD MASK-REGISTER
004C		158	FIGS EQU	4CH ;SPECIFY FIGURE DRAWING-PARAMETER
0070		159	PRAM EQU	70H ;LOAD PARAMETER-RAM
0020		160	WDAT EQU	20H ;WRITE DATA INTO DISPLAY MEMORY
		161		;(COMMAND+TYPE+MODE)
0000		162	TYWORD EQU	00H ;DATA TRANSFER BYTES
		163		;(MODE OF RW MEMORY CYCLE
0000		164	NOREPL EQU	00H ;REPLACE WITH PATTERN
		165		
		166		
		167		
		168		
J 55		169	MODCRT: PUSH	BP
4 56		170	PUSH	SI
5 57		171	PUSH	DI
6 53		172	PUSH	BX
7 51		173	PUSH	CX
8 52		174	PUSH	DX
9 9C		175	PUSHF	
A 50		176	PUSH	AX ;SAVE ASCII-CHARACTER
		177		
		178	; *** CHECK IF ANY CONTROL CHARACTER ***	
		179	; *****	
		180		
3 ABFO		181	TEST	AL,0F0H
1 7438		182	JZ	SHORT SETCHA
1 3C00		183	CHP	AL,NOMOV

BOOTER_IO_DRIVER

LOC	OBJ	LINE	SOURCE		
0271	7457	184	JE	GETRE	;NO CURSOR MOVE
0273	3C88	185	CHP	AL,HEUL	
0275	740A	186	JE	SHORT CARRET	;CARRIDGE RETURN
0277	3C01	187	CHP	AL,HOME	
0279	7420	188	JE	SHORT CHP	;HOME POSITION
027B	3C0C	189	CHP	AL,SCRER	
027D	7423	190	JE	SHORT SCERA	;SCREEN ERASE
027F	EB26	191	JMP	SHORT SETCHA	;SET CHARACTER INTO CRT-BUFFER DEPENDING
		192			
0281	C60600F800	R 193	CARRET:	MOV CURSX,0	
0284	A001F8	R 194		MOV AL,CURSY	
0289	FEC0	195		INC AL	
028B	3C19	196	CHP	AL,25	
028D	7507	197	JNE	CAR1	
028F	50	198	PUSH	AX	
0290	E88700	199	CALL	SCROLLM	
0293	58	200	POP	AX	
0294	FEC8	201	DEC	AL	
0296	A201F8	R 202	CAR1:	MOV CURSY,AL	
0299	EB2F	203	JMP	SHORT GETRE	
		204			
		205	;	*** CURSOR HOME POSITION ***	
		206	;	*****	
		207			
0298	33C0	208	CHP:	XOR AX,AX	
0290	A300F8	R 209		MOV WORD PTR CURSX,AX	
02A0	EB28	210	JMP	SHORT GETRE	
		211			
		212	;	*** SCREEN ERASE ***	
		213	;		
02A2	E88800	214	SCERA:	CALL HCLSCR	
02A5	EBF4	215	JMP	CHP	
		216			
		217	;	*** SET CHARACTER ***	
		218	;		
02A7	A100F8	R 219	SETCHA:	MOV AX,WORD PTR CURSX	
02AA	3C50	220	CHP	AL,80	;END OF LINE ?
02AC	7504	221	JNZ	SHORT CHA1	
02AE	32C0	222	XOR	AL,AL	
02B0	FEC4	223	INC	AH	;INCREMENT CURSY
02B2	80FC19	224	CHA1:	CHP AH,25	
02B5	7507	225	JNZ	SHORT CHA2	;END OF SCREEN ?
02B7	50	226	PUSH	AX	;AH = Y POSITION
02B8	E88F00	227	CALL	SCROLLM	;AL = X POSITION
02B8	58	228	POP	AX	
02BC	FECC	229	DEC	AH	
02BE	88C8	230	CHA2:	MOV CX,AX	;CX = Y AND X POSITION
02C0	FEC0	231	INC	AL	;NEXT CURS POSTION
02C2	A300F8	R 232		MOV WORD PTR CURSX,AX	
02C5	58	233	POP	AX	;RESTORE ASCII CHARACTER
02C6	50	234	PUSH	AX	
02C7	E80900	235	CALL	WRNCHR	; (A) = ASCII CHARACTER
		236			; (B)=Y-POS, (C) = X-POS
02CA	58	237	GETRE:	POP AX	
02CB	90	238	POPF		
02CC	5A	239	POP	DX	
02CD	59	240	POP	CX	
02CE	58	241	POP	BX	
02CF	5F	242	POP	DI	
02D0	5E	243	POP	SI	
02D1	5D	244	POP	BP	
02D2	C3	245	RET		
		246			

BOOTER_IO_DRIVER

LOC	OBJ	LINE	SOURCE
		247	
		248	*****
		249	;
		250	; G D C DRIVER ROUTINES
		251	;
		252	*****
		253	
		254	
		255	;ENTRY: (AL) = ASCII CHARACTER
		256	; (CH) = Y-POSITION
		257	; (CL) = X-POSITION
		258	
0203	53	259	WRMCHR: PUSH BX
0204	52	260	PUSH DX
0205	51	261	PUSH CX
0206	50	262	PUSH AX
		263	
0207	E82C00	264	CALL CALCHR ;CALCULATE CHARACTER-POS.
020A	E8FC00	265	CALL SETCU
020D	E81300	266	CALL SETCCA ;SEND CHARACTER
02E0	0020	267	MOV AL,WDAT+TYWORD+MOREPL
02E2	E82E01	268	CALL OUTCMD
02E5	58	269	POP AX ;(A)=ASCII-CHARACTER
02E6	E83001	270	CALL OUTPAR
02E9	A00BF9	271	MOV AL,INWFLG ;FETCH ATTRIBUT
02EC	E82A01	272	CALL OUTPAR
02EF	59	273	POP CX
02F0	5A	274	POP DX
02F1	5B	275	POP BX
02F2	C3	276	RET
		277	
		278	
02F3	E8C400	279	SETCCA: CALL SETMSK
02F6	B04C	280	MOV AL,FIGS
02F8	E81801	281	CALL OUTCMD
02FB	B002	282	MOV AL,2
02FD	E81901	283	CALL OUTPAR
0300	32C0	284	XOR AL,AL
0302	E81401	285	CALL OUTPAR
0305	C3	286	RET
		287	
0306		288	CALCHR:
0306	F70603F9FFFF	289	TEST SP1,0FFFFH
030C	750C	290	JNZ SHORT CALCH1
030E	B050	291	MOV AL,80 ;Y*80 + X = (CEADL)
0310	F6E5	292	MUL CH ;(CH) = Y-POS
0312	B500	293	CALCH2: MOV CH,0
0314	03C1	294	ADD AX,CX
0316	A300F9	295	MOV WORD PTR CEADL,AX
0319	C3	296	RET
		297	
031A		298	CALCH1: ;(CH)=Y-POS
031A	B050	299	MOV AL,80
031C	F6E5	300	MUL CH ;Y * 80
031E	030603F9	301	ADD AX,SP1 ;(SP1) + (Y*80)=AX
0322	BAD007	302	MOV DX,25*80
0325	30C2	303	CMPS AX,DX
0327	72E9	304	JB CALCH2
0329	28C2	305	SUB AX,DX
032B	EBE5	306	JMP CALCH2
		307	
		308	
		309	

BOOTER_IO_DRIVER

LOC	OBJ	LINE	SOURCE
		310	; **** CLEAR SCREEN ****
		311	; *****
		312	
		313	
0320	E86F00	314	MCLSCR: CALL SETCR ;SET CURSOR IN HOME-POS.
0330	E80700	315	CALL CLRL ;CLEAR SCREEN
0333	E85800	316	CALL INIT1A
0336	E88800	317	CALL SCROL1 ;SET PARTITION
0339	C3	318	RET
		319	
		320	
033A	8020	321	CLRL: MOV AL,UDAT+TYWORD+MOREPL
033C	E80400	322	CALL OUTCMD
033F	8020	323	MOV AL,' ' ;SPACE-WRITE
0341	E80500	324	CALL OUTPAR
0344	32C0	325	XOR AL,AL
0346	E80000	326	CALL OUTPAR ;CLEAR ATTRIB.
0349	C3	327	RET
		328	
		329	
		330	
		331	; **** SCROLL SCREEN ****
		332	; *****
		333	
		334	
034A	8B1E03F9	R 335	SCROLLM: MOV BX,SP1
034E	891E0DF9	R 336	MOV WORD PTR CEADL,BX
0352	E88400	337	CALL SETCU ;SET CURSOR
0355	E84200	338	CALL SETMSK ;SET WORD-TRANSFER
		339	
0358	804C	340	MOV AL,FIGS ;SET LENGTH (80-CHAR)
035A	E88400	341	CALL OUTCMD ;FOR DELETE 1 LINE
035D	8002	342	MOV AL,2
035F	E88700	343	CALL OUTPAR
0362	804F	344	MOV AL,80-1
0364	E88200	345	CALL OUTPAR
0367	32C0	346	XOR AL,AL
0369	E8A000	347	CALL OUTPAR
		348	
036C	E8CBFF	349	CALL CLRL ;CLEAR LAST LINE
036F	8B1E0DF9	R 350	MOV BX,WORD PTR CEADL ;SET CURSOR
0373	E84300	351	CALL SETCU
		352	
0376	830603F950	R 353	ADD SP1,80 ;POINTER + 80-CHAR.
		354	
0378	A006F9	R 355	MOV AL,LP12
037E	FEC8	356	DEC AL ;LENGTH1 - 1
0380	7503	357	JNZ SHORT L_31
0382	EB7D90	358	JMP INIT10
0385		359	L_31:
0385	A206F9	R 360	MOV LP12,AL
		361	
0388	FED60AF9	R 362	INC LP22 ;LENGTH2 + 1
038C	EB43	363	JMP SHORT SCROL1
		364	
		365	
		366	
		367	
		368	
038E	33C0	369	INIT1A: XOR AX,AX
0390	A303F9	R 370	MOV SP1,AX
0393	A307F9	R 371	MOV SP2,AX
0396	A20AF9	R 372	MOV LP22,AL

BOOTER_IO_DRIVER

LOC	OBJ	LINE	SOURCE	
0399	C60606F919	R 373	MOV	LP12,25
039E	C3	374	RET	
		375		
039F	E82600	376	SETCR: CALL	CUHOM1 ;WHOLE GRAPHIC HOME
03A2	E81500	377	CALL	SETMSK
03A5	B04C	378	MOV	AL,FIGS ;SET DIRECTION AND NUMBER AND WORDS
03A7	E86900	379	CALL	OUTCMD
03AA	B002	380	MOV	AL,2 ;DIR = 2
03AC	E86A00	381	CALL	OUTPAR
03AF	B0FF	382	MOV	AL,OFFH ;LENGTH 16K WORD
03B1	E86500	383	CALL	OUTPAR
03B4	B03F	384	MOV	AL,3FH ;HIGH BYTE
03B6	E86000	385	CALL	OUTPAR
03B9	C3	386	RET	
		387		
03BA	B04A	388	SETMSK: MOV	AL,MASK_ ;SET MASK TO STORED VALUES
03BC	E85400	389	CALL	OUTCMD
03BF	B0FF	390	MOV	AL,OFFH ;LOAD LOW BYTE
03C1	E85500	391	CALL	OUTPAR
03C4	E85200	392	CALL	OUTPAR
03C7	C3	393	RET	
		394		
03C8	B049	395	CUHOM1: MOV	AL,CURS
03CA	E84400	396	CALL	OUTCMD ;CURSOR HOME
03CD	32CD	397	XOR	AL,AL
03CF	E84700	398	CALL	OUTPAR
03D2	E84400	399	CALL	OUTPAR
03D5	E84100	400	CALL	OUTPAR
03D8	C3	401	RET	
		402		
03D9	B049	403	SETCU: MOV	AL,CURS
03DB	E83500	404	CALL	OUTCMD ;SET CURSOR
03DE	A000F9	R 405	MOV	AL,CEADL ;LOW BYTE OF CURSOR
03E1	E83500	406	CALL	OUTPAR
03E4	A001F9	R 407	MOV	AL,CEADH ;HIGH BYTE OF CURSOR
03E7	E82F00	408	CALL	OUTPAR
03EA	A002F9	R 409	MOV	AL,CDAD ;POSITION OF PIXEL
03ED	E82900	410	CALL	OUTPAR
03F0	C3	411	RET	
		412		
03F1	B070	413	SCROL1: MOV	AL,PRAM+0
03F3	E81D00	414	CALL	OUTCMD ;SET START AND LENGTH OF
03F6	B90800	415	MOV	CX,8 ;PAGE 1 AND PAGE 2
03F9	801E03F9	R 416	LEA	BX,SP1
03FD	E82A00	417	CALL	SEHPAR
0400	C3	418	RET	
		419		
0401	33C0	420	INI10: XOR	AX,AX
0403	A303F9	R 421	MOV	SP1,AX
0406	A307F9	R 422	MOV	SP2,AX ;START OF PAGE 1 = 0
0409	A20AF9	R 423	MOV	LP22,AL ;LENGTH OF PAGE 2 = 0
040C	C60606F919	R 424	MOV	LP12,25 ;LENGTH OF PAGE 1 = 25
0411	EBDE	425	JMP	SCROL1
		426		
		427		
		428	; *** SUBROUTINES OF GDC ***	
		429	;	*****
		430		
0413	E80900	431	OUTCMD: CALL	FIFRDY ;PUT A COMMAND IN FIFO
0416	E6A1	432	OUT	GDCCON,AL
0418	C3	433	RET	
		434		
0419	E80300	435	OUTPAR: CALL	FIFRDY ;PUT A PARAMETER TO FIFO

BOOTER_ID_DRIVER

LOC	OBJ	LINE	SOURCE	
041C	E6A0	436	OUT	GDCPAR.AL
041E	C3	437	RET	
		438		
041F	9C	439	FIFRDY: PUSH	
0420	50	440	PUSH	AX
0421	E4A0	441	FIF01D: IN	AL,GDCSTA ;READ GDC-STATUS
0423	24D2	442	AND	AL,FIFULL
0425	75FA	443	JNZ	FIF01D ;JUMP IF FIFO FULL
0427	58	444	POP	AX
0428	9D	445	POPF	
0429	C3	446	RET	
		447		
		448		
042A		449	SEMPAR:	
042A	8A07	450	MOV	AL,[BX] ;LOAD PARAMETER
042C	E8EAF7	451	CALL	OUTPAR ;OUTPUT PARAMETER
042F	43	452	INC	BX
0430	E2F8	453	LOOP	SEMPAR ;LOOP UNTIL END OF PARAMETER-LIST
0432	C3	454	RET	
		455		
		456		
		457	;*****	
		458	;	
		459	; SUBROUTINES	
		460	;	
		461	;*****	
		462		
		463		
		464	; *** DISPLAY ROUTINE ***	
		465	; *****	
		466		
0433	2E8A2F	467	DISPL: MOV	CH,CS:[BX]
0436	43	468	L_13: INC	BX
0437	2E8A07	469	MOV	AL,CS:[BX]
043A	E826FE	470	CALL	MODCRT
043D	FEC0	471	DEC	CH
043F	75F5	472	JNZ	L_13
0441	43	473	INC	BX
0442	C3	474	RET	
		475		
0443	8A07	476	DISPL1: MOV	AL,[BX]
0445	E81BFE	477	CALL	MODCRT
0448	43	478	INC	BX
0449	E2F8	479	LOOP	DISPL1
044B	C3	480	RET	
		481		
044C		482	DISPLAY PROC FAR	
		483		
044C	8A0F	484	MOV	CL,[BX]
044E	32ED	485	XOR	CH,CH
0450	43	486	DISP_N: INC	BX
0451	8A07	487	MOV	AL,[BX]
0453	E80DFE	488	CALL	MODCRT
0456	E2F8	489	LOOP	DISP_N
0458	C8	490	RET	
		491		
		492	DISPLAY ENDP	
		493		

BOOTER_IO_DRIVER

XREF SYMBOL TABLE LISTING

NAME	TYPE	VALUE	ATTRIBUTES, XREFS
??SEG	SEGMENT		SIZE=0000H PARA PUBLIC
BOOTER.	SEGMENT		SIZE=0459H BYTE COMMON 'CODE' 64# 67 494
BUFFER.	V WORD	0000H	DATA 30#
BUFFSTA.	V BYTE	F830H	DATA 40#
CALCH1.	L NEAR	031AH	BOOTER 290 298#
CALCH2.	L NEAR	0312H	BOOTER 293# 304 306
CALCHR.	L NEAR	0306H	BOOTER PUBLIC 61 264 288#
CAR1.	L NEAR	0296H	BOOTER 197 202#
CARRET.	L NEAR	0281H	BOOTER 186 193#
CDAD.	V BYTE	F902H	DATA 46# 409
CEADH.	V BYTE	F901H	DATA 45# 407
CEADL.	V BYTE	F900H	DATA 44# 295 336 350 405
CHA1.	L NEAR	0282H	BOOTER 221 224#
CHA2.	L NEAR	028EH	BOOTER 225 230#
CHARA.	NUMBER	0050H	137#
CHP.	L NEAR	029BH	BOOTER 188 208# 215
CLRL.	L NEAR	033AH	BOOTER 315 321# 349
CUHOM1.	L NEAR	03CBH	BOOTER 376 395#
CURS.	NUMBER	0049H	156# 395 403
CURSX.	V BYTE	F800H	DATA 35# 193 209 219 232
CURSY.	V BYTE	F801H	DATA 36# 194 202
DATA.	SEGMENT		SIZE=F90CH WORD COMMON 'DATA' 28# 55 67 67 67
DISP_H.	L NEAR	0450H	BOOTER 486# 489
DISPL.	L NEAR	0433H	BOOTER PUBLIC 61 467#
DISPL1.	L NEAR	0443H	BOOTER PUBLIC 61 476# 479
DISPLAY.	L FAR	044CH	BOOTER PUBLIC 62 482# 492
FIF010.	L NEAR	0421H	BOOTER 441# 443
FIFRDY.	L NEAR	041FH	BOOTER 431 435 439#
FIFULL.	NUMBER	0002H	150# 442
FIGS.	NUMBER	004CH	158# 280 340 378
GDCCON.	NUMBER	00A1H	143# 432
GDCPAR.	NUMBER	00A0H	145# 436
GDCSTA.	NUMBER	00A0H	144# 441
GETRE.	L NEAR	02CAH	BOOTER 184 203 210 237#
HOME.	NUMBER	0001H	124# 187
INIT10.	L NEAR	0401H	BOOTER 358 420#
INIT1A.	L NEAR	038EH	BOOTER 316 369#
INVLFG.	V BYTE	F90BH	DATA 53# 271
KEYDAT.	NUMBER	0040H	90# 97 104
KEYIMP.	L NEAR	0250H	BOOTER PUBLIC 61 94# 96
KEYSTA.	NUMBER	0041H	91# 94 100
KYLOOK.	L NEAR	0259H	BOOTER PUBLIC 61 100#
L_1.	L NEAR	02A0H	BOOTER 102 104#
L_13.	L NEAR	0436H	BOOTER PUBLIC 61 468# 472
L_31.	L NEAR	0385H	BOOTER 357 359#
LP11.	V BYTE	F905H	DATA 48#
LP12.	V BYTE	F906H	DATA 49# 355 360 373 424
LP21.	V BYTE	F909H	DATA 51#
LP22.	V BYTE	F90AH	DATA 52# 362 372 423
M.	V BYTE	0000H	DATA 31#
MASK_.	NUMBER	004AH	157# 388
NCLSCR.	L NEAR	032DH	BOOTER 214 314#
NOCRT.	L NEAR	0263H	BOOTER PUBLIC 61 169# 470 477 488
NOREPL.	NUMBER	0000H	164# 267 321
NEWL.	NUMBER	0088H	123# 185
NOMOV.	NUMBER	0000H	120# 183
OUTCMD.	L NEAR	0413H	BOOTER 268 281 322 341 379 389 396 404 414 431#
OUTPAR.	L NEAR	0419H	BOOTER 270 272 283 285 324 326 343 345 347 381 383 385 391 392 398 399 400 406 408 410 435#

BOOTER_IO_DRIVER

NAME	TYPE	VALUE	ATTRIBUTES, XREFS
PRAM .	NUMBER	0070H	159# 413
PROMPT .	NUMBER	002AH	121#
RAMSEL .	NUMBER	0010H	75#
RAMSEL .	NUMBER	0011H	76#
RUBOUT .	NUMBER	008BH	126#
SCERA .	L HEAR	02A2H	BOOTER 190 214#
SCRER .	NUMBER	000CH	125# 189
SCROL1 .	L HEAR	03F1H	BOOTER 317 363 413# 425
SCROLN .	L HEAR	034AH	BOOTER 199 227 335#
SEMPAR .	L HEAR	042AH	BOOTER 417 449# 453
SETCCA .	L HEAR	02F3H	BOOTER 266 279#
SETCHA .	L HEAR	02A7H	BOOTER 182 191 219#
SETCR .	L HEAR	039FH	BOOTER 314 376#
SETCU .	L HEAR	0309H	BOOTER PUBLIC 61 265 337 351 403#
SETNSK .	L HEAR	038AH	BOOTER 279 338 377 388#
SP1 . .	V WORD	F903H	DATA 47# 289 301 335 353 370 416 421
SP2 . .	V WORD	F907H	DATA 50# 371 422
SPACE .	NUMBER	0020H	122#
START .	NUMBER	006BH	155#
TYWORD .	NUMBER	0000H	162# 267 321
WDAT . .	NUMBER	0020H	160# 267 321
WRMCHR .	L HEAR	0203H	BOOTER 235 259#

END OF SYMBOL TABLE LISTING

We Welcome Your Evaluations Of Our Publications. Please Answer This Questionnaire And Mail It To Us.

