INDEX

REF:8052/1017

DATA FOR FIRST TIME USERS OF THE 8052 BASIC SYSTEM

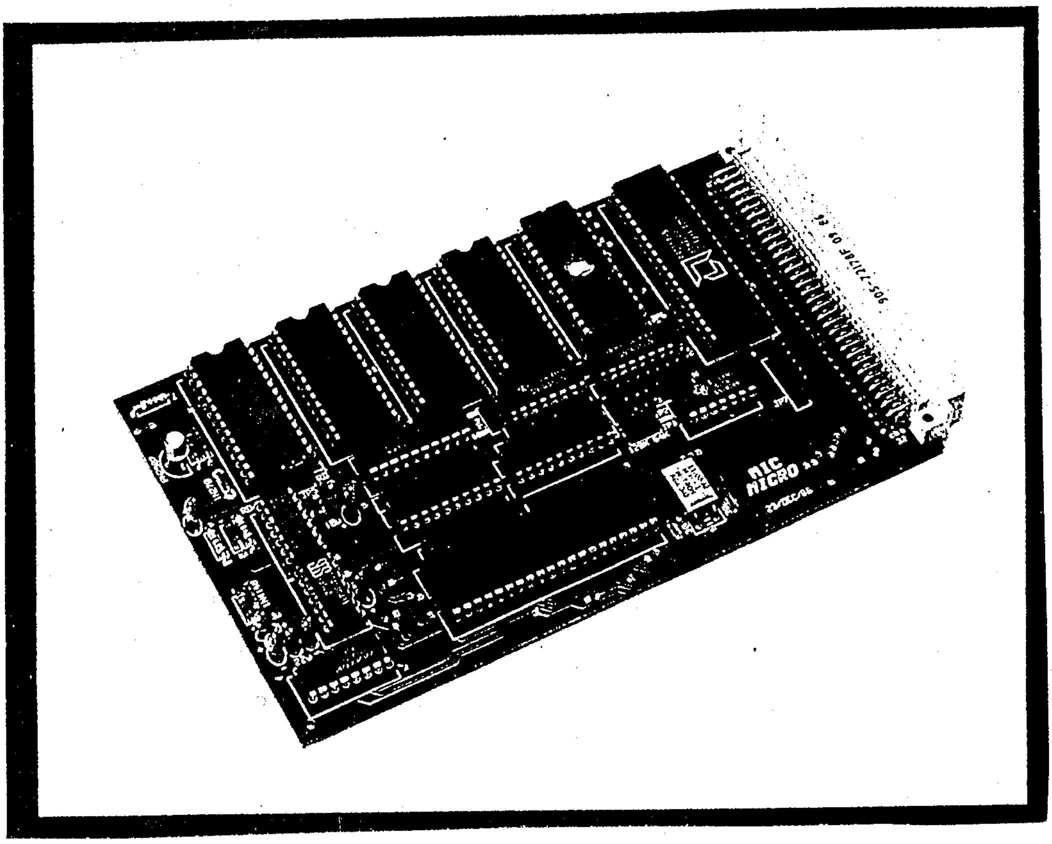
1. Set Baud rate on terminal (any standard Baud rate between 75 and 19200). Set terminal for NO Parity and 1 Stop bit.

2. Switch on terminal.

3. Switch on MIC Micro.

4. Press Space Bar on terminal to get response from board. The 8052 assumes that the first character received is a SPACE and then it sets its Baud rate to match the terminal. Any other characters will result in the MIC Micro printing indistinguishable characters to the terminal.

5. See Page 8 for connection and set up information.



INTRODUCTION

The Mic Micro uses the INTEL 8052 BASIC microcontroller chip which contains an 8K ROM resident BASIC interpreter. The MIC Micro has the following features:

1. 64 K Bytes ROM.
2. 60 K Bytes RAM.
3. Serial Port with AUTO baud rate selection.
4. Choice of RS232, RS422 or RS485 on the console serial port.
5. BASIC Interpreter.
6. Suitable for programming in Basic, PLM-51, C or Assembler.
7. Serial printer port.
8. 3 x 8 Bit Parallel ports.
9. Single 5V supply.

This Basic interpreter is particularly suited for Process Control. The following statements are among its repertoire of instructions;

```
IF - THEN - ELSE
FOR - TO - STEP - NEXT
DO - WHILE/UNTIL
ONTIME
CALL
```

Calculations are handled in Integer or Floating Point maths and are fully supported with Trigonometric and Logical Operators. Due to its low system overhead it is extremely fast and efficient.

Unlike most EPROM programmers which use the entire eprom regardless of program size, the MIC Micro tags each file in eprom and treats it as a mass storage device where as many programs as possible are fitted into the available space. Also, since these programs are stored in directly addressable memory space, as opposed to cassette or disk, they run at full speed.

THE MIC MICRO

The MIC MICRO is a single board controller and/or development system. The board is a standard Eurocard size (100mm x 160mm) and all the connections are made to a 96 way DIN 41612 connector. A dumb terminal or computer with an RS232 port is all that is needed to start your MIC MICRO controller system.

There are five sections to the MIC MICRO board; Processor, Address Decoding and Memory, Parallel I/O, Watchdog timer, Serial I/O and EPROM programmer. Each section is described in detail below.

Processor, Address Decoding and Memory

The MIC MICRO computer/controller board is based on the 8052AH-BASIC chip which is a pre-programmed version on the INTEL 8052AH micro controller. The 8052AH contains 8K bytes of on-chip ROM, 256 bytes of RAM, three 16 bit counter/timers, 6 interrupts and 32 I/O lines. In the 8052AH-BASIC the ROM is a masked BASIC interpreter and the I/O lines are redefined to Address, Data, and Control lines.

The 8052AH-BASIC has a 16 bit address and 8 bit data bus. The 8 least significant address bits (AD0-AD7) and the data bus (D0-D7) are multiplexed together (similar to the 8085 and Z8). When the chip is powered up it sizes consecutive external memory from 0000H to E000H (or memory failure) by alternately writing 55H and 00H to each location. A minimum of 1K bytes of RAM is required for the 8052AH to function and any RAM must be located starting at 0000H.

