

PART I :

MDE-8086 USER'S MANUAL

TABLE OF CONTENTS

1. MDE-8086 SYSTEM CONFIGURATION.....	1
2. OPERATION INTRODUCTION.....	4
2-1. Kind & Function of Key.....	4
2-2. Basic Operation.....	5
3. EXAMPLE PROGRAM.....	10
4. SERIAL MONITOR.....	18
4-1. How to use serial monitor ?.....	18
4-2. How to install to Hard-disk ?.....	18
4-3. If pressing the RES key ?.....	19
4-4. How to connect computer ?.....	19
4-5. Operation serial monitor command.....	23
5. SERIAL MONITOR EXPERIMENT.....	29
6. 8086 INTERRUPT SYSTEM.....	35
6-1. Predefined interrupts(0 to 4).....	36
6-2. Interrupt experiment.....	37
6-3. User-defined software interrupts.....	39
6-4. 8259A interrupt control.....	40
7. 8253 INTERFACE.....	41

APPENDIX

MDE-8086 APPENDIX

TABLE OF CONTENTS

1. MDE-8086 MEMORY CIRCUIT	99
2. MDE-8086 CONNECTOR(CON10) CIRCUIT	100
3. 8086 PIN CONFIGURATION.....	101
4. 8086 INSTRUCTION SET SUMMARY.....	102

PART II :

MDE-8086 EXPERIMENTS (SOFTWARE/HARDWARE)

TABLE OF CONTENTS

Experiment 1. 8255A INTERFACE.....	47
1-1. LED & 7-Segment.....	47
Experiment 2. DOT-MATRIX LED.....	50
2-1. Dot-Matrix LED display.....	50
2-2. Dot-Matrix LED interface.....	51
2-3. Speaker interface.....	53
Experiment 3. 8251A INTERFACE.....	56
Experiment 4. LCD DISPLAY.....	60
4-1. LCD.....	60
4-2. LCD interface.....	64
Experiment 5. KEYBOARD INTERFACE.....	67
5-1. Keyboard interface.....	67
Experiment 6. D/A CONVERTER.....	70
6-1. D/A converter specification.....	70
6-2. D/A converter interface.....	72
6-3. D/A converter experiment.....	73
Experiment 7. A/D CONVERTER.....	75
7-1. A/D converter specification.....	75
7-2. A/D converter interface.....	77
7-3. A/D converter experiment.....	77
Experiment 8. STEPPING MOTOR CONTROL.....	83
8-1. Stepping motor specification.....	83
8-2. Stepping motor interface.....	86
8-3. Stepping motor experiment.....	87