Thank You



NCR DECISION MATE V SYSTEM TECHNICAL MANUAL

● How Do You Rate This Document?

Excellent (Needs No Improvement)

Good (Could Use Minor Improvements)

Fair (Should Have Major Improvements)

Poor (Should Be Completely Changed)

<u>Clarity</u>	<u>Accuracy</u>	<u>Organi- zation</u>	<u>Complete- ness</u>	<u>Usability</u>
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

● How Often Do You Use This Document?

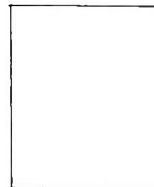
Often _____ Occasionally _____ Rarely _____

Comments:

Your Name: _____

Position: _____

NCR GmbH
Technical Publications Department
Ulmer Strasse 160A
8900 Augsburg
Federal Republic of Germany



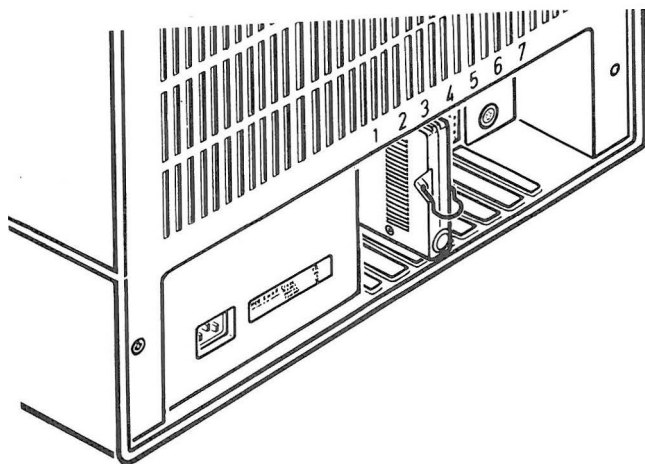
NCR DECISION MATE V**LEERADAPTER
(K214-V001)**

Das beiliegende Blatt erklärt Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

NCR ist ständig bemüht, die Produkte im Zuge der Entwicklung von Technologie, Bauteilen, Soft- und Firmware dem neuesten Stand anzupassen. NCR behält sich deshalb das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern. Nicht alle hier beschriebenen Leistungen werden von NCR in allen Teilen der Welt vertrieben. Nähere Informationen bezüglich eventueller Einschränkungen oder Erweiterungen sowie den aktuellen Stand erfahren Sie von Ihrem Händler oder der nächstgelegenen NCR-Geschäftsstelle.

**LEERADAPTER
(K214-V001)**

Dieser Adapter enthält eine leere Platine und Stifte für alle Anschlußmöglichkeiten einer beliebigen Steckfassung an der Rückseite Ihres NCR DECISION MATE V.



KIT INSTALLATION

for

BLANK INTERFACE

(K214-V001)

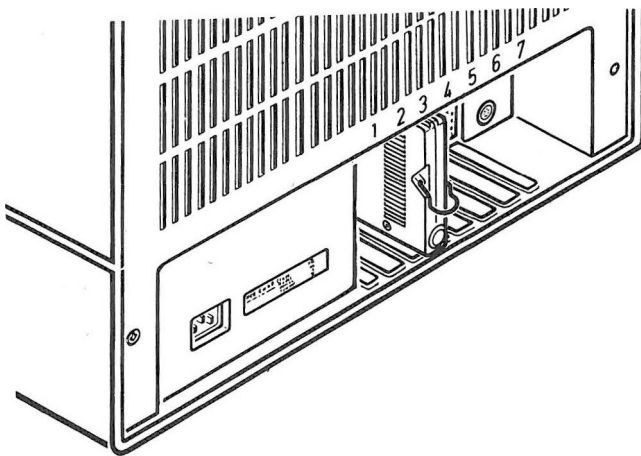
The attached pages provide information for installing this kit into an NCR DECISION MATE V. These pages should be filed in your NCR DECISION MATE V User Information Manual.

It is the policy of NCR Corporation to improve products as new technology, components, software, and firmware become available. NCR Corporation, therefore, reserves the right to change specifications without prior notice.

All features, functions, and operations described herein may not be marketed by NCR in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult your NCR representative or NCR office for information that is applicable and current.

KIT INSTALLATION**BLANK INTERFACE****(K 214-V001)**

1. The interface provided by this kit may be put into a vacant slot at the rear of the NCR DECISION MATE V (slots 2 to 6 may be used).
2. Check with the supplier of the interface for any further installation procedures.



NCR DECISION MATE V

**IEEE-488 ADAPTER
(K804-V001)**

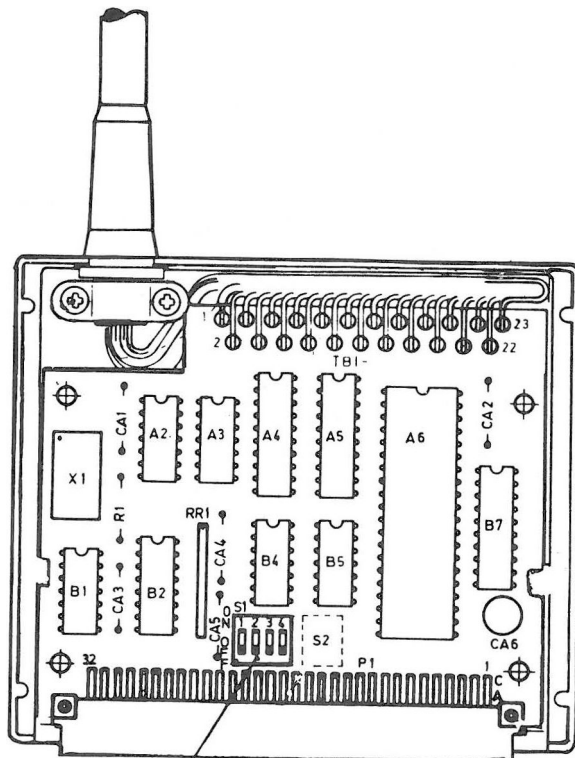
Die beiliegenden Seiten zeigen Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

NCR ist ständig bemüht, die Produkte im Zuge der Entwicklung von Technologie, Bauteilen, Soft- und Firmware dem neuesten Stand anzupassen. NCR behält sich deshalb das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern. Nicht alle hier beschriebenen Leistungen werden von NCR in allen Teilen der Welt vertrieben. Nähere Informationen bezüglich eventueller Einschränkungen oder Erweiterungen sowie den aktuellen Stand erfahren Sie von Ihrem Händler oder der nächstgelegenen NCR-Geschäftsstelle.

IEEE-488 ADAPTER (K804-V001)

Dieser Adapter erlaubt den Anschluß von Peripheriegeräten die einen IEEE-488 Bus verwenden. Der IEEE-488 Bus muß der Norm "IEEE-488 Standard Digital Interface for Programmable Instrumentation ANSI/IEEE Std 488-1978, 488A-1980" genügen.

1. Die Standardvoreinstellung für die IFSEL-Schalter ist 2B. Falls Ihr System diese Einstellung erwartet, können Sie bei 3. weiterlesen.
2. Um die Einstellung der IFSEL-Schalter zu verändern, müssen Sie die vier Gehäuseschrauben lösen und die Gehäusehälfte abnehmen. Stellen Sie anhand der nachstehenden Tabelle die Schalter auf den von Ihnen gewünschten Wert ein.



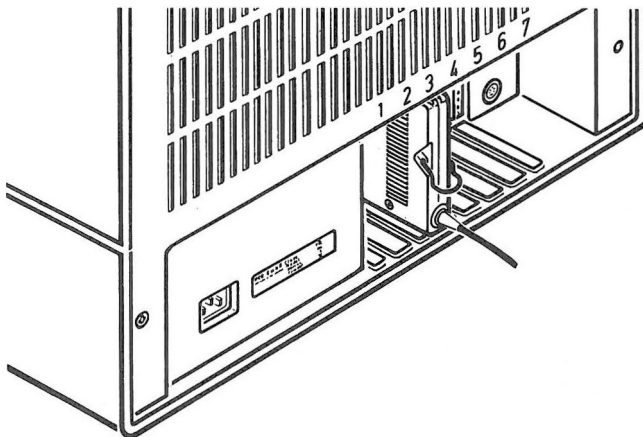
IFSEL - Schalter

IFSEL	SCHALTER 4 2 1 A	PORT-ADRESSE
0A	o o o ●	60H - 67H
0B	o o o o	68H - 6FH
1A	o o ● ●	70H - 77H
1B	o o ● o	78H - 7FH
2A	o ● o ●	30H - 37H
2B	o ● o o	38H - 3FH
3A	o ● ● ●	BOH - B7H
3B	o ● ● o	B8H - BFH
4A	● o o ●	COH - C7H
4B	● o o o	C8H - CFH

o = offen ● = geschlossen

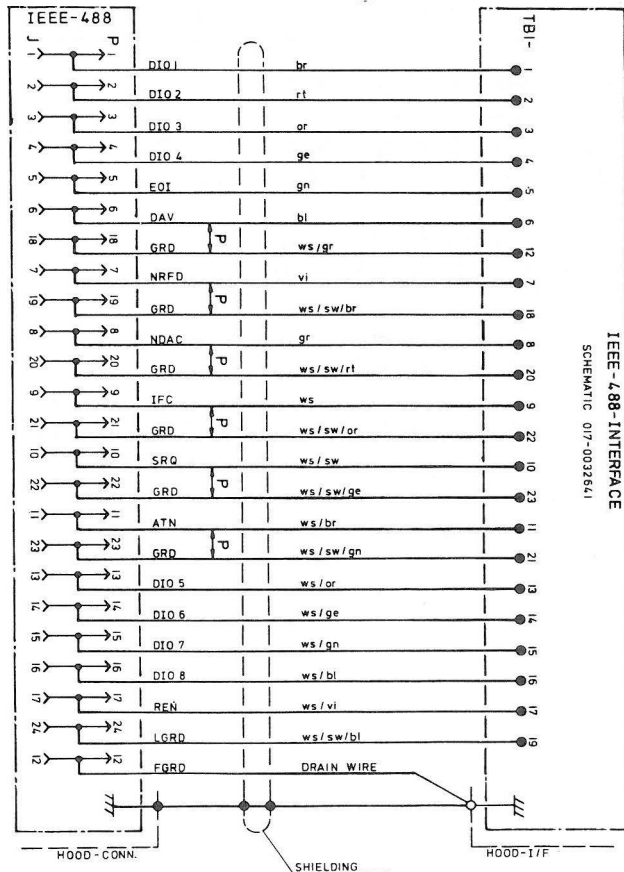
Für die Einstellung der Schalter ist der Aufkleber in der Gehäusenhälfte maßgeblich. Der Aufdruck auf den Schaltern ist nicht gültig.

3. Bauen Sie den Adapter wieder zusammen und schließen Sie ihn an einen der Steckplätze 2 bis 6 Ihres NCR DECISION MATE V an.



4. Schließen Sie den Stecker am Kabelende an das IEEE-488 kompatible Gerät an. Falls Sie hierzu zusätzliche Kabe verwenden, sollten Sie darauf achten, daß diese der Nor entsprechen.
5. Prüfen Sie anhand der Dokumentation Ihres Peripheriege rätés, ob ggf. weitere Anpassungen erforderlich sind (Talker/Listener Adresse).
6. Anhand der Beschreibung Ihrer IEEE-488-Software-Unter stützung können Sie deren Anpassung vornehmen.

Nachstehend finden Sie den Schaltplan des IEEE-488 Kabel und eine Übersicht der verwendeten Signale.



Die Übertragungsleitungen vom IEEE-488 Bus sind in drei Gruppen aufgeteilt:

- Datenbus, 8 Übertragungsleitungen
- Datenübertragungs-Steuer-Bus, 3 Übertragungsleitungen
- Allgemeiner Schnittstellen-Steuer-Bus, 5 Übertragungsleitungen

<u>Datenbus</u>	
DIO1 - DIO8	Die DIO-Leitungen übertragen Daten in paralleler-Bit und serieller-Byte Weise (synchronisiert). Grundsätzlich sind beide Übertragungsrichtungen (E/A) möglich.
<u>Datenübertragungs-Steuer-Bus</u>	
DAV	"Data Valid" zeigt das Vorhandensein und die Gültigkeit der an den DIO Leitungen anstehenden Information an.
NRFD	"Not Ready For Data" zeigt den Status der Datenempfängsbereitschaft angeschlossener Geräte an.
NDAC	"Not Data Accepted" zeigt den Status des Datenempfangs angeschlossener Geräte an.
Die Leitungen DAV, NRFD und NDAC arbeiten in einem verschränkten Drei-Draht-Quittungsbetrieb, um jedes Datenbyte zu übertragen.	

IEEE-488 Bus Signale

<u>Allgemeiner Schnittstellen-Steuer-Bus</u>	
ATN	"Attention" wird von einem "Controller" benutzt, um zu beschreiben, wie die Signale an den DIO Leitungen zu interpretieren sind und welche Geräte angesprochen sind.
IFC	"Interface Clear" wird von einem "Controller" benutzt, um das gesamte Schnittstellensystem (einige Teile hiervon befinden sich in den angeschlossenen Geräten) in einen definierten Ruhezustand zu versetzen.
SRQ	"Service Request" wird von einem Gerät verwendet, um den "Controller" auf sich aufmerksam zu machen und eine Unterbrechung der laufenden Ereignisfolge zu verlangen.
REN	"Remote Enable" wird von einem "Controller" in Verbindung mit anderen Nachrichten verwendet, um zwischen zwei alternativen Datenquellen zur Geräteprogrammierung zu wählen.
EOI	"End Or Identify" wird von einem "Talker" verwendet, um das Ende einer vielfachen Byte-Übertragungsfolge anzuzeigen, oder um in Verbindung mit ATN eine Aufruffolge durchzuführen.

IEEE-488 Bus Signale (Fortsetzung)

KIT INSTALLATION

for

IEEE-488 ADAPTER

(K804-V001)

The attached pages provide information for installing this kit into an NCR DECISION MATE V. These pages should be filed in your NCR DECISION MATE V User Information Manual.

**FEDERAL COMMUNICATIONS COMMISSION (FCC)
RADIO FREQUENCY INTERFERENCE STATEMENT**

WARNING

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Information to User

This equipment must be installed and used in strict accordance with the manufacturer's instructions. However, there is no guarantee that interference to radio communications will not occur in a particular commercial installation. If this equipment does cause interference, which can be determined by turning the equipment off and on, the user is encouraged to consult an NCR service representative immediately.

The NCR Corporation (NCR) is not responsible for any radio or television interference caused by unauthorized modification of this equipment or the substitution or attachment of connecting cables and equipment other than those specified by NCR. The correction of interferences caused by such unauthorized modification, substitution or attachment will be the responsibility of the user.

This equipment is designed for use in a commercial environment, not in a residential area. It fully complies with the relevant FCC regulations; when connected to any Class B equipment then the total product is reclassified accordingly.

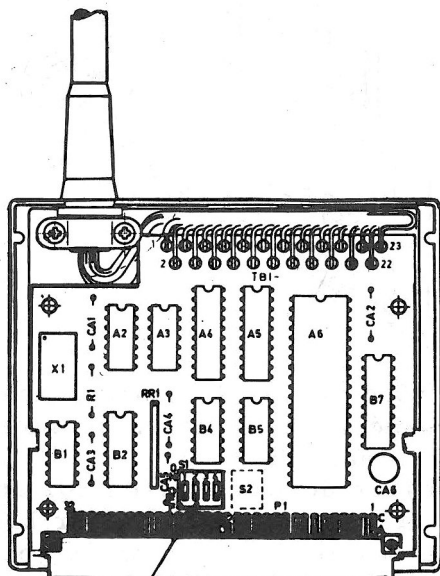
It is the policy of NCR Corporation to improve products as new technology, components, software, and firmware become available. NCR Corporation, therefore, reserves the right to change specifications without prior notice.

All features, functions, and operations described herein may not be marketed by NCR in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult your NCR representative or NCR office for information that

KIT INSTALLATION**IEEE-488 ADAPTER
(K804-V001)**

This kit is designed for use with peripheral devices using the IEEE-488 bus, according to the IEEE-488 Standard Digital Interface for Programmable Instrumentation ANSI/IEEE Std 488-1978, 488A-1980.

1. The standard setting (default value) for the IFSEL switches is 2B. If your system is using this standard setting, continue at Step 3.
2. To change the IFSEL switch setting, remove the four screws and cover from the adapter. Set the IFSEL switches to the desired configuration; the position of the switch and the possible settings are shown in the following figures.