Three control lines, RD (pin 17), WR (pin 16) and PSEN (pin 29) partition the address space as 64K bytes each of program and data memory. User called Assembly Routines must be in program memory space (read using PSEN and not RD). The addressing logic is as follows;

1. The RD and the WR Pins are used to enable RAM memory from 0000H to F000H. Address A15 is used to decode the chip select (CS) for the devices and RD and WR are used to enable the OE and WE or (WR) pins respectively.

2. If the on board 8255 is selected, RAM memory is limited to E000H.

3. RAM Space above F000H is reserved for external I/O.

4. PSEN is used to enable EPROM memory from 0000H to FFFFH. Address A15 is used to decode the chip select (CS) for the devices and PSEN is used to enable the OE pin.

The 8052AH reserves the first 512 bytes of External Data Memory to implement two 'software' stacks. These are the Control Stack and the Arithmetic Stack or the Argument Stack. Understanding how the stacks work is only necessary if the user wishes to link BASIC and 8052 Assembly Language routines. The details of how to link assembly language are covered in the Assembly Language Linkage section of the MCS User's Manual.

The Control Stack occupies locations 96 (60H) to 254 (0FEH) in external RAM memory. This memory is used to store all information associated with loop control (ie. DO-WHILE, DO-UNTIL and FOR-NEXT) and BASIC subroutines (GOSUB). The stack is initialised to 254 (0FEH) and 'grows down'.

The Argument Stack occupies locations 301 (12DH) to 510 (1FEH) in external RAM memory. This stack stores all constants that BASIC is currently using. Operations such as Add, Subtract, Multiply and Divide always operate on the first two numbers on the Argument Stack and return the result to the Argument Stack. The Argument Stack is initialised to 510 (1FEH) and 'grows down' as more values are placed on the Argument Stack. Each floating point number placed on the Argument Stack requires 6 Bytes of storage.

The Stack Pointer on the 8052AH (Special Function Register, SP) is initialised to 77 (4DH). The 8052AH's Stack Pointer 'grows up' as values are placed on the stack.

Parallel I/O

The MIC MICRO contains an 8255 PIA which provides three 8 bit input/ output software configurable parallel ports. The three I/O ports are labelled A, B and C.

Jumper JPI is used to enable or disable the 8255. When the 8255 is selected, its address is fixed at E000H. When JPI is used to select pin 12 of U13 the 8255 is disabled and RAM memory address space on board is increased to F000H. See Table 'A' for more information.

Watchdog Timer

The MIC Micro contains a Max 690 watchdog timer. This timer has two modes of operation

a) Mode 1 - Monitor VCC

In this mode the MAX 690 monitors the supply voltage. Should the voltage drop below 4.65 or rise above 5.25 Volts, the MAX 690 will issue a 50 mS reset pulse to pin 9 of U4.

b) Mode 2 - WDI & VCC Monitor

In this mode VCC is still monitored as in a) above. However, if any of the jumpers connected to pin 6 are enabled, the MAX 690 enters the watchdog mode. (Please note that only <u>ONE</u> of 'cons out', 'cons in' or 'aux ser' may be used as WDI to max 690). In this mode the MAX 690 will issue a 50mS reset pulse if ;

[i] Supply goes out of range [see (a) above].
[ii] The WDI input is held high or low for more than 1,6 seconds.

Serial I/O

There are two serial ports on the MIC Micro board. One is for the console I/O terminal and the other is an Auxiliary serial output, frequently referred to as the line printer port. When using an 11.0592 MHz crystal, the console port does auto baud rate determination on power up (a pre-set baud rate can be alternatively stored in EPROM as well). It will function at 19200 BAUD with no degradation in operation.

The BAUD [expr] statement is used to set the baud rate for the line printer port. In order for this statement to calculate the Baud rate, the crystal (special function operator - XTAL) must be correctly assigned;[e.g. XTAL = 9000000]. BASIC assumes a crystal value of 11.0592 MHz if no XTAL value is assigned.

The main purpose of the software line printer port is to let the user make a "hard copy" of programme listings and/or data. The command, LIST# and PRINT# direct output to the software line printer port. If the BAUD [expr] statement is not executed before LIST# or PRINT# command statement is entered, the output to the software line printer port will be at about 1 BAUD and it will take a long time to print data. It is necessary to assign a Baud rate to the software printer port before using LIST# or PRINT#. The maximum Baud rate that can be assigned by the BAUD statement depends on the crystal, but 1200BAUD is a reasonable maximum rate, unless a very fast printer or a printer with a large buffer is used.

The MIC Micro serial ports can be connected via RS232 (3 wire), RS485 (2 wire) or RS422 (4 wire). Operation is as follows ;

[a] RS232

A MAX232 driver converts the TTL logic levels from the console and line printer ports to RS232C.

[b] RS422 and RS485

U17 and U18 embody the differential line driver chips. The "Molex connector" on the schematic is labelled for an RS422 configuration using two wires as the TX bus and two wires as the RX bus.

Please note that the RS485/RS422 protocol allows multiple listeners on one line. However, only one speaker is permitted at a time. For this reason, in almost all applications, pin 2 of U18 will be jumpered to connect to pin 7 of U1 (P1.6) On the RS485/RS422 chips, holding pin 2 High places the chip in the Transmit mode, while pulling the pin Low forces the chip into the Receiver mode.

Typical RS485 configuration

Only U18 is required in this configuration. U17 is not used. Connect direction jumper (JP5) to select P1.6. Transmission and reception is done via U18.

Set P1.6 High to Transmit

Set P1.6 Low to Receive

Software default should set pin Low to enable multiple devices to be connected together.

EPROM Programmer

One of the powerful features of the MIC Micro is that it has the ability to execute and save programs in an EPROM. The 8052AH chip actually generates all of the timing signals needed to program EPROMS. Saving programs in EPROM is a much more attractive and reliable alternative to cassette tape, especially in control and/or noisy environments.

Port 1, Bit 4 (U1 pin 5) is used to provide a 1 or 50 millisecond programming pulse. The length of the programming pulse depends upon whether INTEL brand fast program EPROMs or generic brand EPROMs are being programmed. The 8052 calculates the length of the programming pulse from the assigned crystal value. The accuracy of this pulse is within 10 CPU clock cycles. This pin is normally in a logical High (1) state. It is asserted Low to programme the EPROMs.