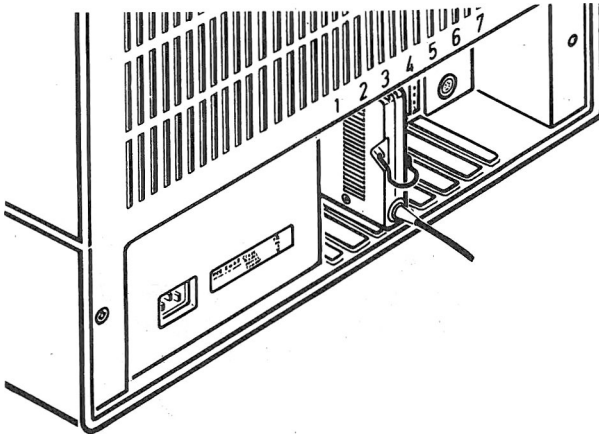
IFSEL Switches

IFSEL	Switch				Port Addresses
	4	2	1	A	
0A	○	○	○	●	60-67 Hex
0B	○	○	○	○	68-6F Hex
1A	○	○	●	●	70-77 Hex
1B	○	○	○	○	78-7F Hex
2A	○	●	○	●	30-37 Hex
2B	○	●	○	○	38-3F Hex
3A	○	○	●	●	80-87 Hex
3B	○	●	○	○	88-8F Hex
4A	●	○	○	●	C0-C7 Hex
4B	●	○	○	○	C8-CF Hex

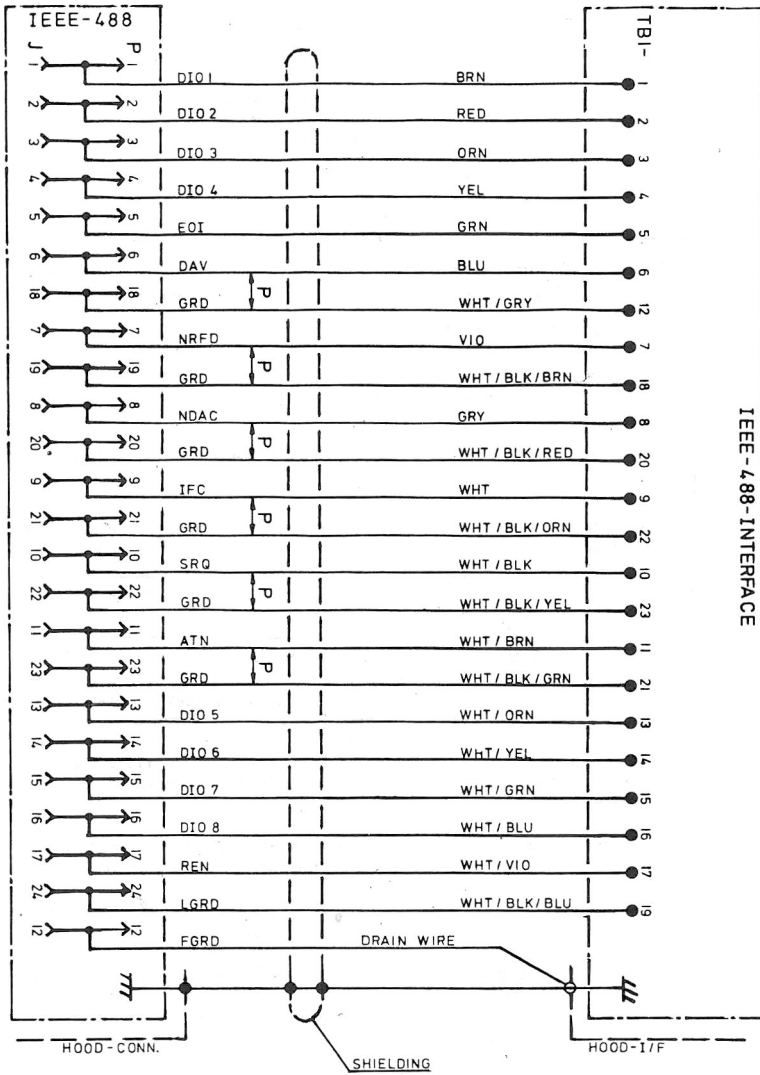
○ = Open ● = Closed

Follow instructions inside adapter cover:
ignore any markings on switch assembly

3. Install the IEEE-488 adapter into a vacant slot at the rear of the NCR DECISION MATE V (slots 2 to 6 may be used)



4. Connect the plug to the IEEE-488 compatible device; if you are using additional cables be sure that they match the IEEE-488 standards.
5. Check the device documentation for any strapping or switch setting requirements (Talker/Listener address).
6. Refer to the CONFIG description in your IEEE-488 Software Support Package for software installation procedures.



IEEE Cable

The bus is organized into three sets of signal lines:

- Data bus, 8 signal lines
- Data byte transfer control bus, 3 signal lines
- General interface management bus, 5 signal lines

Data Bus

DIO1 - DIO8

Data Input Output message bytes are carried on the DIO signal lines in a bit-parallel byte serial form, synchronously, and generally in a bidirectional manner.

Data Byte Transfer Control Bus

DAV

Data Valid is used to indicate the availability and validity of information on the DIO signal lines.

NRFD

Not Ready For Data is used to indicate the condition of readiness of devices to accept data.

NDAC

Not Data Accepted is used to indicate the condition of acceptance of data by devices.

The DAV NRFD and NDAC signal lines operate in a three wire (interlocked) handshake process to transfer each data byte across the interface.

General Interface Management Bus

ATN

Attention is used by a controller to specify how data on the DIO signal lines is to be interpreted and which devices must respond to the data.

IFC

Interface Clear is used by a controller to place the interface system (portions of which are contained in all interconnected devices) in a known quiescent state.

SRQ

Service Request is used by a device to indicate the need for attention, and to request an interruption of the current sequence of events.

REN

Remote Enable is used by a controller in conjunction with other messages to select between two alternate sources of device programming data.

EOI

End Or Identify is used by the talker to indicate the end of a multiple byte transfer sequence or, in conjunction with ATN to execute a polling sequence.

IEEE - 488 Bus Signals

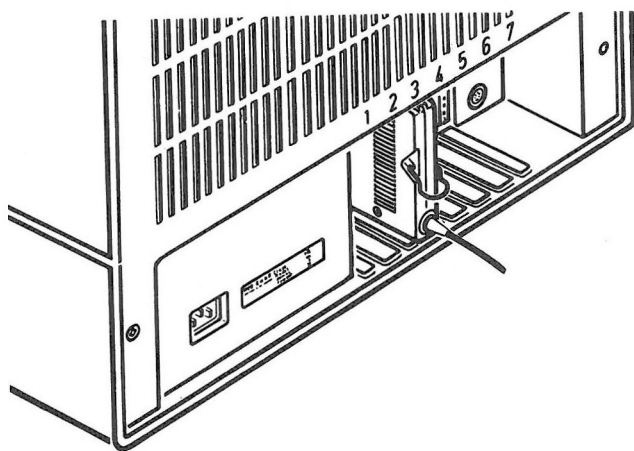
NCR DECISION MATE V**CENTRONICS-ADAPTER
(K210-V001)**

Die beiliegenden Seiten zeigen Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

NCR ist ständig bemüht, die Produkte im Zuge der Entwicklung von Technologie, Bauteilen, Soft- und Firmware dem neuesten Stand anzupassen. NCR behält sich deshalb das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern. Nicht alle hier beschriebenen Leistungen werden von NCR in allen Teilen der Welt vertrieben. Nähere Informationen bezüglich eventueller Einschränkungen oder Erweiterungen sowie den aktuellen Stand erfahren Sie von Ihrem Händler oder der nächstgelegenen NCR-Geschäftsstelle.

CENTRONICS-ADAPTER (K210-V001)

1. Setzen Sie diesen Adapter in eine der an der Rückseite Ihres NCR DECISION MATE V befindlichen Fassungen (2...6) ein.

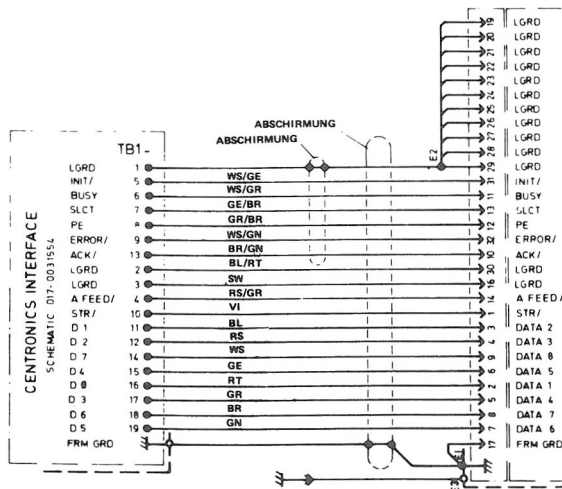


2. Schließen Sie den am Kabelende vorhandenen Stecker an einen Centronics-kompatiblen Drucker an.
3. Das zu Ihrem Drucker gehörende Handbuch beschreibt die Handhabung von Druckpapier und Farbband.
4. Schließen Sie den Drucker an die erforderliche Netzspannung an.

WICHTIG: Der gleichzeitige Betrieb dieses Adapters mit dem Adapter K212 oder K213 ist nicht möglich.

Umseitig finden Sie eine Übersicht der im Rahmen des Übertragungsprotokolls benutzten Leitungen.

STROBE/	Taktfrequenz zur Druckersynchronisierung. Mindesttaktzeit 1 µsec. Empfang erfolgt während Signal-"low".
ACKNLG/	Signal (2.5 bis 10 µsec) zur Datenempfangsbestätigung. Empfang neuer Daten vor Rücksetzung dieses Signals unmöglich.
BUSY	Datenannahme nicht möglich aus einem der folgenden Gründe : - während des Datenempfangs (je nach Drucker) - während des Druckvorgangs - im Off-Line-Zustand - beim Vorliegen eines Fehlerzustandes - während des Zeilenvorschubs
DATA 1-8	Parallele Datenübertragung ("High" = log 1; "Low" = log 0)
PE	"High" zeigt das Papierende an.
SLCT	Signal zeigt Wahlzustand des Druckers an (je nach Drucker)
AUTO FEED/	Automatischer Zeilenvorschub (nicht bei allen Druckern).
INIT/	Computersignal zur Druckerrückstellung (je nach Drucker).
ERROR/	Fehlerzustand des Druckers (z.B. Papierende, Off-Line, o.ä).
TEST 1-4	Eventuell vorhandene Prüfleitungen.



KIT INSTALLATION
for
CENTRONICS ADAPTER
(K210-V001)

The attached pages provide information for installing this kit into an NCR DECISION MATE V. These pages should be file in your NCR DECISION MATE V User Information Manual.

FEDERAL COMMUNICATIONS COMMISSION (FCC)
RADIO FREQUENCY INTERFERENCE STATEMENT

WARNING

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminal printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

Information to User

This equipment generates and uses radio frequency energy and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of 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 television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna
- Relocate the computer with respect to the receiver
- Move the computer away from the receiver
- Plug the computer into a different outlet so that computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful: "How to Identify and Resolve Radio-TV Interference Problems". This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, Stock No. 004-000-00345-4.

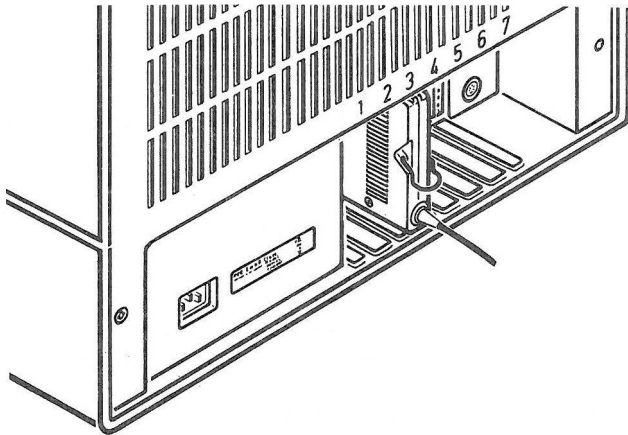
The NCR Corporation (NCR) is not responsible for any radio or television interference caused by unauthorized modifications of this equipment or the substitution or attachment of connecting cable and equipment other than those specified by NCR. The correction of interferences caused by such unauthorized modification, substitution or attachment will be the responsibility of the user.

It is the policy of NCR Corporation to improve products as new technology, components, software and firmware become available. NCR Corporation, therefore, reserves the right to change specifications without prior notice.

All features, functions, and operations described herein may not be marketed by NCR in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult your NCR representative or NCR office for information that is applicable and current.

KIT INSTALLATION**CENTRONICS ADAPTER
(K210-V001)**

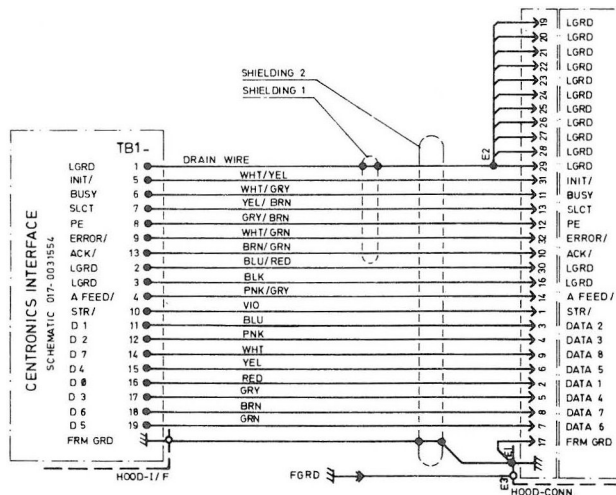
1. Install this kit into a vacant slot at the rear of the NCR DECISION MATE V (slots 2 to 6 may be used).
2. Connect the plug to the Centronics compatible printer.



3. Check the printer documentation for ribbon installation and paper loading.

NOTE: When this kit is installed, then kits K212 and K213 cannot be installed at the same time.

STROBE/	Pulse is used to clock read data in. The pulse width must be more than 1 μ sec at receiving printer. Read in of data is performed at the "low" level of this signal.
ACKNLG/	Indicates that data has been received by the printer. New data cannot be received before ACKNLG is reset. Pulse width 2.5 μ sec to 10 μ sec.
BUSY	When "HIGH" the printer indicates that it cannot receive data. The signal becomes "HIGH" in the following cases: <ul style="list-style-type: none"> - During data entry (depending on printer) - During printing operation - In off-line state - During printer error status - During line feed
DATA 1-8	Signals represent information of parallel data "HIGH" = log 1; "LOW" = log 0.
PE	"HIGH" signal indicates that the printer is out of paper.
SLCT	(Input) Printer is in selected mode (option).
AUTO FEED/	(Output) Auto line feed (option).
INIT/	(Output) Printer reset (option).
ERROR/	(Input) Indicates printer error (option) for example at paper end, off-line, safety switch open, failure status.
TEST 1-4	Optional loop test inputs.



NCR DECISION MATE V

16-Bit Microprocessor
with
Programmable Interrupt Controller (PIC) and
Memory Retention Facility
(K235 V001)

FEDERAL COMMUNICATIONS COMMISSION (FCC)
RADIO FREQUENCY INTERFERENCE STATEMENT

WARNING

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

Information to User

This equipment generates and uses radio frequency energy and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of 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 television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna
- Relocate the computer with respect to the receiver
- Move the computer away from the receiver
- Plug the computer into a different outlet so that computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful: "How to Identify and Resolve Radio-TV Interference Problems". This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, Stock No. 004-000-00345-4.

The NCR Corporation (NCR) is not responsible for any radio or television interference caused by unauthorized modifications of this equipment or the substitution or attachment of connecting cables and equipment other than those specified by NCR. The correction of interferences caused by such unauthorized modification, substitution or attachment will be the responsibility of the user.

It is the policy of NCR Corporation to improve products as new technology, components, software, and firmware become available. NCR Corporation, therefore, reserves the right to change specifications without prior notice.

All features, functions, and operations described herein may not be marketed by NCR in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult your NCR representative or NCR office for information that is applicable and current.

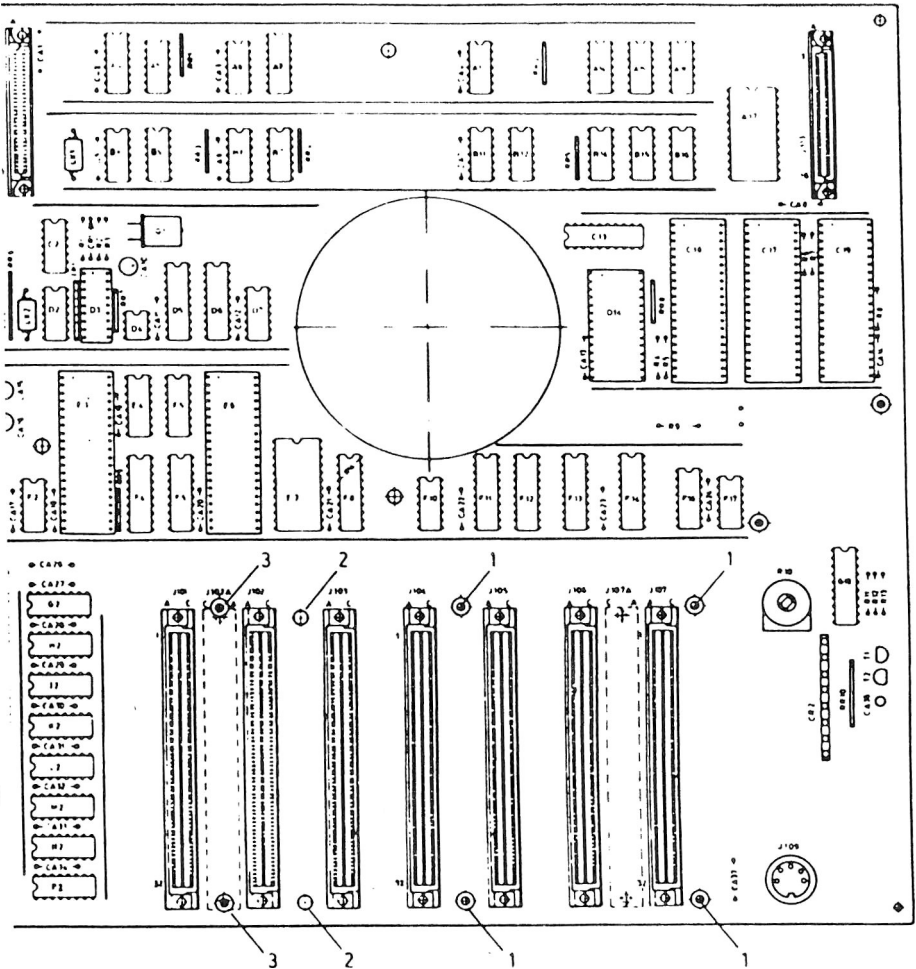


Figure 2: Controller PCB

- Screw
- Additional screws (see text)