Raising VCC of the EPROM to 6V is automatically provided on board, to program with the use of the Intelligent Algorithm.

Port 1, bit 5 (U1 pin 6) is used to enable the EPROM programming voltage. This pin is normally in a logical High (1) state. Prior to the EPROM programming operation, this pin is brought to a logical Low state. This pin is used to turn the programming voltage ON and OFF (12,5 or 21 volts).

N O T E: The 12,5 or 21V programming voltage should only be applied after the board is switched on and should be disconnected before the board is switched off.

The 8052 will save a single program on the EPROM if the size of the program and the EPROM are the same. However, if the programs are short it will store any number of them until the memory space is filled. The programs are stored sequentially in the EPROM and any program can be retrieved and executed. This sequential storing of programs is referred to as the EPROM FILE. The full set of file commands is detailed in the MCS 8052AH User's Manual.

****

SET UP INFORMATION

The following notes are intended to assist in operating the MIC Micro.

1. Connection to terminal

Terminal		MIC Micro
Transmitted data	(Pin 2)	30c
Received data	(Pin 3)	29c
Ground	(Pin 7)	32abc

Terminal settings

BAUD rate — Any standard BAUD rate between 110 and 9600

Parity - 0
Wordlength - 8 bit

2. Connection to printer

MIC Micro	Printer
30a	Received data [Pin 3]
32abc	Ground [Pin 7]

Printer settings

BAUD rate — Any standard BAUD rate between 110 and 1200 (higher BAUD rates not practical unless buffer is used).

Parity - OFF
Wordlength - 8 bit

3. Power Supply Requirements

Power requirements for the MIC Micro are as follows ;
5 volts +/- 5% at 350mA (CMOS versions +/- 200mA).
12,5 or 21 volts +/- 2% at 30mA. (Required only for EPROM programming).

****

4. GENERAL NOTES

a A quick release EPROM socket (normally found on an EPROM programmer) can be fitted into the EPROM socket at address 8000H. A 28 pin solder socket of the low cost type should be sandwiched between the quick release socket and the socket in the PC board. This low cost socket will accept the 'sideways' pins of the quick release EPROM socket and also provide the necessary stand off height.

b. Battery backed up RAMs will work in the board without any modifications. (RAM sockets only).

c. Assembly routines

NOTE: The BASIC instruction CALL 4000H looks for the assembly coding in CODE (EPROM) memory and not in RAM memory.

XBY (ØEØØ3H) = 8ØH

8ØH sets all 8255 ports to mode Ø output.

TABLE 'A'

PARALLEL I/O

When the power is turned on, the 8255 is in an unknown configuration. Before the ports can be used they must be initialized by loading a Control Word into the Control Register. For example, the BASIC statement XBY (0E003H) = 80H will load the value 80H into the Control Register. The value 80H sets all three ports to mode 0 output operation (bit 7 of the Control Register must always be set to logic 1). At this point, 8 bit values can be directed to the specific ports e.g. XBY(0E001H) = 56H means write 56H to Port B).

set 3 ports to mode 0 out

write to port

All three ports can be configured as mode 0 inputs by loading 9BH as the control word. The command is XBY (0E003H) = 9BH. Reading the value of input port A then becomes PRINT XBY (0E000H). The following is a list of control word values for some typical 8255 port configurations. To use any of them, simply load the control register address with the XBY command:

8255 (IC2) Port Configuration

Control Word Value	Port A	Port B	Port C
80H	Output	Output	Output
89H	Output	Output	Input
82H	Output	Input	Output
8BH	Output	Input	Input
92H	Input	Input	Output
99H	Input	Output	Input
90H	Input	Output	Output

These are some of the many port configurations available. Please see the INTEL reference to the 8255 for the full set of commands and handshaking modes.

****

TABLE 'B'

JUMPER SETTINGS

Jumper Name	Function
J1	'64K'- External data memory is enabled to F000H. 'CS-PIA'-Memory is limited to E000H and the 8255 is fixed at address E000H.
JP5	'GND'. - U18 is permanently in the receive mode 'P1.1' - The direction pin of U18 is controlled from Port 1.6 of the processor.
JP6	'RS485'. - In this position the RX pin of U18 is selected. 'RS422' - the RX pin of U17 is selected.
JP7	'PSEN' - the PSEN signal is used as a chip select signal to the EPROM at 8000H(U9). 'RD/PSEN' - The ANDED READ and PSEN signal is used as a chip select signal to the EPROM at 8000H(U9).
JP8	'WR' - The write line is used as the write signal to the RAM chip at 8000H(U11). 'PGM PULSE' - The program pulse signal is used as the write signal to the RAM chip at 8000H(U11).
JP9	'RD' - the READ signal is used as a chip select signal to the RAM at 8000H(U11). 'RD/PSEN' - The ANDED READ and PSEN signal is used as a chip select signal to the RAM at 8000H(U11).
8 Pin Jumper	Select watchdog input source to MAX690. 1. Console out 2. Spare 3. Auxilliary serial out 4. Console input

TABLE B (CONT)

NOTES:

1. Eprom Programming:

a) Eprom to be programmed must be in Eprom Socket 8000H(U9).

b) JP7 must be set to RD/PSEN.

c) RAM Socket U11 must not be populated i.e. maximum RAM available when programming an Eprom is 32K.