2 -- Pre-drilled hole

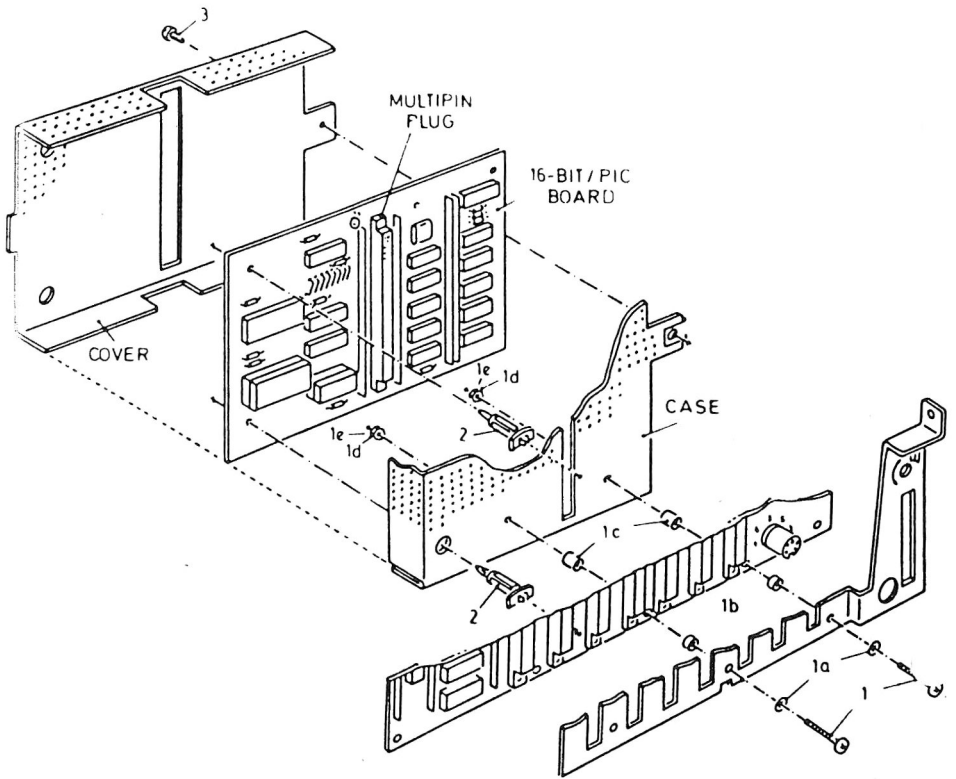


Figure 3: K235 V001 Assembly

- | | | | |
|----|------------------|---|-----------------------|
| 1 | -- Screw 3x16 | 2 | -- PVC stand-off clip |
| 1a | -- Washer | 3 | -- Screw 2.9x6.5 |
| 1b | -- Spacer washer | | |
| 1c | -- PVC spacer | | |
| 1d | -- Washer | | |
| 1e | -- Nut | | |

NCR DECISION MATE V

16-Bit Microprocessor with Programmable Interrupt Controller (PIC) and Memory Retention Facility (K235 V001)

This Kit can be installed in your NCR DECISION MATE V, provided that it is capable of executing the System Restart by means of the CONTROL-F20 key combination (refer to the User Information for the NCR DECISION MATE V).

CAUTION: Installation of this Kit requires knowledge of and experience in handling semiconductor components and other computer hardware. It should therefore be installed by a qualified engineer. The following installation procedures should be carried out only by such a person.

The 16-bit/PIC board contains two wire jumper switches (S1, S2). These must be set to the positions indicated in Figure 1, before installation of the Kit.

Installation of the 16-bit/PIC board on a multilayer board which already contains a 16-bit processor board:

1. Remove the multilayer controller printed circuit board from the computer. This procedure is described in the Service Manual for the NCR DECISION MATE V.
2. Refer to Figure 3 in this description and remove the screw marked 3. Remove the metal cover.

4. Refer to Figure 3. Using the longer screws (1) and the additional PVC spacers (1c) provided with the Kit, assemble the metal case onto the controller PCB.
5. Insert the 16-bit/PIC board into the multipin socket which is now protruding into the metal case. Press the multipin plug firmly into the socket. Press the 16-bit/PIC board downwards over the noses of the PVC stand-off clips.
6. Close the case with the metal cover provided and secure it by means of the screw (3) provided with the Kit.
7. Re-install the controller PCB and carry out a complete diagnostic check of the NCR DECISION MATE V.

3. Squeeze the noses of the two stand-off clips (2) to overcome their latching mechanism and pull the 16 bit PCB fractionally away from the controller PCB at these two corners. Then grip the edges of the 16-bit PCB at a position adjacent to the multipin plug and pull it away from the controller PCB.
4. Install the 16-bit/PIC board in place of the 16-bit PCB which you have just removed.
5. Re-install the controller PCB and carry out a complete diagnostic check of the NCR DECISION MATE V.

Installation of the 16-bit/PIC board on a multilayer controller board which has not previously contained a 16-bit PCB:

1. Remove the multilayer controller printed circuit board from the computer. This procedure is described in the Service Manual for the NCR DECISION MATE V.
2. Refer to Figure 2 which shows the multilayer controller PCB from the component side. Remove the four screws marked 1 in this Figure.

If the 96-pin socket J107A (supplied with the Kit) is not already present, additionally remove the screws marked 3. Then remove the metal plate containing the adapter retention clips, taking care not to misplace the spacer washers. Then solder the socket into place, so that the cut-out corners of the socket are on the right-hand side when viewing the solder side of the controller board with the socket in a bottom left position. Replace the metal plate with screws 3 and their spacer washers.

3. Insert the bases of the two PVC stand-off clips into the pre-drilled holes (2) from the solder side of the controller PCB.

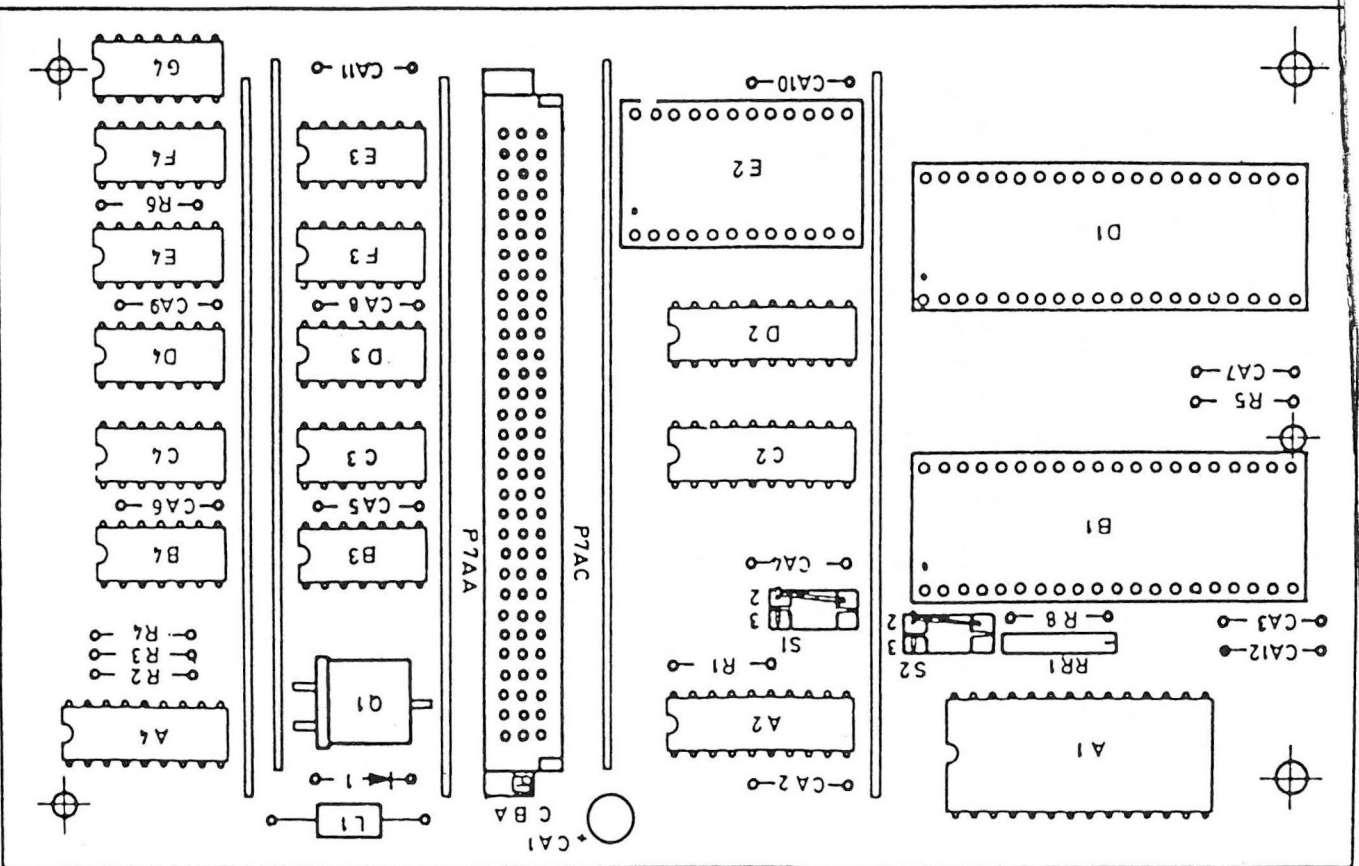


Figure 1: K235 V001 Component Layout

NCR DECISION MATE V

16-Bit Mikroprozessor

mit

Programmierbarem Interrupt Controller (PIC) und

Speicherschnelltester (K235 V001)

(Memory Retention Facility)

NCR ist ständig bemüht, die Produkte im Zuge der Entwicklung von Technologie, Bauteilen, Soft- und Firmware dem neuesten Stand anzupassen. NCR behält sich deshalb das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern. Nicht alle hier beschriebenen Leistungen werden von NCR in allen Teilen der Welt vertrieben. Nähere Informationen bezüglich eventueller Einschränkungen oder Erweiterungen sowie den aktuellen Stand erfahren Sie von Ihrem Händler oder der nächstgelegenen NCR-Geschäftsstelle.

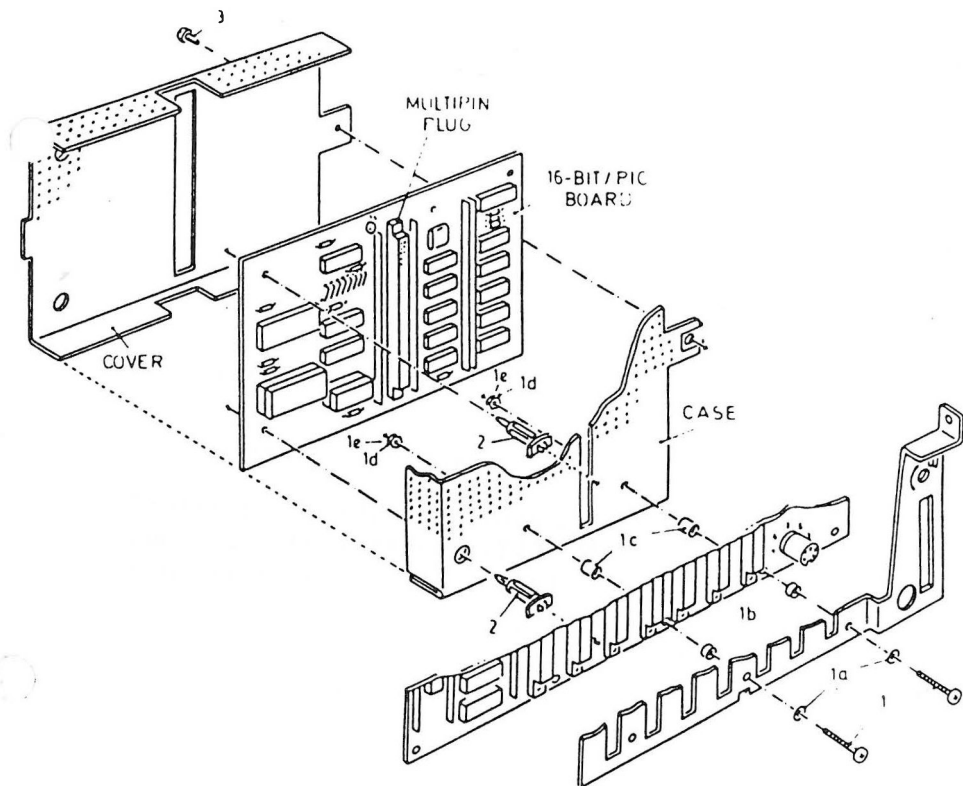


Bild 3: K235 V001 Einbau

- | | | | |
|----|---------------------|---|-------------------|
| 1 | Schraube 3x16 | 2 | PVC-Abstandhalter |
| 1a | Unterlegscheibe | 3 | Schraube 2,9x6,5 |
| 1b | Abstandsringe | | |
| 1c | PVC-Unterlegscheibe | | |
| 1d | Unterlegscheibe | | |
| 1e | Mutter | | |

NCR DECISION MATE V

16-Bit Mikroprozessor

mit

Programmierbarem Interrupt Controller (PIC) und
Speicherschnelltester (K235 V001)

Diese Leistungserweiterung kann in Ihren NCR DECISION MATE V nur eingebaut werden, wenn Sie den Systemneustart mit CONTROL-F20 Tastenkombination durchführen können (siehe Bedienungsanleitung für NCR DECISION MATE V).

VORSICHT: Für den Einbau dieser Erweiterung brauchen Sie Kenntnisse und Erfahrung im Umgang mit Halbleiter-Bauelementen und anderer Computerhardware. Überlassen Sie den Einbau einem qualifizierten Techniker.

Die 16-Bit/PIC-Platine enthält zwei Drahtbrücken (S1, S2). Vor dem Einbau der Leistungserweiterung müssen diese in den im Bild 1 gezeigten Stellungen stecken.

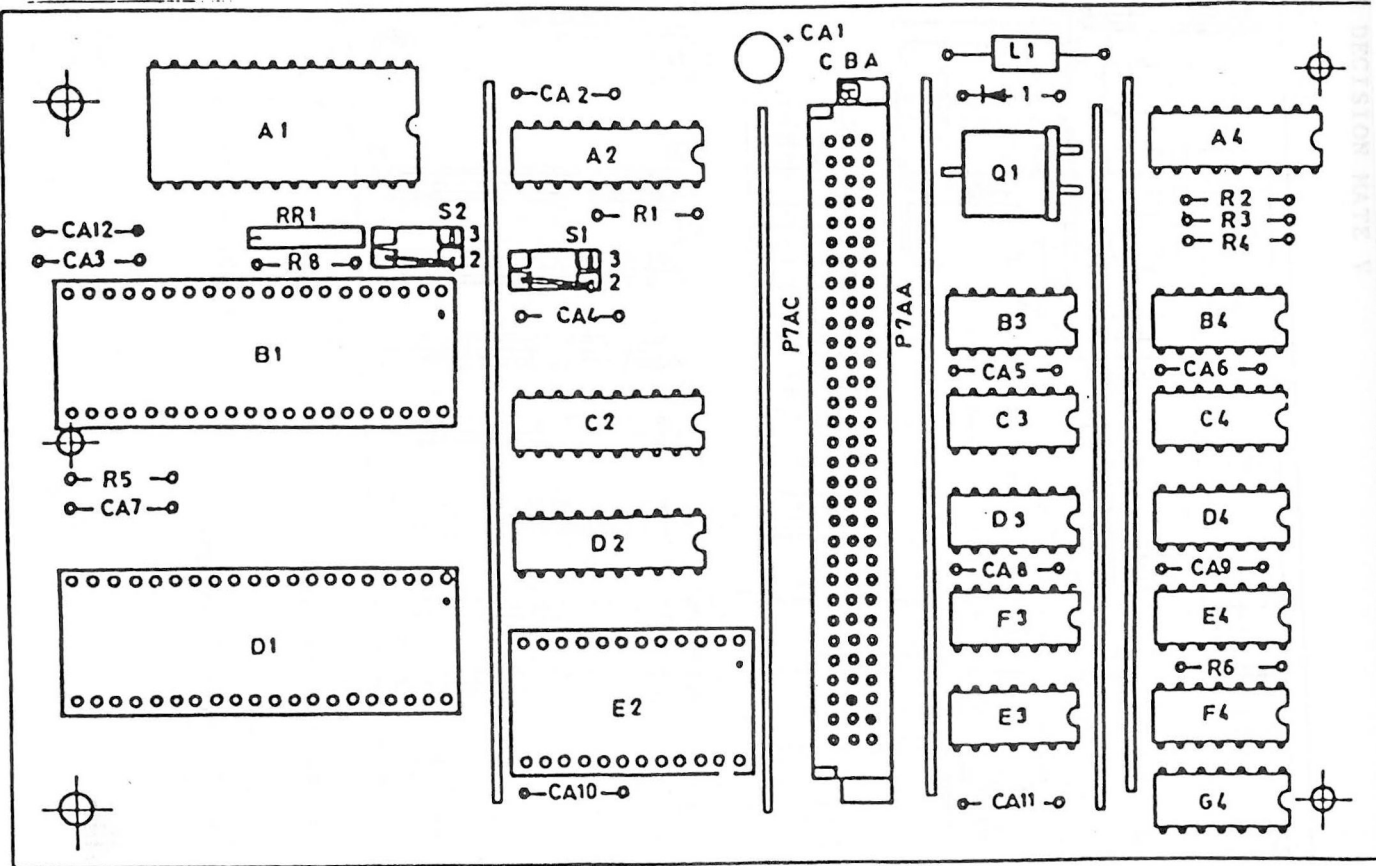
Einbau der 16-Bit/PIC-Platine auf eine Hauptplatine, auf der schon eine 16-Bit Prozessor-Platine montiert ist:

1. Entfernen Sie die Haupt- mit aufgesteckter Grafik-Platine aus dem Computer. Dieser Vorgang ist im Service Manual für den NCR DECISION MATE V beschrieben.
2. Halten Sie sich an das Bild 3 in dieser Beschreibung und entfernen Sie die Schraube 3. Nehmen Sie das Metallgehäuse ab.
3. Ziehen Sie die 16-Bit Platine von den zwei Abstandshaltern (2) und der mehrpoligen Buchse von der Haupt-Platine ab.
4. Ersetzen Sie die 16-Bit Platine durch die 16-Bit/PIC Platine.
5. Bauen Sie die Haupt- mit aufgesteckter Grafik-Platine wieder ein und machen Sie einen vollständigen Diagnose-Check des NCR DECISION MATE V.