2. 64K RAM AND 64K EPROM

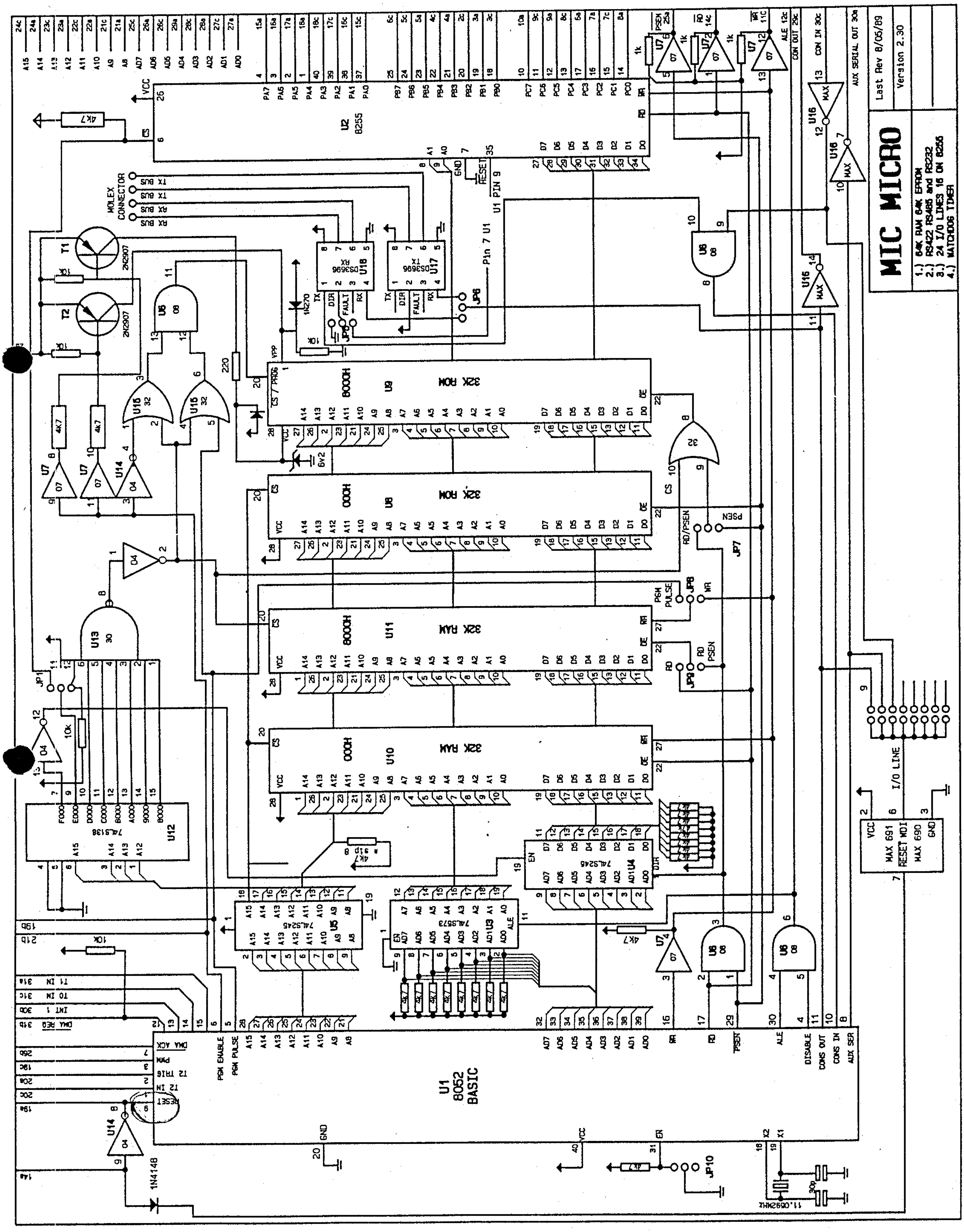
JP7 must be set to PSEN.
JP9 must be set to RD.
JP8 must be set to WR.

3. CMOS SYSTEM

An 80C32 can be used with Basic in an External EPROM.

NOTE:

The EPROM Programming function does not work in this configuration.


MIC MICRO
Last Rev 8/05/89
Version 2.30
1.) 64K RAM 64K EPROM
2.) RS422 RS485 and RS232
3.) 24 I/O LINES 16 ON 8255
4.) WATCHDOG TIMER
U1 8052 BASIC
32K ROM
32K RAM
MOLEX CONNECTOR
I/O LINE

PARTS LIST

MIC 1017

DESIGNATION		DESCRIPTION
ICS & IC SOCKETS		
U 1	40 pin	P8052 AH-BASIC
U 2	40 pin	D8255 AC-2
U 3	20 pin	74LS573N
U 4	20 pin	74LS245N
U 5	20 pin	74LS245N
U 6	14 pin	74LS08N
U 7	14 pin	74LS07N
U 8	28 pin	32K Rom
U 9	28 pin	32K Rom
U10	28 pin	32K Ram
U11	28 pin	32K Ram
U12	16 pin	74LS138N
U13	14 pin	74LS30
U14	14 pin	74LS04N
U15	14 pin	74LS32N
U16	16 pin	MAX232
U17	8 pin	DS3696N
U18	8 pin	DS3696N
U19	8 pin	MAX690

RESISTORS

3 x 4k7
6 x 10k
3 x 1k
1 x 220
1 x 66

3 x 4k7 Sip

CAPACITORS

4 x 10 uf 25V Tant
4 x 22 uf Tant
2 x 30 uf

PARTS LIST (CONT)

TRANSISTORS

2 x 2N 2907

CRYSTALS

1 x 11.0592MHZ

DIODES

1 x 6V2 Zener
1 x 1N277
2 x 1N4148

CONNECTORS

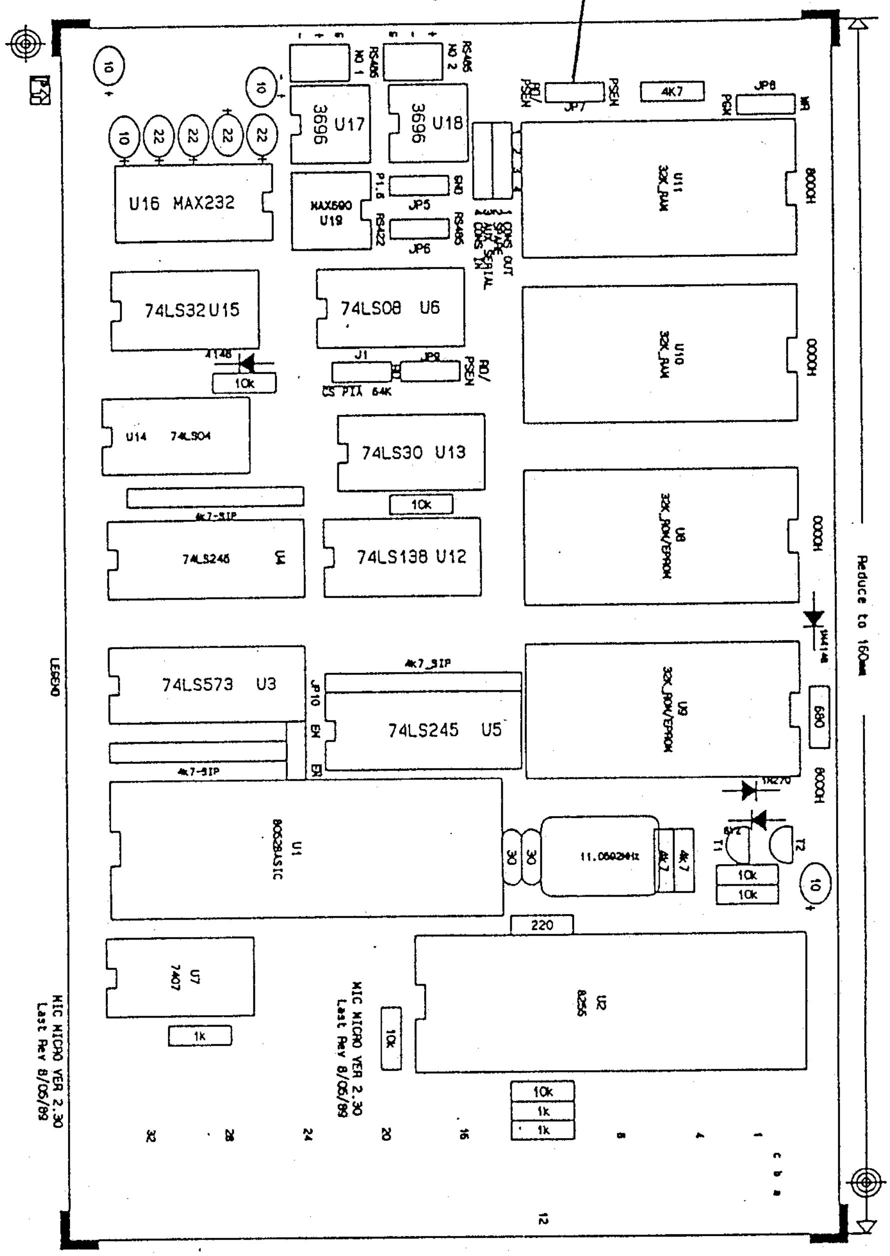
2 x 3pin Molex
1 x 96 way Right angle Din 41612

JUMPERS

J1	1 x 3 pin Header
J2	1 x 3 pin Header
J5	1 x 3 pin Header
J6	1 x 3 pin Header
J7	1 x 3 pin Header
J8	1 x 3 pin Header
J9	1 x 4 pin Double Row Header

Set to RD/PSEN

1+G G1+



Remove When Programming Gprom

Gprom To Be Programmed

MURTON INDUSTRIAL CONTROLS (PTY) LTD

Tel: (011) 315-0688
315-2318
Fax: (011) 315-1110

PRICE LIST:

No. 5 With effect from 1st August 1989

MIC MICRO'S	Stock No.	RANDS
A. MIC 1010		
1. FULLY SOLDERED I.E. COMPLETE WITH IC SOCKETS, RESISTORS, CRYSTAL CAPACITORS. NO IC'S	MIC1010	198
2. WITH IC'S AND 8K RAM	MIC1010	398
3. FULL HOUSE - 32K RAM/16 EPROM	MIC1010	495
B. MIC 1017		
1. FULLY SOLDERED I.E. COMPLETE WITH IC SOCKETS, RESISTORS, CRYSTAL CAPACITORS. NO IC'S	MIC1017	198
2. WITH IC'S AND 32K RAM	MIC1017	495
3. FULL HOUSE 64K RAM; 64K ~~RAM~~ EPROM	MIC1017	595
C. MCS-BASIC 52 USER'S MANUAL (INTEL)		35
NOTE: MIC MANUAL INCLUDED WITH ALL BOARDS		NO CHARGE

MIC PARALLEL I/O (USES 3 X 8255)

FULL HOUSE	MIC1031	243

MIC A/D BOARD

A. 12 BIT A/D [USES ADC 1205]	MIC1014	~~320~~ 355
B. 12 BIT A/D AND D/A BOARD	MIC1024	350
C. 8 BIT A/D BOARD	MIC1009	350
D. 8 BIT A/D AND D/A BOARD	MIC1009	434

MIC SERIAL I/O BOARDS

SERIAL CARD (USES 16450 OR 8250)	MIC1013	337

MIC KEYBOARD/DISPLAY BOARD

A. FULL HOUSE PC BOARD	MIC1011	270
B. 4 X 20 LCD DISPLAY (NEW TYPE - WIDE VIEWING ANGLE)	MIC1011A	240
C. 8 X 40 LCD DISPLAY (LM 2138) Price on application	MIC1011B	P.O.A.
D. MEMBRANE KEYBOARD - FULL CHARACTER SET	MIC1011C	70
E. KEYPAD; 0 - 9 NUMERICAL; A - F ALPHA	MIC1011D	38
E. 4 X 20 DISPLAY WITHOUT BACKLIGHT	MIC1011E	192

		Stock No.	RANDS
MIC 4 1/2 DIGIT DISPLAY			
	7 SEGMENT, 4 1/2 DIGIT DISPLAY BOARD	MIC1036	160
MIC MOTHERBOARDS			
A.	3 SLOT	MIC1006	230
B.	6 SLOT	MIC1004	290
MIC MODEM			
	SUITABLE FOR RADIO TELEPHONE OR LEASED LINE	MIC1037	258
MIC GPIB/IEEE 488 (USES TMS 9914 - TEXAS)			
	FULL HOUSE	MIC1012	296
MIC POWER SUPPLIES (LOCALLY MANUFACTURED)			
A.	5 VOLT 5 AMP EUROCARD TYPE COMING SOON	MIC1008	260
B.	+/- 12 VOLT 1 AMP EUROCARD TYPE	MIC1007	260
C.	TRIPLE OUTPUT +5V 1A AND +/- 12 V 200mA	MIC1018	260
DELTA POWER SUPPLIES (PC BOARD MODULES)			
A.	1000 5 - 6V 1A	U5	PRICES ON APPLICATION
B.	500 12 - 15V 0,5A	U15	
C.	200 +/- 12 - 15V 0,2A OR 24 - 30V 0,2A	UD15	
D.	EUROCARD FOR 2X MODULES (INC DIN CONNECTOR)	P300	
DELTA POWER SUPPLIES (EUROCARD TYPE)			
A.	5 - 6V 5A	30S5	PRICES ON APPLICATION
B.	5 - 6V 12A	60S5	
C.	15 - 15 2 X 12 - 15V 1,1A	30S15	
D.	15 - 15 2 X 12 - 15V 2,5A	75S15	
E.	5V 6A 2 X 12 1A	ST60	
SOFTWARE (ASSEMBLY LANGUAGE SUPPLIED IN PRE-PROGRAMMED EPROM)			
A.	BASIC EXTENSIONS AND CROSS ASSEMBLER	MIC1010A	75
B.	BASIC EXTENSIONS, CROSS ASSEMBLER, BASIC, KEYBOARD & SMARTWATCH ROUTINES	MIC1017A	89
C.	DISPLAY AND KEYBOARD ROUTINES PLUS SMARTWATCH ROUTINES	MIC1010B	30
D.	SMARTWATCH SOCKET FOR MIC1010 & MIC1017	MIC1010D	110