Einbau der 16-Bit/PIC Platine auf eine Haupt-Platine, auf die vorher keine 16-bit Platine montiert war:

1. Entfernen Sie die Haupt- mit aufgesteckter Grafik-Platine aus dem Computer. Dieser Vorgang ist im Service Manual für den NCR DECISION MATE V beschrieben.
2. Halten Sie sich an das Bild 2. Es zeigt die Haupt-Platine von der bestückten Seite. Entfernen Sie die vier Schrauben 1.
Wenn die 96-polige Buchse J107A (wird mit der Leistungserweiterung geliefert) nicht schon montiert ist, entfernen Sie zusätzlich die Schrauben 3. Dann entfernen Sie die Metallplatte mit den Adapterhaltern. Achten Sie auf den richtigen Sitz der Unterlegscheiben. Dann löten Sie die Buchse so an, daß die ausgeschnittenen Ecken der Buchse auf der rechten Seite sind; betrachten Sie die Lötseite der Haupt-Platine mit der Buchse unten links. Montieren Sie die Metallplatte mit Schrauben 3 und ihren Unterlegscheiben.
3. Stecken Sie die Füße der zwei PVC-Abstandhalter in die vorgebohrten Löcher von der Lötseite der Haupt-Platine aus.
4. Betrachten Sie Bild 3. Mit den längeren Schrauben (1) und den zusätzlichen PVC-Unterlegscheiben (1c) (wird mit der Leistungserweiterung geliefert) befestigen Sie das Metallgehäuse auf der Haupt-Platine.
5. Stecken Sie die 16-Bit/PIC-Platine in die mehrpolige Buchse, die jetzt in das Metallgehäuse ragt. Drücken Sie die Steckerleiste fest in die Buchse.
Drücken Sie die 16-BIT/PIC Platine auf die Spitzen der PVC-Abstandhalter.
6. Schließen Sie das Gehäuse mit dem mitgelieferten Metalldeckel und sichern Sie es mit der Schraube 3.
7. Bauen Sie die Haupt- mit aufgesteckter Grafik-Platin wieder ein und machen Sie einen kompletten Diagnose-Check des NCR DECISION MATE V.

Bild 1: K235 V001 Bestückung



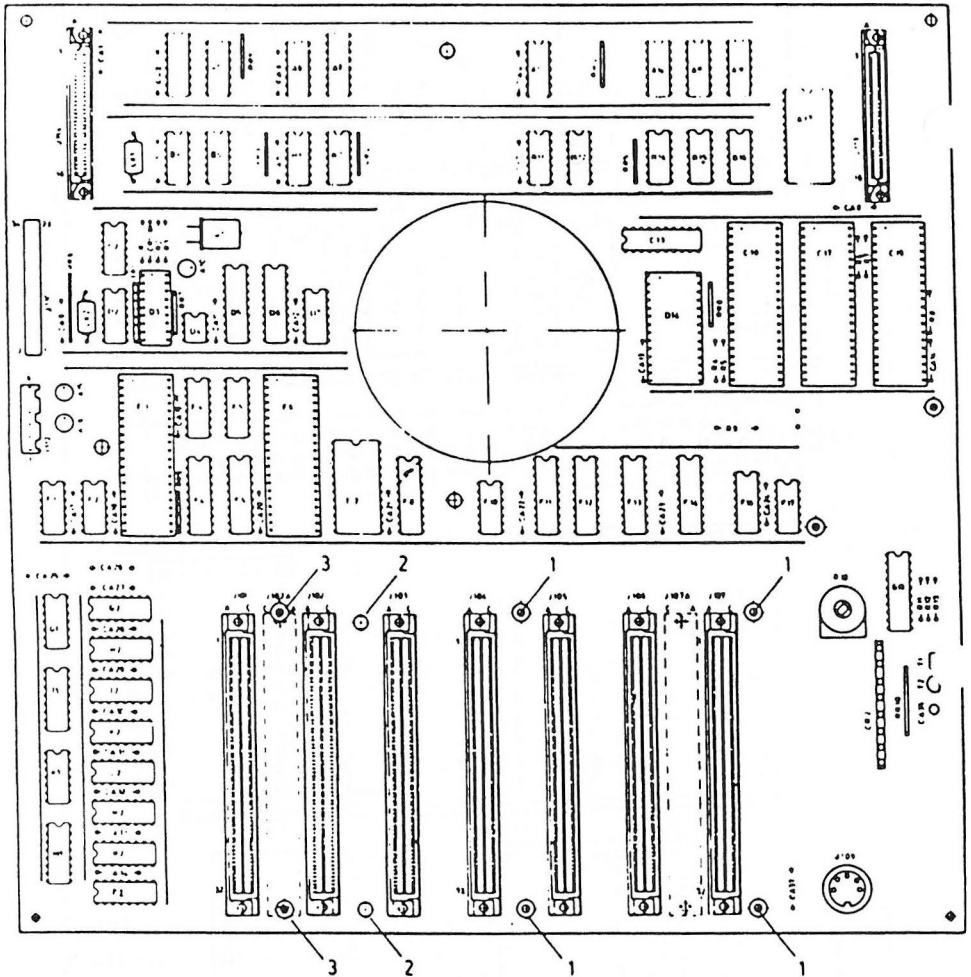


Bild 2: Haupt-Platine

- | | |
|-------------------------------------------------------------------|-----------------------------|
| <p>1 Schraube</p> <p>3 zusätzliche Schrauben
(siehe Text)</p> | <p>2 vorgebohrte Löcher</p> |
|-------------------------------------------------------------------|-----------------------------|

KIT INSTALLATION

for

**EXTERNAL 16-BIT PROCESSOR UPGRADE
(K 231)**

The attached pages provide information for installing this kit into an NCR DECISION MATE V. These pages should be filed in your NCR DECISION MATE V User Information Manual.

FEDERAL COMMUNICATIONS COMMISSION (FCC)
RADIO FREQUENCY INTERFERENCE STATEMENT

WARNING

This equipment has been certified to comply with the limits for a Class B computing device, pursuant to Subpart J of Part 15 of FCC Rules. Only peripherals (computer input/output devices, terminals, printers, etc.) certified to comply with the Class B limits may be attached to this computer. Operation with non-certified peripherals is likely to result in interference to radio and TV reception.

Information to User

This equipment generates and uses radio frequency energy and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of 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 television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna
- Relocate the computer with respect to the receiver
- Move the computer away from the receiver
- Plug the computer into a different outlet so that computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful: "How to Identify and Resolve Radio-TV Interference Problems". This booklet is available from the U.S. Government Printing Office, Washington, DC 20402, Stock No. 004-000-00345-4.

The NCR Corporation (NCR) is not responsible for any radio or television interference caused by unauthorized modifications of this equipment or the substitution or attachment of connecting cables and equipment other than those specified by NCR. The correction of interferences caused by such unauthorized modification, substitution or attachment will be the responsibility of the user.

It is the policy of NCR Corporation to improve products as new technology, components, software, and firmware become available. NCR Corporation, therefore, reserves the right to change specifications without prior notice.

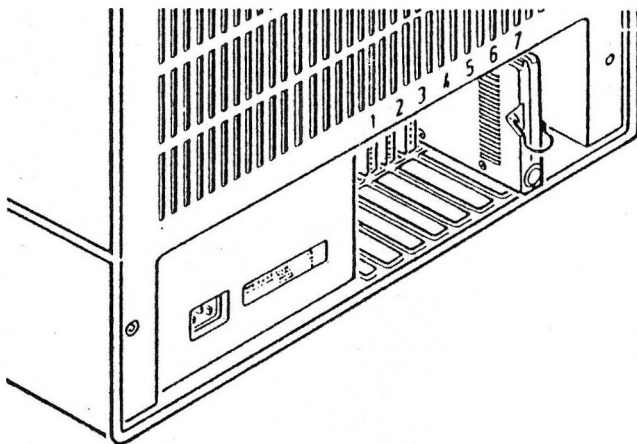
All features, functions, and operations described herein may not be marketed by NCR in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult your NCR representative or NCR office for information that is applicable and current.

KIT INSTALLATION**EXTERNAL 16-BIT PROCESSOR UPGRADE
(K 231)**

1. Install this kit into slot 7 at the rear of the NCR DECISION MATE V.

NOTE: No other position may be used for this kit.

Your system is now ready for the use of either 8-bit or 16-bit software.



NCR DECISION MATE V**MAUS-ADAPTER
(K806)**

Die beiliegenden Seiten zeigen Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

**MAUS-ADAPTER
(K806)**

INHALT

INBETRIEBNAHME

EINLEITUNG.....
IFSEL- UND MAUS-AUSWAHL.....

SOFTWARE

EINLEITUNG.....
BENUTZUNG DER PORT-ADRESSEN.....
FUNKTIONSPRÜFUNG.....
DAS STATUS-BYTE.....
BEFEHLE UND IHRE PARAMETER.....
INTERRUPT-BEHANDLUNG..... 1:

INBETRIEBNAHME**EINLEITUNG**

Mit Hilfe einer "Maus" lassen sich Handbewegungen auf einem Schreibtisch oder einer ähnlichen Fläche in Bildschirmgrafik umsetzen. Sie können folgende Mäuse in Verbindung mit dem Maus-Adapter benutzen:

- Hawley Mouse MARK II
- Alps Encoder-Mouse
- Mouse Systems Quad Mouse
- Depraz Souris P4
- Logitech LM-P-5
- Logitech P4

Die Verständigung zwischen einem Adapter und Ihrem NCR DECISION MATE V erfolgt über einen von insgesamt 10 Datenübertragungskanälen. Jeder Datenübertragungskanal (oder IFSEL: engl. InterFace SElect) verfügt über 8 Portadressen. Der Maus-Adapter kann grundsätzlich jede beliebige der 10 IFSEL-Nummern benutzen. Die für diesen Adapter vorhandene p-System-Software betrachtet die IFSEL-Nummer 2A (Portadressen 30H...37H) als Standardwert. Diese IFSEL-Nummer ist bereits bei der Lieferung des Maus-Adapters eingestellt. Die zur IFSEL-Auswahl gehörenden Schalter befinden sich innerhalb des Gehäuses.

Ebenfalls innerhalb des Gehäuses sind Schalter für die Auswahl der von Ihnen benutzten Maus vorhanden. Diese Schalter sind für den Betrieb der folgenden Mäuse werksseitig eingestellt:

- Hawley Mouse Mark II
- Alps Encoder-Mouse
- Mouse Systems Quad Mouse

Vorausgesetzt, daß Sie eine dieser Mäuse mit der Standardeinstellung der IFSEL-Nummer betreiben wollen, ist ein

Öffnen des Gehäuses des Maus-Adapters nicht erforderlich. In diesem Fall können Sie ohne weiteres den Adapter in eine der Steckfassungen 2...6 an der Rückseite Ihres NCR DECISION MATE V einsetzen (siehe Abb. 1.1). Anschließend sollten Sie den zweiten Teil dieser Anleitung ("Software") bezüglich der für den Betrieb einer Maus benötigten Software lesen.

Wenn Sie eine Depraz- oder eine Logitech-Maus benutzen, oder im Fall, daß Sie von einer vom Standardwert (2A) abweichenden IFSEL-Nummer Gebrauch machen möchten, sollten Sie gemäß der im folgenden Abschnitt enthaltenen Beschreibung verfahren.

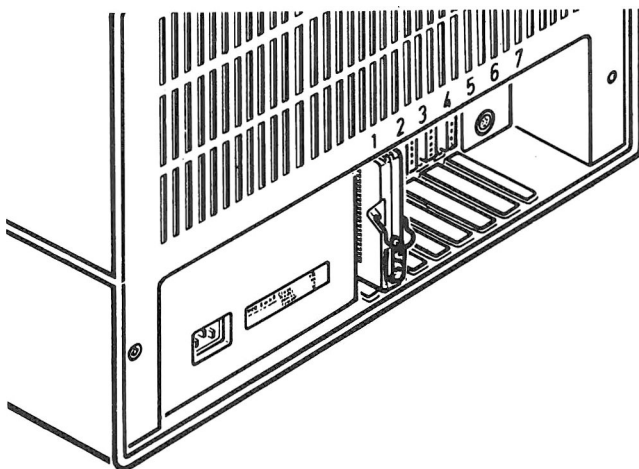


Abbildung 1.1: Der Maus-Adapter

IFSEL- UND MAUS-AUSWAHL

Die nachstehend beschriebenen Arbeitsschritte sind nur dann erforderlich, wenn Sie die IFSEL-Nummer ändern oder eine der folgenden Mäuse benutzen möchten:

WICHTIG:

Ein Federring befindet sich zwischen der mit den Leiterbahnen belegten Fläche der Platine und dem Gehäuse.

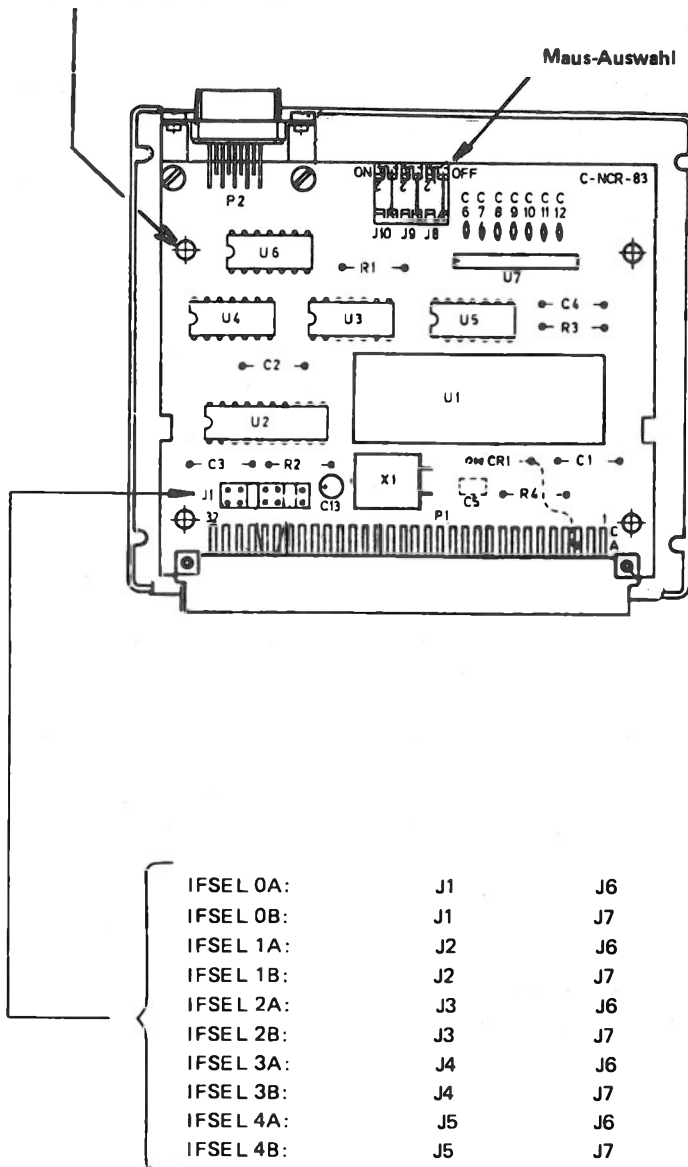


Abbildung 1.2: IFSEL- und Maus-Auswahl

- Depraz Souris P4
- Logitech LM-P-5
- Logitech P4

1. Entfernen Sie den am Gehäuse befestigten Drahtgriff und die vier Schrauben. Das Gehäuse sollten Sie noch nicht öffnen.
2. Halten Sie die zwei Teile des Gehäuses zusammen, und legen Sie den Adapter auf einen Schreibtisch o.ä., so daß die Schraublöcher nach unten zeigen. Entfernen Sie den oberen Teil des Gehäuses.
3. Auf der Platine des Maus-Adapters sind drei Schaltbrücken mit J10, J9, J8 gekennzeichnet (siehe auch Abbildung 1.2). Die Standardeinstellung 2A hat zur Folge, daß jede dieser drei Schaltbrücken in der OFF-Stellung ist. Wenn Sie eine Depraz- oder eine Logitech-Maus benutzen, müssen Sie alle drei Schalter in die ON-Stellung bringen. Hierzu ist jeder Draht aus seinem Haken zu lösen und in den jeweils unmittelbar links befindlichen Haken einzusetzen.
4. Die Aufschrift J1 auf der Platine (siehe auch Abbildung 1.2) zeigt Ihnen, wo die Schalter für die IFSEL-Auswahl zu finden sind. Die Einstellung einer IFSEL-Nummer erfordert eine Neubelegung der zwei Steckverbindungen auf den paarweise angeordneten Stiften (siehe Abb. 1.2). Die zwei am weitesten links befindlichen Stifte bilden die Schaltbrücke J1, die zwei am weitesten rechts befindlichen Stifte bilden die Schaltbrücke J7. Die zwei Steckverbindungen befinden sich auf den Schaltbrücken J3 bzw. J6, solange die IFSEL-Nummer 2A eingestellt ist.
5. Schrauben Sie das Gehäuse anhand der vier Schrauben zusammen; bringen Sie den Drahtgriff wieder an.

Abbildung 1.3 enthält eine Aufstellung der IFSEL-Nummern mit den jeweils verfügbaren Portadressen.

IFSEL-NR	PORT-ADRESSEN
0A	60-67
0B	68-6F
1A	70-77
1B	78-7F
2A	30-37
2B	38-3F
3A	B0-B7
3B	B8-BF
4A	C0-C7
4B	C8-CF

Abbildung 1.3: IFSEL/Portadressen

SOFTWARE**EINLEITUNG**

Der Maus-Adapter beinhaltet einen eigenen Mikroprozessor sowie eigene Firmware. Dies hat zur Folge, daß die zusätzliche Belastung des Prozessors des NCR DECISION MATE V durch den Betrieb des Maus-Adapters sehr gering ist. Der Maus-Adapter ist in der Lage, bis zu 4000 Positionen der angeschlossenen Maus in einer Sekunde an den Computer zu senden. Ihr Programm kann einen Bereich definieren, innerhalb dessen die Maus sich bewegen muß. (Sämtliche Positionen und Bereiche werden als X/Y-Koordinaten angegeben.) Der Maus-Adapter kann nicht nur die Position der Maus sondern auch den Zustand von bis zu drei an der Maus befindlichen Schaltern vermitteln. Ein Status-Byte gibt Aufschluß darüber, ob die internen ROM- und RAM-Speicher des Adapters einwandfrei funktionieren, ob der Adapter bereit ist, Befehle oder Daten zu senden oder zu empfangen, und ob das Interrupt-Signal gesetzt wurde.

Der Maus-Adapter bietet zwei Verfahrensweisen, auf die Ihr Programm die Bewegung der Maus und die Betätigung eines Schalters an der Maus verfolgen kann: Es kann in bestimmten Zeitabständen diese Daten von dem Maus-Adapter lesen. Die zweite Methode besteht darin, auf ein Interruptsignal zu warten, und erst dann die Positions- und Schalterdaten zu lesen. Sofern Sie von der Interrupt-Leitung des Computers Gebrauch machen wollen, sind in der Regel zusätzliche Interrupt-Controller-Schaltungen erforderlich. Wenn Sie aber Ihren NCR DECISION MATE V als 8-Bit-System benutzen, können Sie aufgrund bestimmter Eigenschaften des Z80A-Mikroprozessors Interrupt-Behandlung auch ohne solche Schaltungen verwirklichen (siehe hierzu den Abschnitt "Interrupt-Behandlung").

Die von NCR erhältliche p-System-Software beinhaltet eine Anzahl von BASIC-, FORTRAN- und Pascal-Programme für den Betrieb des Maus-Adapters. Eine Beschreibung dieser Programme in englischer Sprache liegt dieser Anleitung bei.

Sie können den Maus-Adapter selbstverständlich auch mit den Betriebssystemen CP/M und MS-DOS benutzen. Dieser Teil ("Software") beschreibt die Anweisungen für die Programmierung des Maus-Adapters. Die besten Ergebnisse lassen sich mit Programmen in Assemblersprache erzielen.