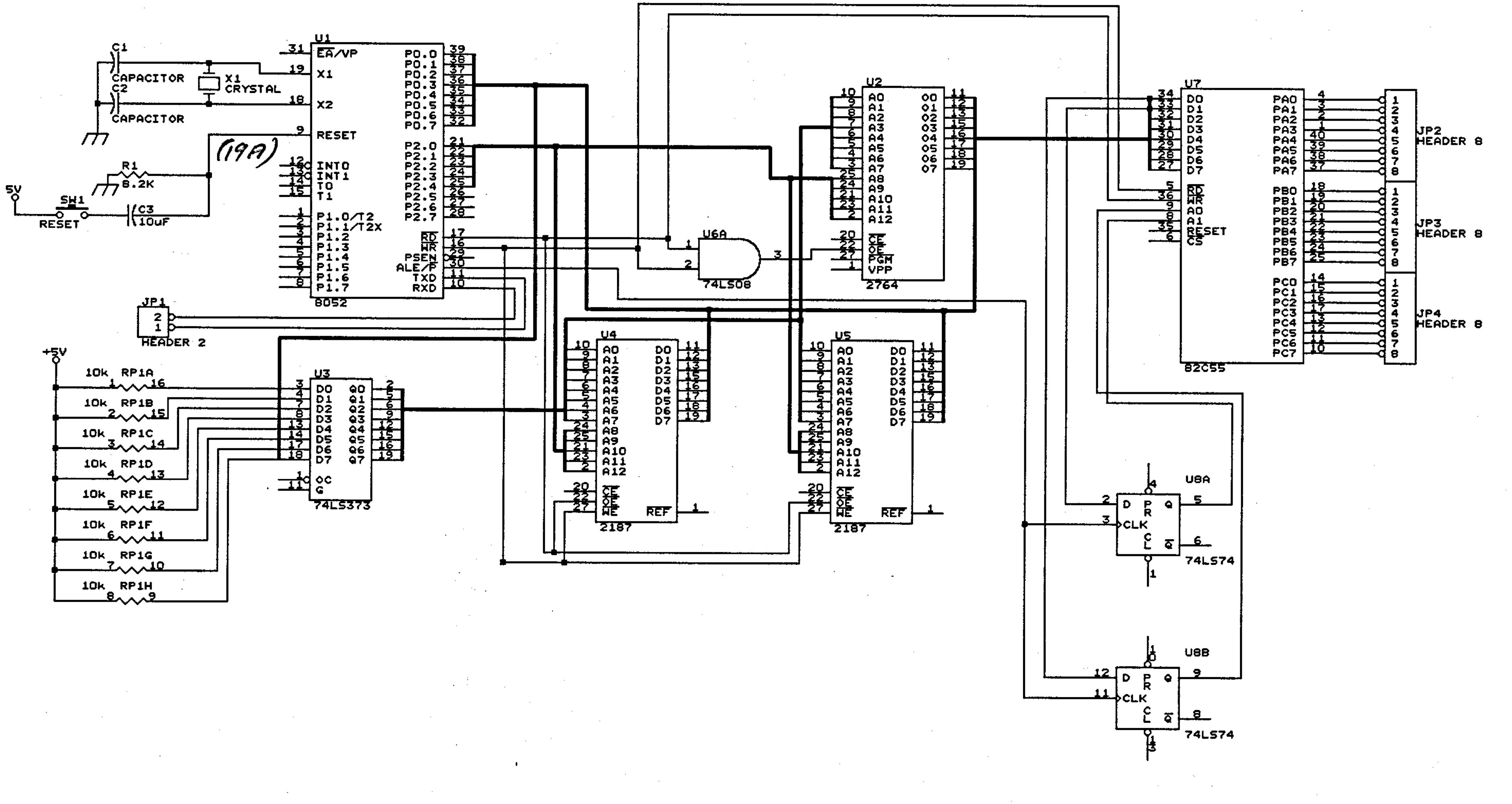
NOTE: STOCK NO'S GENERICALLY CALLED MIC1010 ARE USED IN CONJUNCTION WITH MICROPROCESSOR BOARDS MIC1010. THERE IS ONLY 1 EPROM FOR MIC1017 WHICH CONTAINS ALL SOFTWARE (MIC1017A).

			Stock No.		RANDS
PC INTERFACE CARDS (PLUG INTO PC EXPANSION SLOT)					
A.	RS485 INTERFACE CARD		MIC1015		315
B.	RS232 INTERFACE CARD	COMING SOON	MIC1041	APPROX	210
C.	FIBRE OPTIC CARD	COMING SOON	MIC1042	APPROX	385
PROTOTYPE BOARDS					
A.	100 X 160mm EUROCARD		MIC1038		35
B.	IBM PC PROTOTYPING CARD WITH ADDRESS DECODING		MIC1043	APPROX	280
EXTENDER CARDS					
A.	96 WAY EURO EXTENDER CARD		MIC1025		88
B.	IBM PC XT/AT EXTENDER CARD		MIC1044	APPROX	170
ZERO INSERTION FORCE SOCKETS					
A.	28 PIN ZERO INSERTION FORCE SOCKET		MIC1010F		48
B.	40 PIN ZERO INSERTION FORCE SOCKET		MIC1010F		65
HARDWARE INTERFACE CARDS					
20 OPTO COUPLED INPUTS & 16 RELAY OUTPUTS			MIC1039		350
MIC ISOLATED I/O SYSTEMS					
1.	I/O MODULES (AC OR DC)				40
2.	I/O MODULE MOUNTING BASES				
	a. 4 POSITION				80
	b. 8 POSITION				160

The above prices do not include General Sales Tax.
Where applicable please add R10,00 for insured postage.