BENUTZUNG DER PORT-ADRESSEN

Der Maus-Adapter verwendet die erste und die zweite der zu gewählten IFSEL-Nummer gehörenden Port-Adressen. Vorausgesetzt, daß Sie die Standard-IFSEL-Nummer nicht geändert haben, sind die benutzten Port-Adressen 30H (Port 1) und 31H (Port 2). Die Ein- und Ausgabe über die Port-Adressen erfolgt gemäß folgender Einteilung:

- IN Port 1 - Das Programm liest Daten vom Maus-Adapter.
- OUT Port 1 - Der Maus-Adapter empfängt Daten von Ihrem Programm.
- IN Port 2 - Das Programm liest das Status-Byte des Maus-Adapters.
- OUT Port 2 - Das Programm sendet Befehle an den Maus-Adapter.

FUNKTIONSPRÜFUNG

Wenn Sie Ihren NCR DECISION MATE V bei eingesetztem Maus-Adapter einschalten, setzt das Reset-Signal des Computers auch den Maus-Adapter in seinen Anfangszustand zurück. Der Prozessor des Maus-Adapters liest dann Anweisungen im eigenen ROM: Zunächst wird geprüft, ob ROM und RAM des Maus-Adapters einwandfrei funktionieren. Dann initialisiert der Maus-Adapter seine I/O-Ports.

Sollte ein Fehlerzustand im ROM oder im RAM vorliegen, wird ein entsprechendes Bit im Status-Byte gesetzt. Der folgende Abschnitt erläutert die Bedeutung der einzelnen Bits des Status-Bytes und zeigt Ihnen, wie Ihr Programm dieses Byte lesen kann.

DAS STATUS-BYTE

Ihr Programm kann das Status-Byte mit einer IN-Anweisung an Port 2 (31H bei IFSEL 2A) lesen. Abbildung 2.1 zeigt die Bedeutung der einzelnen Bits dieses Bytes.

Bit:	7	6	5	4	3	2	1	0
	X	INT	RAM	ROM	X	X	IBF	OBF

Abbildung 2.1: Das Status-Byte

- X Dieses Bit wird nicht benutzt.
- INT Sobald der Maus-Adapter ein Interruptsignal ausgibt, wird dieses Bit gesetzt (logisch 1). Es bleibt in diesem Zustand, während die Interruptleitung aktiviert ist (active "low").
- ROM Ein Fehlerzustand im ROM des Maus-Adapters führt dazu, daß dieses Bit gesetzt wird. Sonst ist es zurückgesetzt.
- RAM Ein Fehlerzustand im RAM des Maus-Adapters führt dazu, daß dieses Bit gesetzt wird. Sonst ist es zurückgesetzt.
- IBF Input Buffer Full: Solange dieses Bit gesetzt ist, kann der Maus-Adapter weder Befehle noch Daten von Ihrem Programm empfangen. Vor der Ausgabe von Befehlen oder Daten an den Maus-Adapter sollte ein Programm den Zustand dieses Bits abfragen und die Ausgabe erst dann ausführen, wenn das Bit zurückgesetzt ist.
- OBF Sobald dieses Bit gesetzt wird, kann Ihr Programm ein Datenbyte über Port 1 des Maus-Adapters lesen. Das Programm sollte keinen Lesevorgang versuchen, während dieses Bit zurückgesetzt ist.

BEFEHLE UND IHRE PARAMETER

Dieser Abschnitt befaßt sich mit den vom Maus-Adapter anerkannten Befehlen. Einige dieser Befehle leiten die Übertragung einer Reihe von Parametern (Daten) ein.

WICHTIG: Nachdem ein zum Senden oder zum Empfangen von Parametern gehörender Befehl gesendet worden ist, müssen alle zu diesem Befehl gehörenden Parameter ausgegeben bzw. gelesen werden. Ihr Programm sollte die Übertragung der Parameterliste nicht vorzeitig abbrechen, indem es z.B. einen neuen Befehl an den Maus-Adapter sendet.

Alle Befehle an den Maus-Adapter sind über Port 2 auszugeben; Parameter werden über Port 1 sowohl gesendet als auch empfangen.

Die folgenden Werte werden beim Einschalten Ihres NCF DECISION MATE V mit Maus-Adapter automatisch eingestellt. Sie verlieren ihre Gültigkeit, erst wenn sie von Ihrem Programm bzw. einer Bewegung der Maus aufgehoben werden.

XMAX: Der höchste Wert, den die Maus in der X- (horizontalen) Richtung erreichen kann: 640
 YMAX: Der höchste Wert, den die Maus in der Y- (vertikalen) Richtung erreichen kann: 400
 XMIN: Der niedrigste Wert, den die Maus in der X-Richtung erreichen kann: ()
 YMIN: Der niedrigste Wert, den die Maus in der Y-Richtung erreichen kann: ()
 X- und Y-Koordinaten der gegenwärtigen Maus-Position: ()
 Alle Interruptsignale sind außer Kraft.

Nachstehend finden Sie eine Beschreibung jedes Befehls und der etwa dazugehörigen Parameter. Für jeden Befehl ist der hexadezimale Wert angegeben, der über Port 2 an den Maus-Adapter zu senden ist. Negative (Minus-) Werte werden immer als Zweierkomplement dargestellt.

00 Die absolute Position der Maus wird gelesen

Die Reihenfolge der zu lesenden Parameter:

X-Koordinate, niederwertiges Byte
 X-Koordinate, höherwertiges Byte
 Y-Koordinate, niederwertiges Byte
 Y-Koordinate, höherwertiges Byte
 Schalter-Status-Byte (s. Ende dieses Abschnitts)

Die absolute Position bezieht sich auf die Änderung der Position der Maus (ggf. innerhalb des definierten Bewegungsbereichs) seit dem letzten Setzen der Anfangsposition. (Diese Anfangsposition wird beim Einschalten auf X=0, Y=0 gesetzt. Sie kann ebenfalls anhand des Befehls 04 gesetzt werden.)

01 Das aufgrund einer Bewegung der Maus erzeugte Interruptsignal wird anerkannt

Keine Parameter.

Dieser Befehl hat gleichzeitig zur Folge, daß das Interruptsignal für die Mausschalter außer Kraft gesetzt wird.

02 Interruptsignale werden nicht erkannt

Keine Parameter.

03 Das Signal an der Interruptleitung und das INT-Bit im Status-Byte werden zurückgesetzt

Keine Parameter.

04 Die logische Position der Maus wird gesetzt

Die Reihenfolge der Parameter, die an den Maus-Adapter zu senden sind:

X-Koordinate, niederwertiges Byte

X-Koordinate, höherwertiges Byte

Y-Koordinate, niederwertiges Byte

Y-Koordinate, höherwertiges Byte

Wenn diese Position sich außerhalb des für die Bewegung der Maus definierten Bereichs liegt (s. Befehle 05 und 0A), findet die Definition dieses Bereichs keine Anwendung.

05 Der Höchstwert, den die Maus in der X- bzw. Y-Richtung erreichen kann, wird gesetzt

Die Reihenfolge der Parameter, die an den Maus-Adapter zu senden sind:

XMAX, niederwertiges Byte

XMAX, höherwertiges Byte

YMAX, niederwertiges Byte
YMAX, höherwertiges Byte

Das Verhältnis der mit der Maus zurückgelegten Entfernung zum Wert, der für diese Entfernung vom Maus-Adapter gesendet wird, ist für die verschiedenen Mäuse unterschiedlich. Die beim Einschalten automatisch eingestellten Werte (XMAX = 640, YMAX = 400) entsprechen einer angenehmen Reichweite bei der Benutzung einer Maus auf einem Schreibtisch.

06 Das aufgrund der Betätigung eines Maus-Schalters erzeugte Interruptsignal wird anerkannt

Keine Parameter.

Dieser Befehl hat gleichzeitig zur Folge, daß das bei einer Bewegung der Maus erzeugte Interruptsignal außer Kraft gesetzt wird.

07 Die Interruptsignale sowohl für die Betätigung eines Schalters als auch für eine Bewegung der Maus werden erkannt

Keine Parameter.

08 Der ROM des Maus-Adapters wird überprüft

Keine Parameter.

Diese Überprüfung erfolgt auch beim Einschalten.

09 Die relative Position der Maus wird gelesen

Parameter: Siehe 00.

Die relative Position der Maus bezieht sich auf den Positionsunterschied zwischen der gegenwärtigen Position und der zuletzt gelesenen Position.

0A Der Niedrigstwert, den die Maus in der X- bzw. Y-Richtung erreichen kann

Die Reihenfolge der Parameter, die an den Maus-Adapter zu senden sind:

XMIN, niederwertiges Byte
XMIN, höherwertiges Byte
YMIN, niederwertiges Byte
YMIN, höherwertiges Byte

WICHTIG: Absolute und relative Position der Maus werden in denselben Registern des Maus-Adapters gespeichert. Infolgedessen ist bei der Benutzung beider Positionsformate in einem Programm Vorsicht geboten. Im Fall, daß sowohl die absolute als auch die relative Position der Maus in Ihrem Programm benötigt wird, empfiehlt es sich, die absolute Position als X/Y-Koordinaten in Programmvariablen festzuhalten. Sie können mit Hilfe des Befehls 04 diese Werte in die Positionsregister wieder zurückschreiben.

Der Maus-Adapter berücksichtigt die am Gehäuse der Maus vorhandenen Schalter. Der am weitesten links befindliche Schalter wird als S1, der am weitesten rechts befindliche Schalter als S2 bezeichnet. Ein etwa in der Mitte vorhandener Schalter wird als S3 bezeichnet. Jedem Schalter ist ein Bit im Schalter-Status-Byte (Befehl 00) zugeordnet:

S1 - Bit 0
S2 - Bit 1
S3 - Bit 2

(Die verbleibenden fünf Bits dieses Bytes werden nicht benutzt.) Das Bit ist gesetzt, während der entsprechende Schalter sich in gedrückter Stellung befindet.

INTERRUPT-BEHANDLUNG

Dieser Abschnitt befaßt sich mit den Interruptsignalen, die von dem Maus-Adapter ausgegeben werden. Vorausgesetzt, daß die Interruptsignale nicht außer Kraft sind, wird ein solches Signal bei jeder Bewegung der Maus bzw. bei jedem Öffnen und jedem Schließen der Kontakte eines Schalters.

WICHTIG: Die Benutzung der Interrupt-Leistungen des Maus-Adapters (über die Abfrage des INT-Bits des Status-Bytes hinaus) setzt Erfahrung im Umgang mit Halbleiter-Bauteilen voraus. Wichtige Informationen über die Hardware Ihres NCR DECISION MATE V finden Sie im Band "Hardware" des von NCR herausgegebenen System Technical Manual.

Wenn Sie Ihren NCR DECISION MATE V als 16-Bit-System benutzen, sind für die Behandlung von Interruptsignalen zusätzliche Interrupt-Controller unerlässlich.

Bestimmte Eigenschaften des Z80A-Mikroprozessors ermöglichen in einem 8-Bit-System eine Interrupt-Behandlung ohne Interrupt-Controller: Beim Interrupt-Modus 1 (IM 1 / Opcode: EI 56) wird beim Erzeugen eines Signals auf der Interruptleitung die Programmsteuerung an die hexadezimale Adresse 38 übergeben. Das Betriebssystem CP/M-80 enthält an dieser Adresse einen Sprungbefehl (JP) an das zu Testzwecken eingesetzte DDT-Programm. Vorausgesetzt, daß DDT für Ihre Anwendung nicht benötigt wird, können Sie einen Sprungbefehl an dieser Adresse ablegen. Dieser könnte dann auf Ihr eigenes Interrupt-Behandlungsprogramm verweisen.

C H A P T E R 8
M O U S E

(This is the description of the Mouse unit of the p-SystemTM. Please insert it in the manual UCSD p-System, Programming, chapter 8)

p-SystemTM is a trademark of Softech Microsystems, Inc., and is used pursuant to a license granted by Softech Microsystems, Inc.

Introduction

The following is a description of the procedures you can use with the p-System™ to work with your mouse.

The software you need for using the Mouse interface and working with the Mouse in the p-System™ is contained in the MOUSE.CODE file. If you have a Runtime System MOUSE.CODE is incorporated in your SYSTEM.LIBRARY. If you have a Plus System, you have to insert this code in your SYSTEM.LIBRARY, in any of slots 0 through 15. Chapter 6 utility 'LIBRARY' in the 'UCSD p-System, Programming' Manual describes how this is done.

The interface part of the Mouse unit affords access to the following procedures:

GMOUS: gives the status of the Mouse switches and Mouse position. This procedure supplies the absolute position (x and y coordinates) of the Mouse and the status of the switch(es) (up to 3). Use all three switch variables, even if your Mouse has only one or two switches.

X,Y: absolute x and y coordinates

SW1,SW2,SW3: status of the Mouse switches (1 = pressed, 0 = not pressed or not existent)

SETPOS: sets the logical position of the Mouse.

X,Y: New x and y coordinates for the Mouse position.

Note: Use this option to set a new position for the Mouse. If you require no direct correlation between the physical position of the Mouse and its program position, you can reseat

the logical Mouse position by means of SETPOS.

SETMAX: sets maximum values for the Mouse position. Sets the maximum x and y values the Mouse can attain (default: x=640, y=400).

XMAX,YMAX: are maximum values for the x and y coordinates

For SETPOS note that the initial Mouse position is x=0, y=0. The Mouse can move within the area delimited by x=0, y=0 and the values set in SETMAX.

Pascal Procedures

To use the Mouse procedures in a Pascal program, declare
USES MOUSE;

The first lines of the respective procedures have the following forms:

GMOUS

```
PROCEDURE GMOUS  
  (VAR SW3, SW2, SW1, Y, X: INTEGER)
```

SETPOS

```
PROCEDURE SETPOS (X,Y: INTEGER)
```

SETMAX

PROCEDURE SETMAX (XMAX,YMAX: INTEGER)

Example

Here is an example for the use of the Mouse procedures in Pascal.

In the first part of the main program, the maximum Mouse position is set by means of the X and Y coordinates; and the present Mouse position is set to zero. In the second part of the main program, the variables are collected and displayed on the screen. SW1, SW2, and SW3 designate the switches on the Mouse (here the HAWLEY MARK II Mouse) which can be on or off. X is the Mouse position on the X coordinate, Y the Mouse position on the Y coordinate.

```
{ EXAMPLE MOUSE PROGRAM PASCAL }

PROGRAM MOUSETP;

  USES MOUSE;

  VAR SW1,SW2,SW3,X,Y,X0,Y0,XM,YM:INTEGER; C:CHAR;

  PROCEDURE FIN(VAR C:CHAR);
  { PROMPT FOR FINISH }
  BEGIN
    GOTOXY(0,18);
    WRITE(' DO YOU WANT TO FINISH? Y/N ');
    READLN(C);
  END;

  BEGIN
  { CLEAR SCREEN, CURSOR INVISIBLE }
    WRITELN(CHR(27),'@0',CHR(27),CHR(69));

    WRITELN(' ***** M O U S E   E X A M P L E   P A S C A L   *****')
    GOTOXY(0,16);
    WRITELN(' TO STOP PRESS SWITCH 1 ');
```

```

{ SET MAXIMUM }
  XM:=10000;
  YM:=10000;
  SETMAX(XM,YM);

{ SET START POSITION }
  X0:=0;
  Y0:=0;
  SETPOS(X0,Y0);

  C:='N';

{ LOOP MOUSE ACTIONS }
  REPEAT
    GMOUS(SW3,SW2,SW1,Y,X);
    GOTOXY(0,4);
    WRITELN('      X=',X:6,'      Y=',Y:6);
    GOTOXY(0,6);
    WRITELN('      SWITCH 1:',SW1,' SWITCH 3:',SW3,' SWITCH 2:',SW2);
    GOTOXY(0,18);

{ SET OR CLEAR PROMPT FOR FINISH DEPENDING ON SW1 }

    IF SW1=1 THEN FIN(C)
      ELSE WRITE(CHR(27),'L');

    UNTIL C in ['y','Y'];

{ CURSOR VISIBLE }
  WRITE(CHR(27),'@1')
  END.

```

Fortran Procedures

To use the Mouse procedures in a Fortran program, declare

```
$USES MOUSE
```

The subroutine and parameter definition can be found below:

GMOUS

```
SUBROUTINE GMOUS(SW3,SW2,SW1,Y,X)
  INTEGER SW1, SW2, SW3, Y, X
```

SETPOS

```
SUBROUTINE SETPOS (X,Y)
  INTEGER X,Y
```

SETMAX

```
SUBROUTINE SETMAX (XMAX,YMAX)
  INTEGER XMAX,YMAX
```

Example

Here is an example for the use of the Mouse procedures in FORTRAN.

In the line starting with 1, the maximum Mouse position is set by means of the X and Y coordinates. In the line starting with 2, the present Mouse position is set to zero. In the line starting with 10, SW1, SW2, and SW3 designate the switches on the Mouse (here the HAWLEY MARK I Mouse), which can be on or off. X is the Mouse position.

on the X coordinate; y the Mouse position on the Y coordinate.

Then the program collects the values and displays them on the screen.

```
C
C      MOUSE EXAMPLE PROGRAM FOR FORTRAN
C
$USES MOUSE

      PROGRAM MOUSTF

      INTEGER SW1,SW2,SW3,X,Y,X0,Y0,XM,YM
      CHARACTER C

C      CLEAR SCREEN, CURSOR INVISIBLE
      WRITE(*,'(AAAA)') CHAR(27),'@0',CHAR(27),'E'

      WRITE(*,200)
200    FORMAT(4X,'***** M O U S E   E X A M P L E   F O R   T R A N   *****')
C      SCREEN POSITION LINE 16 COLUMN 0
      WRITE(*,'(AA\)' ) CHAR(27),'Y0 '
      WRITE(*,'(A)' ) '    TO STOP PRESS SWITCH 1 '

C      SET MOUSE MAX VAL
      XM=10000
      YM=10000
1      CALL SETMAX(XM,YM)

C      SET MOUSE POSITION
      X=0
      Y=0
2      CALL SETPOS(X,Y)

      C='N'
```

```

C      LOOP MOUSE ACTIONS
      DO 1000 I=0,0,0
100    CALL GMOUS(SW3,SW2,SW1,Y,X)
      WRITE(*,'(AA\)' ) CHAR(27),'Y$ '
C      SCREEN POSITION LINE 4, COLUMN 0
      WRITE(*,300) X,Y
300    FORMAT ('      X=',I6,'      Y=',I6//)
      WRITE(*,100) SW1,SW3,SW2
100    FORMAT('      SWITCH1:',I1,'      SWITCH3:',I1,'      SWITCH2:',I1)
C      SCREEN POSITION LINE 18, COLUMN 0
      WRITE(*,'(AA\)' ) CHAR(27),'Y2 '

C      SET OR RESET PROMPT FOR FINISH DEPENDING ON SW1
      IF (SW1.EQ. 1) THEN
        CALL FIN(C)
      ELSE
        WRITE(*,'(AA\)' ) CHAR(27),'L'
      ENDIF

      IF ((C.EQ. 'Y') .OR. (C.EQ. 'y')) THEN
        I=1
      ENDIF
1000   CONTINUE

C      CURSOR VISIBLE
      WRITE(*,'(AA)' ) CHAR(27),'@1'
      END

      SUBROUTINE FIN (C)
C      PROMPT FOR FINISH
      CHARACTER C
      WRITE(*,'(A\)' ) '      DO YOU WANT TO FINISH?      Y/N '
      READ(*,'(A1)' ) C
      RETURN
      END

```

BASIC Interface

To use the Mouse procedures in a BASIC program, declare
USES MOUSE

The subroutine and parameter definition can be found below:

GMOUS

```
SUB GMOUS(SW3,SW2,SW1,Y,X)
  INTEGER SW1, SW2, SW3, Y, X
```

SETPOS

```
SUB SETPOS (X,Y)
  INTEGER X,Y
```

SETMAX

```
SUB SETMAX (XMAX,YMAX)
  INTEGER XMAX,YMAX
```

Example

Here is an example for the use of the Mouse procedures :
BASIC.

In the line starting with 10, the maximum Mouse position is
set by means of the X and Y coordinates. In the line
starting with 20, the present Mouse position is set to zero.
In the line of the main program starting with 100, the
variables are collected and displayed on the screen.

SW1, SW2, and SW3 designate the switches on the Mouse (here the HAWLEY MARK II Mouse), which can be on or off. X is the Mouse position on the X coordinate; y the Mouse position on the Y coordinate.

```

REM
REM  MOUSE TEST PROGRAM  B A S I C
REM

USES MOUSE

INTEGER SW1,SW2,SW3,X,Y,X0,Y0,XM,YM
DIM C$*1

SUB FIN (C$)
REM PROMPT FOR FINISH
INPUT AT (19,1) " DO YOU WANT TO FINISH?  Y/N ":C$
SUBEND

REM CLEAR SCREEN, CURSOR INVISIBLE
DISPLAY ERASE ALL:CHR$(27);"@0"

DISPLAY AT (2,1): " *****  M O U S E  E X A M P L E  B A S I C  *****
DISPLAY AT (17,1): " TO STOP PRESS SWITCH 1 "

REM SET MOUSE MAX VAL
XM=10000
YM=10000
10 CALL SETMAX(XM,YM)

REM SET MOUSE POSITION
X0=0
Y0=0
20 CALL SETPOS(X0,Y0)

C$="N"

REM LOOP MOUSE ACTION
FOR I=0 TO 0 STEP 0
100 CALL GMOUS(SW3,SW2,SW1,Y,X)
DISPLAY AT(5,1) USING " X=##### Y=#####":X,Y
DISPLAY AT(7,1): " SWITCH 1: ";SW1;" SWITCH 3: ";SW3;" SWITCH 2: ";SW

REM SET OR CLEAR PROMPT FOR FINISH DEPENDING ON SW1
IF SW1=1 THEN CALL FIN(C$) ELSE DISPLAY AT (19,1):CHR$(27);"L";

IF C$="Y" OR C$="y" THEN I=1
NEXT I

1000 REM CURSOR VISIBLE
DISPLAY AT (1,1):CHR$(27);"@1"
END

```



KIT INSTALLATION

for

**MOUSE INTERFACE
(K806)**

The attached pages provide information for installing this kit into an NCR DECISION MATE V. These pages should be filed in your NCR DECISION MATE V User Information Manual.

**FEDERAL COMMUNICATIONS COMMISSION (FCC)
RADIO FREQUENCY INTERFERENCE STATEMENT**

WARNING

This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.

Information to User

This equipment must be installed and used in strict accordance with the manufacturer's instructions. However, there is no guarantee that interference to radio communications will not occur in a particular commercial installation. If this equipment does cause interference, which can be determined by turning the equipment off and on, the user is encouraged to consult an NCR service representative immediately.

The NCR Corporation (NCR) is not responsible for any radio or television interference caused by unauthorized modification of this equipment or the substitution or attachment of connecting cables and equipment other than those specified by NCR. The correction of interferences caused by such unauthorized modification, substitution or attachment will be the responsibility of the user.

This equipment is designed for use in a commercial environment, not in a residential area. It fully complies with the relevant FCC regulations; when connected to any Class B equipment then the total product is reclassified accordingly.

It is the policy of NCR Corporation to improve products as new technology, components, software, and firmware become available. NCR Corporation, therefore, reserves the right to change specifications without prior notice.

All features, functions, and operations described herein may not be marketed by NCR in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult your NCR representative or NCR office for information that is applicable and current.

KIT INSTALLATION

**MOUSE INTERFACE
(K806)**

CONTENTS

**Part 1
INSTALLATION**

INTRODUCTION.....	1
INTERFACE AND MOUSE SELECTION.....	2

**Part 2
SOFTWARE**

INTRODUCTION.....	5
USE OF PORTS.....	6
POWER-UP CHECK.....	6
THE STATUS BYTE.....	6
COMMANDS AND PARAMETERS.....	7
INTERRUPTS.....	10

INSTALLATION

INTRODUCTION

A mouse is a hand-held device which facilitates movement on a screen display in response to movement of the hand on a level surface such as a desk-top. The Mouse Interface for your NCR DECISION MATE V provides the means of using the following "mice":

- Hawley Mouse MARK II
- Alps Encoder-Mouse
- Mouse Systems Quad Mouse
- Depraz Souris P4
- Logitech LM-P-5
- Logitech P4

The Mouse Interface, like other adapters which can be installed in your NCR DECISION MATE V, makes use of an IFSEL (InterFace SElect) number in order to communicate with the computer. There is a total of 10 IFSELS available, each governing 8 I/O ports. The p-System software available for the Mouse Interface uses the IFSEL 2A (ports 30H to 37H) as a default setting. This is the IFSEL number already set when you receive the Mouse Interface.

The Mouse Interface is factory set to operate with the following mice:

- Hawley Mouse MARK II
- Alps Encoder-Mouse
- Mouse Systems Quad Mouse

If you are using one of these mice, and provided that you do not wish to change the default IFSEL number (2A), there is no need to open the Mouse Interface. In this case, you can insert the adapter into a vacant slot 2...6 at the rear of your NCR DECISION MATE V (see Figure 1.1). Then refer to Part 2 of this document for information about software support for the Mouse Interface.

If you are using a Depraz or Logitech mouse, or if your system requires the use of a different IFSEL number, you should proceed in accordance with the following section.

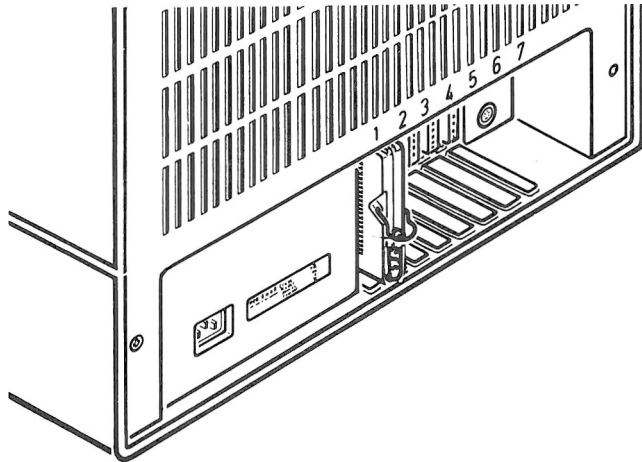


Figure 1.1 Mouse Interface

INTERFACE AND MOUSE SELECTION

The following procedures are necessary only if you wish to change the IFSEL number or use one of the following mice:

- Depraz Souris P4
 - Logitech LM-P-5
 - Logitech P4
1. Remove the wire handle and the four screws of the Mouse Interface. Do not yet open the housing.
 2. Holding the two halves of the housing together, turn the interface upside down and place it on a level surface. Now lift off the upper half of the housing, so that the

CAUTION:

Do not lose the spring washer situated between the solder side of the printed circuit board and the housing

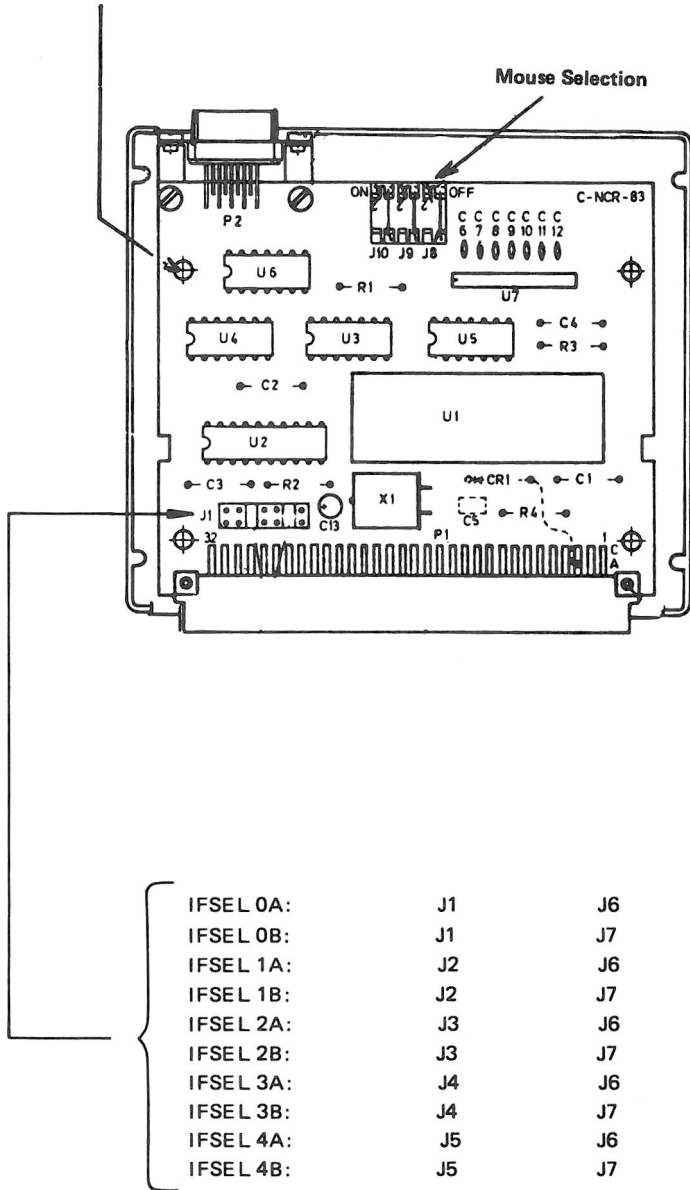


Figure 1.2 Interface and Mouse Selection

component side (not the solder side) of the component board is showing.

3. Refer to Figure 1.2. The factory setting of the wire jumpers J10, J9, and J8 as shown in this Figure is all OFF. If you are using one of the three mice named above, these jumpers must be all ON. To set a jumper to the ON position, unhook the wire from its retaining clip and hook it under the retaining clip immediately to the left.
4. A small panel holding 7 pairs of connector pins is situated immediately behind the multipin connector plug at the position marked J1 on the component board. If you wish to change the IFSEL number, you must place the two bridge connectors over the appropriate pins. In Figure 1.2, the leftmost pair of pins is J1, the rightmost J7. These connectors are factory set to IFSEL 2A. The table included in Figure 1.2 states which two pairs of pins have to be bridged for each IFSEL number.
5. Re-assemble the Mouse Interface.

Figure 1.3 shows the group of port addresses belonging to each IFSEL number.

IFSEL	PORTS
0A	60-67
0B	68-6F
1A	70-77
1B	78-7F
2A	30-37
2B	38-3F
3A	B0-B7
3B	B8-BF
4A	C0-C7
4B	C8-CF

Figure 1.3 IFSEL Port Assignments

SOFTWARE

INTRODUCTION

The Mouse Interface has its own microprocessor and firmware. This offloads the microprocessor of your NCR DECISION MATE V to a great extent. The Mouse Interface is capable of transmitting the position of the mouse up to 4000 times a second. It is possible to define a range of movement in terms of Cartesian co-ordinates. In addition to the position of the mouse, the status of up to three switches is included in the transmission protocol to the computer. A status byte provides information about function integrity checks and whether the Mouse Interface is ready to accept or transmit commands and data. In addition, the status byte records whether an interrupt has been issued.

There are two methods of checking mouse position and switch status. The first is simply to poll the Mouse Interface for the appropriate information. The second method is to make use of interrupts issued by the interface. Normally, this requires additional hardware in the form of interrupt controller circuitry. However, if you are using your NCR DECISION MATE V as an 8-bit system, it is possible to utilize certain characteristics of the Z80A microprocessor in order to create a limited interrupt handling facility. This is described in the section "Interrupts".

p-System software provides support, consisting of a number of BASIC, FORTRAN, and Pascal routines, for the Mouse Interface. A description of these routines is included at the end of this document. This description is ready for inserting as Chapter 8 of the p-System Plus, Volume 2, Programming Manual.

The Mouse Interface can, of course, be used with the CP/M and MS-DOS operating systems. This Part of the document is intended for programmers who wish to write their own Mouse Interface driver routines. The most effective results are to be achieved by programs in assembly language.

USE OF PORTS

The Mouse Interface uses the first two addresses of the eight available under an IFSEL number. If the IFSEL number is 2A (the default setting), then these two ports are 30H (Port 1) and 31H (Port 2). The significance of these ports is as follows:

- IN Port 1 - Read data from Mouse Interface.
- OUT Port 1 - Write data to Mouse Interface.
- IN Port 2 - Read the Mouse Interface status byte.
- OUT Port 2 - Write commands to the Mouse Interface.

POWER-UP CHECK

When you switch on the computer with the Mouse Interface inserted, the signal which resets the whole system also initiates the local processor in the Mouse Interface to execute instructions read from its own read-only memory (ROM). These instructions include an integrity check of both the ROM itself and the read/write memory (RAM) on the Mouse Interface component board. The I/O ports are then initialized.

Should the Mouse Interface register a fault in ROM or RAM, the appropriate bit of the status byte is set. Otherwise these two bits are set to zero. The following section includes information about how to read and evaluate the status byte.

THE STATUS BYTE

Your program can read the status byte by means of an IN instruction to Port 2 of the two ports used by the Mouse Interface. Figure 2.1 shows the significance of the individual bits within this byte.

Bit:	7	6	5	4	3	2	1	0
	X	INT	RAM	ROM	X	X	IBF	OBF

Figure 2.1 The status byte

- X Indicates that this bit is unused.
- INT When this bit is set, the Mouse Interface has issued an interrupt and the interrupt line is still active (low).
- ROM If this bit is set, there is a fault in the ROM of the Mouse Interface.
- RAM If this bit is set, there is a fault in the RAM of the Mouse Interface.
- IBF Input Buffer Full: when this bit is set, the Mouse Interface is busy. Your program should wait until this bit is reset before transmitting commands or data to the Mouse Interface.
- OBF Output Buffer Full: when this bit is set, one byte of data is available for reading by the processor of your NCR DECISION MATE V. Your program should not attempt to read data while this bit is zero.

COMMANDS AND PARAMETERS

This section consists of a summary of the commands recognized by the microprocessor of the Mouse Interface. A number of these commands require subsequent parameters, either to be transmitted to or read from the Mouse Interface.

NOTE: After issuing a command entailing parameters, it is important to complete all parameters formally detailed for that command. Therefore, do not terminate prematurely the parameter sequence of a command by transmitting a further command.

Remember that all commands are transmitted to the Mouse Interface via Port 2, parameters (data) via Port 1. Parameters are read from the Mouse Interface via Port 1.

The following default values are automatically set by the Mouse Interface upon switching on. They remain in force until changed by your program, or, in the case of the current mouse position, by movement of the mouse itself.

XMAX: maximum range in the X (horizontal) direction 640
YMAX: maximum range in the Y (vertical) direction 400
XMIN: minimum range in the X direction 0
YMIN: minimum range in the Y direction 0
X and Y coordinates of the current mouse position, both 0
All interrupts disabled.

The summary of commands and parameters follows. The hexadecimal value for the OUT instruction via Port 2 two is given for each command, together with a description of its purpose and any ensuing parameters. Negative values are represented as two's complement.

00 Read absolute position of mouse

Sequence of parameters subsequently read:

X co-ordinate, low byte
X co-ordinate, high byte
Y co-ordinate, low byte
Y co-ordinate, high byte
Switch status byte (see below)

The absolute position refers to movement of the mouse (within the delimited range) since the starting position of the mouse was last set (initially set at switching on to 0,0; can be subsequently set by command 04).

01 Enable interrupt issued when mouse is moved

No parameters.

At the same time, this command disables the switch interrupt.

02 Disable all interrupts

No parameters.

03 Reset the interrupt line and INT bit (status byte)

No parameters.

04 Set logical mouse position

Sequence of parameters transmitted to the interface:

X co-ordinate, low byte
X co-ordinate, high byte
Y co-ordinate, low byte
Y co-ordinate, high byte

If the position thus set is outside the delimited range (see commands 05 and 0A), that range does not apply.

05 Set maximum X and Y values the mouse can attain.

Sequence of parameters transmitted to the interface:

XMAX, low byte
XMAX, high byte
YMAX, low byte
YMAX, high byte

The relation between the X/Y co-ordinates computed by the Mouse Interface and the actual distance covered by the mouse varies according to the type of mouse being used. The default settings (XMAX = 640, YMAX = 400) correspond to practicable ranges for using a mouse on a desk top.

06 Enable switch interrupt

No parameters.

At the same time, this command disables the mouse movement interrupt.

07 Enable both the mouse movement and the switch interrupt.

No parameters.

08 Perform a ROM check

No parameters.

This is the same check as is performed at switching on.

09 Read the relative position of the mouse

Parameters as for command 00.

The relative position of the mouse refers to movement in the X and Y directions since the mouse position was last

read. (The absolute position refers to movement from the starting position.)

0A Set the minimum X and Y values the mouse can attain

Sequence of parameters to be transmitted to the Mouse Interface:

XMIN, low byte
XMIN, high byte
YMIN, low byte
YMIN, high byte

NOTE: Absolute and relative position are stored in the same set of internal registers. This means that a program cannot read both the absolute and the relative mouse position. If this facility is required, you should consider storing a position required for later reference in an area in your program. You can then write this position back to the Real-Time-Clock (Command 04) at an appropriate point in your program.

The Mouse Interface can handle up to three switches on the mouse. The leftmost switch is designated S1, the rightmost S2, and the centre switch, if present on the mouse, S3. The switch status byte (command 00) uses bit 0 to record the status of S1; bit 1 is used for S2, and bit 2 for S3. In each case, the bit is set when the appropriate switch is in the depressed condition, otherwise the bit is zero. The five most significant bits of the switch status byte are unused.

INTERRUPTS

This section is intended for specialist applications of the Mouse Interface, for which the use of interrupts is required. Assuming that both interrupts are enabled, the Mouse Interface issues an interrupt whenever the mouse is moved, and whenever any one of the switches changes state (ON to OFF, or OFF to ON).

CAUTION: The setting up of interrupt facilities beyond that of software sampling of the interrupt bit in the status byte requires knowledge of and experience in dealing with semiconductor components and circuitry. In addition, you should consult the Hardware volume of the System Technical Manual for the NCR DECISION MATE V.