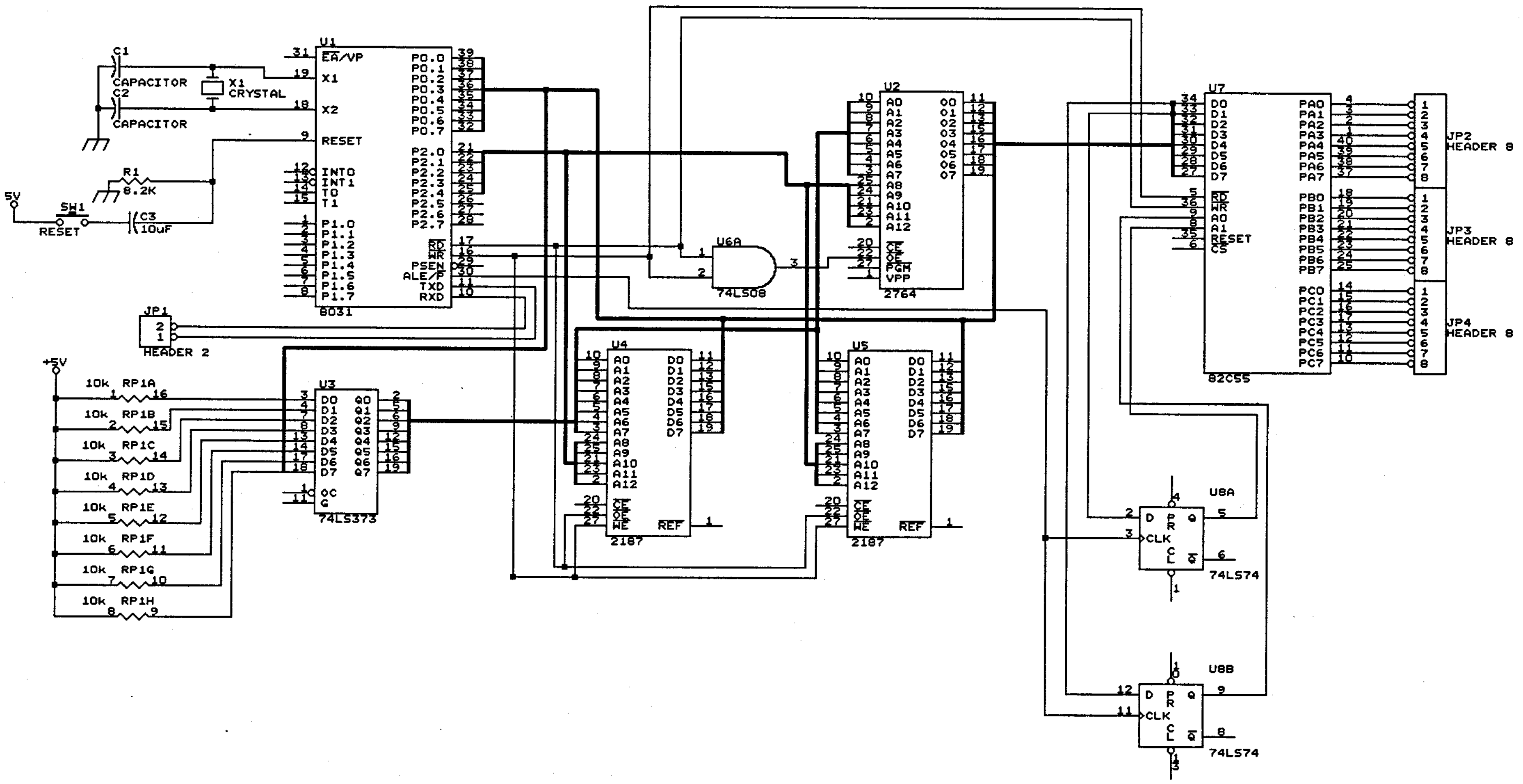
PRICES SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

MICROTRONIX SYSTEM CONTROLLER



ELECTRONIC DESIGN AND COMMUNICATION SERVICES			
Title	8031 BASED SYSTEM CONTROL UNIT		
Size	Document Number		RE
B	2		
Date:	November 30, 1991	Sheet	1 of