If you are using the 16-bit version of the processor for the NCR DECISION MATE V, you will require interrupt controller circuitry, in addition to access to the interrupt line of the computer bus system.

The 8-bit processor (Z80A) has a characteristic which offers a limited interrupt handling facility with a minimum of hardware overhead, namely the Interrupt Mode 1. When this interrupt mode is set (Instruction: IM 1 / Opcode: ED 56) the setting of an interrupt (interrupt line goes low) forces the transfer of program control to the hexadecimal address: 38. The CP/M-80 operating system contains at this address an unconditional jump to the DDT utility. This address is used when a breakpoint is encountered during the testing of a program. If use of the DDT is not required, it is possible to write a different jump instruction to address 38H. This jump instruction can pass program control to your own interrupt service routine.

C H A P T E R 8

M O U S E

(This is the description of the Mouse unit of the p-System™. Please insert it in the manual UCSD p-System, Programming, chapter 8)

p-System™ is a trademark of Softech Microsystems, Inc., and is used pursuant to a license granted by Softech Microsystems, Inc.

Introduction

The following is a description of the procedures you can use with the p-System™ to work with your mouse.

The software you need for using the Mouse interface and working with the Mouse in the p-System™ is contained in the MOUSE.CODE file. If you have a Runtime System MOUSE.CODE is incorporated in your SYSTEM.LIBRARY. If you have a Plus System, you have to insert this code in your SYSTEM.LIBRARY, in any of slots 0 through 15. Chapter 6, utility 'LIBRARY' in the 'UCSD p-System, Programming' Manual describes how this is done.

The interface part of the Mouse unit affords access to the following procedures:

GMOUS: gives the status of the Mouse switches and Mouse position. This procedure supplies the absolute position (x and y coordinates) of the Mouse and the status of the switch(es) (up to 3). Use all three switch variables, even if your Mouse has only one or two switches.

X,Y: absolute x and y coordinates

SW1,SW2,SW3: status of the Mouse switches (1 = pressed, 0 = not pressed or not existent)

SETPOS: sets the logical position of the Mouse.

X,Y: New x and y coordinates for the Mouse position.

Note: Use this option to set a new position for the Mouse. If you require no direct correlation between the physical position of the Mouse and its program position, you can reset

the logical Mouse position by means of SETPOS.

SETMAX: sets maximum values for the Mouse position. Sets the maximum x and y values the Mouse can attain (default: x=640, y=400).

XMAX,YMAX: are maximum values for the x and y coordinates

For SETPOS note that the initial Mouse position is x=0, y=0. The Mouse can move within the area delimited by x=0, y=0 and the values set in SETMAX.

Pascal Procedures

To use the Mouse procedures in a Pascal program, declare
USES MOUSE;

The first lines of the respective procedures have the following forms:

GMOUS

```
PROCEDURE GMOUS  
  (VAR SW3, SW2, SW1, Y, X: INTEGER)
```

SETPOS

```
PROCEDURE SETPOS (X,Y: INTEGER)
```

SETMAX

PROCEDURE SETMAX (XMAX,YMAX: INTEGER)

Example

Here is an example for the use of the Mouse procedures in Pascal.

In the first part of the main program, the maximum Mouse position is set by means of the X and Y coordinates; and the present Mouse position is set to zero. In the second part of the main program, the variables are collected and displayed on the screen. SW1, SW2, and SW3 designate the switches on the Mouse (here the HAWLEY MARK II Mouse), which can be on or off. X is the Mouse position on the X coordinate, Y the Mouse position on the Y coordinate.

```
{ EXAMPLE MOUSE PROGRAM PASCAL }
PROGRAM MOUSETP;
  USES MOUSE;
  VAR SW1,SW2,SW3,X,Y,X0,Y0,XM,YM:INTEGER; C:CHAR;
  PROCEDURE FIN(VAR C:CHAR);
  { PROMPT FOR FINISH }
  BEGIN
    GOTOXY(0,18);
    WRITE('    DO YOU WANT TO FINISH?   Y/N ');
    READLN(C);
  END;
  BEGIN
  { CLEAR SCREEN, CURSOR INVISIBLE }
    WRITELN(CHR(27),'@0',CHR(27),CHR(69));
    WRITELN('    *****  M O U S E  E X A M P L E  P A S C A L  *****');
    GOTOXY(0,16);
    WRITELN('    TO STOP PRESS SWITCH 1 ');
```

```

{ SET MAXIMUM }
  XM:=10000;
  YM:=10000;
  SETMAX(XM,YM);

{ SET START POSITION }
  X0:=0;
  Y0:=0;
  SETPOS(X0,Y0);

  C:='N';

{ LOOP MOUSE ACTIONS }
  REPEAT
    GMOUS(SW3,SW2,SW1,Y,X);
    GOTOXY(0,4);
    WRITELN('      X=',X:6,'      Y=',Y:6);
    GOTOXY(0,6);
    WRITELN('      SWITCH 1:',SW1,' SWITCH 3:',SW3,' SWITCH 2:',SW2);
    GOTOXY(0,18);

{ SET OR CLEAR PROMPT FOR FINISH DEPENDING ON SW1 }

    IF SW1=1 THEN FIN(C)
      ELSE WRITE(CHR(27),'L');

  UNTIL C in ['y','Y'];

{ CURSOR VISIBLE }
  WRITE(CHR(27),'@1')
  END.

```

Fortran Procedures

To use the Mouse procedures in a Fortran program, declare

```
$USES MOUSE
```

The subroutine and parameter definition can be found below:

GMOUS

```
SUBROUTINE GMOUS(SW3,SW2,SW1,Y,X)  
  INTEGER SW1, SW2, SW3, Y, X
```

SETPOS

```
SUBROUTINE SETPOS (X,Y)  
  INTEGER X,Y
```

SETMAX

```
SUBROUTINE SETMAX (XMAX,YMAX)  
  INTEGER XMAX,YMAX
```

Example

Here is an example for the use of the Mouse procedures in FORTRAN.

In the line starting with 1, the maximum Mouse position is set by means of the X and Y coordinates. In the line starting with 2, the present Mouse position is set to zero. In the line starting with 10, SW1, SW2, and SW3 designate the switches on the Mouse (here the HAWLEY MARK II Mouse), which can be on or off. X is the Mouse position

on the X coordinate; y the Mouse position on the Y coordinate.

Then the program collects the values and displays them on the screen.

```
C
C      MOUSE EXAMPLE PROGRAM FOR FORTRAN
C
$USES MOUSE

      PROGRAM MOUSTF

      INTEGER SW1,SW2,SW3,X,Y,X0,Y0,XM,YM
      CHARACTER C

C      CLEAR SCREEN, CURSOR INVISIBLE
      WRITE(*,'(AAAA)') CHAR(27),'@0',CHAR(27),'E'

      WRITE(*,200)
200    FORMAT(4X,'***** M O U S E  E X A M P L E  F O R  T R A N  *****')
C      SCREEN POSITION LINE 16 COLUMN 0
      WRITE(*,'(AA\)' ) CHAR(27),'Y0 '
      WRITE(*,'(A)' ) '      TO STOP PRESS SWITCH 1 '

C      SET MOUSE MAX VAL
      XM=10000
      YM=10000
1      CALL SETMAX(XM,YM)

C      SET MOUSE POSITION
      X=0
      Y=0
2      CALL SETPOS(X,Y)

      C='N'
```



```

C      LOOP MOUSE ACTIONS
      DO 1000 I=0,0,0
10     CALL GMOUS(SW3,SW2,SW1,Y,X)
      WRITE(*,'(AA\)' ) CHAR(27),'Y$ '
C      SCREEN POSITION LINE 4, COLUMN 0
      WRITE(*,300) X,Y
300    FORMAT ('      X=',I6,'      Y=',I6//)
      WRITE(*,100) SW1,SW3,SW2
100    FORMAT('      SWITCH1:',I1,'      SWITCH3:',I1,'      SWITCH2:',I1)
C      SCREEN POSITION LINE 18, COLUMN 0
      WRITE(*,'(AA\)' ) CHAR(27),'Y2 '

C      SET OR RESET PROMPT FOR FINISH DEPENDING ON SW1
      IF (SW1.EQ. 1) THEN
        CALL FIN(C)
      ELSE
        WRITE(*,'(AA\)' ) CHAR(27),'L'
      ENDIF

      IF ((C.EQ. 'Y') .OR. (C.EQ. 'y')) THEN
        I=1
      ENDIF
1000   CONTINUE

C      CURSOR VISIBLE
      WRITE(*,'(AA)' ) CHAR(27),'@1'
      END

      SUBROUTINE FIN (C)
C      PROMPT FOR FINISH
      CHARACTER C
      WRITE(*,'(A\)' ) '      DO YOU WANT TO FINISH?      Y/N '
      READ(*,'(A1)' ) C
      RETURN
      END

```

BASIC Interface

To use the Mouse procedures in a BASIC program, declare
USES MOUSE

The subroutine and parameter definition can be found below:

GMOUS

```
SUB GMOUS(SW3,SW2,SW1,Y,X)
    INTEGER SW1, SW2, SW3, Y, X
```

SETPOS

```
SUB SETPOS (X,Y)
    INTEGER X,Y
```

SETMAX

```
SUB SETMAX (XMAX,YMAX)
    INTEGER XMAX,YMAX
```

Example

Here is an example for the use of the Mouse procedures in BASIC.

In the line starting with 10, the maximum Mouse position is set by means of the X and Y coordinates. In the line starting with 20, the present Mouse position is set to zero. In the line of the main program starting with 100, the variables are collected and displayed on the screen.

SW1, SW2, and SW3 designate the switches on the Mouse (here the HAWLEY MARK II Mouse), which can be on or off. X is the Mouse position on the X coordinate; y the Mouse position on the Y coordinate.

```

REM
REM  MOUSE TEST PROGRAM  B A S I C
REM

USES MOUSE

INTEGER SW1,SW2,SW3,X,Y,X0,Y0,XM,YM
DIM C$*1

SUB FIN (C$)
REM PROMPT FOR FINISH
INPUT AT (19,1) " DO YOU WANT TO FINISH?  Y/N ":C$
SUBEND

REM CLEAR SCREEN, CURSOR INVISIBLE
DISPLAY ERASE ALL:CHR$(27);"@0"

DISPLAY AT (2,1): " *****  M O U S E  E X A M P L E  B A S I C  *****
DISPLAY AT (17,1):" TO STOP PRESS SWITCH 1 "

REM SET MOUSE MAX VAL
XM=10000
YM=10000
10 CALL SETMAX(XM,YM)

REM SET MOUSE POSITION
X0=0
Y0=0
20 CALL SETPOS(X0,Y0)

C$="N"

REM LOOP MOUSE ACTION
FOR I=0 TO 0 STEP 0
100 CALL GMOUS(SW3,SW2,SW1,Y,X)
DISPLAY AT(5,1) USING " X=##### Y=#####":X,Y
DISPLAY AT(7,1):" SWITCH 1:";SW1;" SWITCH 3:";SW3;" SWITCH 2:";S

REM SET OR CLEAR PROMPT FOR FINISH DEPENDING ON SW1
IF SW1=1 THEN CALL FIN(C$) ELSE DISPLAY AT (19,1):CHR$(27);"L";

IF C$="Y" OR C$="y" THEN I=1
NEXT I

1000 REM CURSOR VISIBLE
DISPLAY AT (1,1):CHR$(27);"@1"
END

```



KIT INSTALLATION

for

MEMORY EXPANSION

(K202-V001)

The attached pages provide information for installing this kit into an NCR DECISION MATE V. These pages should be filed in your NCR DECISION MATE V User Information Manual.

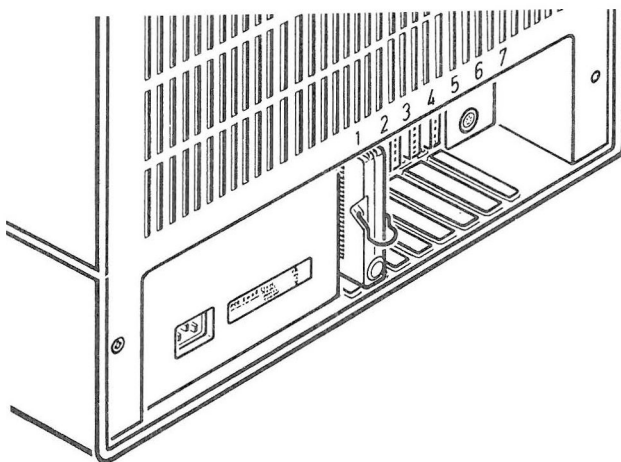
KIT INSTALLATION

MEMORY EXPANSION (K202-V001)

1. Install this kit into slot 1 at the rear of the NCR DECISION MATE V.

NOTE: No other slot position may be used for this kit.

You have now increased the memory capacity of your system to 256 Kilobytes.



IMPORTANT INFORMATION FOR MS-DOS USERS WITH 256 KB MEMORY

Do not use the /HIGH switch when linking a program with MS-LINK on systems with 256 KB memory. (The switch can never be used with PASCAL and FORTRAN programs, regardless of memory size.) While MS-LINK performs successfully, the computer “goes down” when MS-DOS attempts to load the .EXE program into memory.

If using interpretive BASIC, you can store a subroutine in high memory, but you should only use this technique if you are familiar with debugger:

1. Debug subroutine
2. Move subroutine to higher memory
3. Record register values (segment address of subroutine)
4. Load BASIC
5. Load BASIC program
6. Change DEF SEG to refer to subroutine
7. Save subroutine (BSAVE)

Finally, use the following sequence in programs that compute the size of the user area. This sequence ensures a successful program load.

1. Load the application program.
2. Save segment address of program (in CS).
3. Issue the System Call, “Modify allocated memory block”:
 - move CS to ES
 - move length of program to BX
 - move 4AH to AH
 - INT 21H
4. Issue the System Call “Allocate memory”:
 - move FFFFH to BX
 - move 48H to AH
 - INT 21H
5. Multiply the contents of BX by 16 to determine the size of the user area.

(For detailed information see the MS-DOS Programmer’s Manual, “System Calls” chapter.)

NCR DECISION MATE V**SPEICHERERWEITERUNG
(K202-V001)**

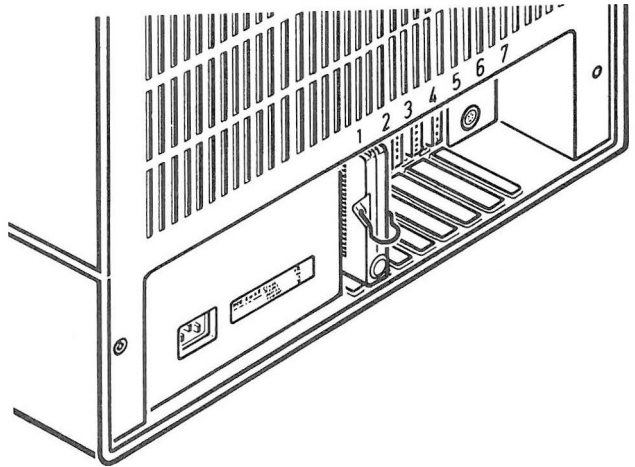
Die beiliegende Seite erklärt Ihnen, wie Sie diese Leistungserweiterung an Ihren NCR DECISION MATE V anschließen können. Bitte ordnen Sie diese Beschreibung in Ihre Bedienungsanleitung für den NCR DECISION MATE V ein.

NCR ist ständig bemüht, die Produkte im Zuge der Entwicklung von Technologie, Bauteilen, Soft- und Firmware dem neuesten Stand anzupassen. NCR behält sich deshalb das Recht vor, Spezifikationen ohne vorherige Ankündigung zu ändern. Nicht alle hier beschriebenen Leistungen werden von NCR in allen Teilen der Welt vertrieben. Nähere Informationen bezüglich eventueller Einschränkungen oder Erweiterungen sowie den aktuellen Stand erfahren Sie von Ihrem Händler oder der nächstgelegenen NCR-Geschäftsstelle.

**SPEICHERERWEITERUNG
(K202-V001)**

Setzen Sie dieses Erweiterungsmodul in die dafür vorgesehene Steckfassung 1 an der Rückseite Ihres NCR DECISION MATE V ein.

Beachten Sie bitte, daß diese Erweiterung in keine andere Steckfassung gesteckt werden darf.



Sie haben damit die Speicherkapazität Ihres NCR DECISION MATE V auf 256 Kilobyte erweitert.

WICHTIGER HINWEIS FÜR DIE BENUTZUNG VON MS-DOS MIT 256KB-SPEICHER

Wenn Ihr System einen 256 KB-Speicher beinhaltet, sollten Sie bei Verwendung des MS-LINK-Dienstprogramms die /HIGH-Option nicht benutzen. (Diese Option kann ohnehin mit PASCAL- und FORTRAN-Programmen unabhängig von der verfügbaren Speichergröße nicht benutzt werden.) Der MS-LINK-Vorgang wird wie üblich durchgeführt. Sobald MS-DOS aber versucht, das maschinenausführbare .EXE-Programm in den Speicher zu laden, ist ein Neubetätigen des Netzschalters am Computer erforderlich.

Es ist möglich, ein Unterprogramm, das vom BASIC-Interpreter übersetzt werden soll, in den oberen Bereich des Speichers zu laden. Dies ist aber nur dann zu empfehlen, wenn Sie bereits mit der Funktionsweise der Programm-Testhilfe (debugger) vertraut sind:

1. Vergewissern Sie sich, daß Ihr Unterprogramm keine Fehler enthält.
2. Übertragen Sie das Unterprogramm in den oberen Speicherbereich.
3. Notieren Sie die neue Segmentadresse entsprechend der neuen Anfangsadresse des Unterprogramms.
4. Laden Sie den BASIC-Interpreter.
5. Laden Sie das BASIC-Programm.
6. Ändern Sie die DEF SEG-Instruktion, damit sie auf die neue Anfangsadresse zeigt.
7. Übertragen Sie das Unterprogramm auf Platte (BSAVE).

Die nachstehende Erläuterung sollten Sie lesen, wenn Sie Programme einsetzen wollen, die die Größe des Benutzerbereichs berechnen. Die folgenden Anweisungen sorgen dafür, daß das Programm einwandfrei geladen wird.

1. Laden Sie das Anwender-Programm.
2. Kopieren Sie den Inhalt von CS (Segmentadresse des Programms) in eine Speicherstelle (2 Byte), wo er nicht versehentlich gelöscht werden kann.
3. Rufen Sie die System-Funktion "Modify allocated memory block" auf, indem Sie folgende Anweisungen ausführen:
 - Übertragen Sie den Inhalt von CS in ES
 - Laden Sie BX mit der Länge des Programms

- Laden Sie das AH-Register mit 4AH
 - INT 21H
4. Rufen Sie die System-Funktion “Allocate memory” auf, indem Sie folgende Anweisungen ausführen:
- Laden Sie BX mit FFFFH
 - Laden Sie das AH-Register mit 48H
 - INT 21H
5. Multiplizieren Sie den Inhalt von BX mit 16, um die Größe des Benutzerbereichs zu berechnen.

(Das Kapitel “System Calls” im MS-DOS “Programmer’s Manual” enthält eine ausführliche Beschreibung der hier benutzten System-Funktionen.)