MICROTRONIX SYSTEM CONTROLLER



ELECTRONIC DESIGN AND COMMUNICATION SERVICES		
Title: 8031 BASED SYSTEM CONTROL UNIT		
Size: B	Document Number: 2	RE
Date: November 30, 1991	Sheet 1 of	

```
1000 REM        ************************************************
1010 REM        **        8052-AH BASIC VER. 1.1 ( INTEL)    **
1020 REM        **                                            **
1030 REM        **               R O M C O P Y                **
1040 REM        **                      by B.Stander          **
1050 REM        **                                            **
1060 REM        **       THIS PROGRAM TRANSFERS THE BASIC     **
1070 REM        **     INTERPRETER IN THE ROM BURNED IN THE   **
1080 REM        **    8052AH-BASIC PROCESSOR TO A 2764 EPROM  **
1090 REM        **    BY USING THE ON-BOARD EPROM PROGRAMMER. **
1100 REM        **    INTERPRETER 8052 IC:       0000H-1FFFH  **
1110 REM        **    RAM ADDRESS ON CARD:       2000H-3FFFH  **
1120 REM        ** EPROM ADDRESS ON CARD (2764): 8000H-9FFFH  **
1130 REM        **--------------------------------------------**
1140 REM        **      8052-AH BASIC   Copywrite by INTEL    **
1150 REM        ** For further information refer to Intel's MCS **
1160 REM        **          BASIC - 52 User's Manual          **
1170 REM        **--------------------------------------------**
1180 REM        **       THIS PROGRAM IS THE PROPERTY OF      **
1190 REM        **                                            **
1200 REM        **            M I C R O T R O N I X           **
1210 REM        **                   Tel: 011-674-4477        **
1220 REM        ************************************************
1230 REM
1240 PRINT "** PART 1: MOVE INTERPRETER TO RAM AT 2000H-3FFFH"
1250 PRINT : PRINT
1260 MTOP=1FFFH:                  REM : RESERVE RAM FROM 2000H
1270 DIM A(15):                   REM : DIMENSION ARRAY 1x16
1280 FOR X=0000H TO 1FFFH STEP 16: REM : SET ROM ADDRESSING
1290    FOR Y = 0 TO 15:          REM : IN BLOCKS OF 16 BYTES
1300       A(Y) = CBY(X + Y):     REM : READ 1 BYTE ROM, PLACE IN ARRAY
1310       Z=X+Y+2000H:           REM : COMPUTE CORRESPONDING RAM ADDR.
1320       XBY(Z) = A(Y):         REM : WRITE BYTE IN RAM
1330       B = XBY(Z):            REM : READ IT BACK FROM RAM
1340                              REM : AND COMPARE WITH BYTE IN ROM
1350                              REM : ERROR? => REPORT AND STOP
1360       IF A(Y)<> THEN ? "PROGRAM ERROR ,ROM= ",A,"RAM=",B:END
1370    NEXT Y:                   REM : 16 BYTE LOOP
1380                              REM : SHOW 16-BYTE LINE ON TERMINAL
1390    PH1. X,"     ",:PH0. A(0),A(1),A(2),A(3),A(4),A(5),A(6),A(7),
1400    PH0. A(8), A(9), A(10), A(11), A(12), A(13), A(14), A(15)
1410 NEXT X:                      REM : LOOP FOR ENTIRE ROM 0000H-1FFFH
1430 PRINT "** PART2: SET INTERNAL MEMORY FOR PROGRAMMING PROCEDURE"
1450                              REM : INTERPRETER (size = 8 Kbyte)
1460                              REM : NOW IN RAM FROM 2000H TO 3FFFH
1470                              REM : THE EPROM (also 8 Kbytes)
1480                              REM : OCCUPIES 8000H TO 9FFFH
1490 DBY(18H)=0FFH:               REM : TARGET ADDRESS LOW BYTE-1
1500 DBY(19H)=00H:                REM : SOURCE ADDRESS LOW BYTE
1510 DBY(1AH)=7FH:                REM : TARGET ADDRESS HIGH BYTE
1520 DBY(1BH)=20H:                REM : SOURCE ADDRESS HIGH BYTE
1530 DBY(1EH)=00H:                REM : NUMBER OF BYTES LOW ADDRESS
1540 DBY(1FH)=20H:                REM : NUMBER OF BYTES HIGH ADDRESS
1550 W = .05:                     REM : W=WIDTH OF PROGRAMMING PULSE IN S
1560 R = 65536 - W * XTAL / 12:   REM : R=RELOAD VALUE
1570 DBY(40H)=R/256:              REM : SET VALUE OF R (LOW BYTE)
1580 DBY(41H)=R .AND. 0FFH:       REM : SET VALUE OF R (HIGH BYTE)
1590 DBY(38)=DBY(38) .AND. 0F7H:  REM : RESET BIT 38.3 FOR 50 ms PULSES
1600 PRINT:PRINT;PRINT SPC(25),"** SETTING OKAY **":PRINT:PRINT
1610
1620 PRINT "** PART 3: PROGRAMMING THE EPROM"
1630
1640 FOR R = 1 TO 5: PRINT : NEXT R
```

```
1650 PRINT "** SWITCH ON THE PROGRAMMING VOLTAGE": PRINT
1660 PRINT "** BE SURE TO APPLY THE CORRECT VOLTAGE 12.5 OR 21 VOLT!": PRINT
1670 FOR R = 1 TO 3: PRINT : NEXT R
1680 PRINT "PRESS ENTER TO START PROGRAMMING"
1690 PRINT "OR PRESS ESCAPE TO STOP": PRINT : PRINT
1700 PRINT "PLEASE TYPE <ENTER> OR <ESC>",
1710 K=GET:IF K=0 THEN 1710
1720 IF K=1BH THEN PRINT:END:       REM ** 1BH is ASCII-code for Esc
1730 IF K<>0DH THEN 1710:           REM ** 0DH is ASCII-code for Enter
1740 PRINT : PRINT : PRINT "BUSY PROGRAMMING EPROM"
1750 PRINT : PRINT "THIS WILL TAKE ABOUT 7 MINUTES": PRINT
1760
1770 PGM                            REM ** PROGRAMMING INSTRUCTION **
1780 PRINT
1790 PRINT : PRINT "** PART 4: CHECKING INTERNAL POINTERS FOR ERRORS"
1800 PRINT
1810 H=DBY(1AH):L=DBY(18H):HL=H*256+L
1820 IF (DBY(30).OR.DBY(31)) <>0 THEN 1830 ELSE 1840
1830 PRINT "INCORRECT PROGRAMMING OF EPROM AT ADDRESS", : PH1.HL: END
1840 PRINT "PROGRAMMING FINISHED      ** NO ERRORS **": PRINT
1850 PRINT
1860 PRINT "** PART 5: DIRECT COMPARISON BETWEEN ROM AND EPROM": PRINT
1870 PRINT
1880 FOR X=0000H TO 1FFFH          :REM ** ADDRESS IN ROM
1890    Y=X+8000H                  :REM ** ADDRESS IN EPROM
1900    A = CBY(X): B = XBY(Y): REM ** READ OUT RAM AND EPROM
1910    PRINT"ROM",:PH1.X,:PRINT" =>",:PH0.A" =",
1920    PH0.B,:PRINT" ,<",:PH1.Y,:PRINT" EPROM",CR,
1930    IF A <> B THEN PRINT "EPROM ERROR": END
1940 NEXT X: PRINT : PRINT
1950 PRINT "EPROM CORRECTLY PROGRAMMED": PRINT : PRINT
1960 PRINT "PROGRAM FINISHED, BYE !!!"
1970 END
```