1. MDE-8086 SYSTEM CONFIGURATION

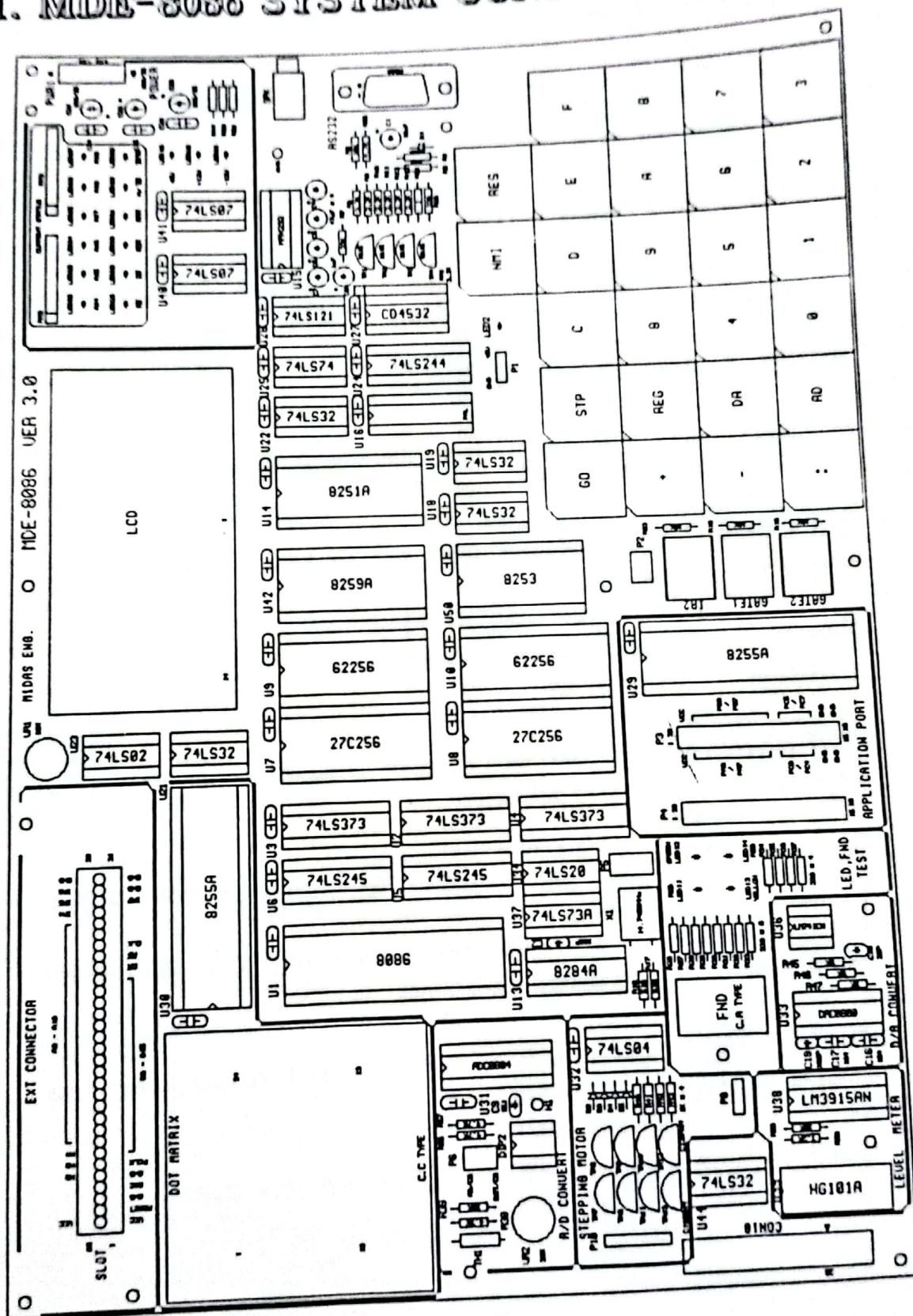


FIGURE 1. MDE-8086 System configuration.

2. 1. MDE-8086 SYSTEM CONFIGURATION

The function of ICs at Figure 1.

- ① CPU(Central processing unit) : Using Intel 8086. Using 4.9152Mhz.
- ② ROM(Read Only Memory) : It has program to control user's key input, LCD display, user's program. 64K Byte, it has data communication program.
Range of ROM Address is F0000~FFFFF11,
- ③ SRAM(Static Random Access Memory) : Input user's program & data.
Address of memory is 0000011~0FFFF11, totally 64K Byte.
- ④ DISPLAY : It is LCD, 16(Character)×2(Line)
- ⑤ KEYBOARD : It is used to input machine language and has 16 of hexa-decimal keys and 8 of function keys.
- ⑥ SPEAKER : Able to test sound using with speaker and further more able to test synthesizer.
- ⑦ RS-232C : It is ready to do data communication with IBM compatible personal computer.
- ⑧ DOT MATRIX LED : To understand & test of dot matrix structure and principle of display, it is interfaced to 8255A(PPI).
- ⑨ A/D CONVERTER : Convert analog signal to digital signal using with ADC0804.
- ⑩ D/A CONVERTER : Convert digital signal to analog signal using with DAC0800 and it is interfaced so as to more Level meter.
- ⑪ STEPPING MOTOR INTERFACE : So as to control stepping motor driver circuit of stepping motor is interfaced.
- ⑫ POWER : AC 110~220V, DC +5V 3A, +12V 1A, -12V 0.5A SMPS.

MDE-8086 ADDRESS MAP**① Memory map**

ADDRESS	MEMORY	DESCRIPTION
00000H ~ 0FFFFH	RAM	PROGRAM & DATA MEMORY
F0000H ~ FFFFFH	ROM	MONITOR ROM
10000H ~ EFFFFH	USER'S RANGE	

② I/O address map

ADDRESS	I/O PORT	DESCRIPTION
00H ~ 07H	LCM & KEYBOARD	LCD Display 00H : INSTRUCTION REGISTER 02H : STATUS REGISTER 04H : DATA REGISTER KEYBOARD 01H : KEYBOARD REGISTER (Only read) 01H : KEYBOARD FLAG (Only write)
08H ~ 0FH	8251 / 8253	8251(Using to data communication) 08H : DATA REGISTER 0AH : INSTRUCTION / STATUS REGISTER 8253(TIMER/COUNTER) 09H : TIMER 0 REGISTER 0BH : TIMER 1 REGISTER 0DH : TIMER 2 REGISTER 0FH : CONTROL REGISTER
10H ~ 17H	8259/SPEAKER	8259(Interrupt controller) 10H : COMMAND REGISTER 12H : DATA REGISTER SPEAKER → 11H : SPEAKER
18H ~ 1FH	8255A-CS1/ 8255A-CS2	8255A-CS1(DOT & ADC INTERFACE) 18H : A PORT DATA REGISTER 1AH : B PORT DATA REGISTER 1CH : C PORT CONTROL REGISTER 8255-CS2(LED & STEPPING MOTOR) 19H : A PORT DATA REGISTER 1BH : B PORT DATA REGISTER 1DH : C PORT CONTROL REGISTER 1FH : CONTROL REGISTER
20H ~ 2FH	I/O EXTEND CONNECTOR	
30H ~ FFH	USER'S RANGE	

2. OPERATION INTRODUCTION

2-1. KIND & FUNCTION OF KEY

MDE-8086 has high performance 64K-byte monitor program. It is designed for easy function. After power is on, the monitor begins to work. In addition to all the key function the monitor has a memory checking routine.

The following is a simple description of the key functions.

FUNCTION KEY DATA KEY

GO	STP	C	D	E	F
+	REG	8	9	A	B
-	DA	4	5	6	7
:	AD	0	1	2	3

RES system reset.

STP execute user's program, a single step.

AD set memory address.

GO go to user's program or execute monitor functions.

DA Update segment & Offset and input data to memory.

MON Immediately break user's program and Non makable interrupt.

: Offset set.

REG Register Display.

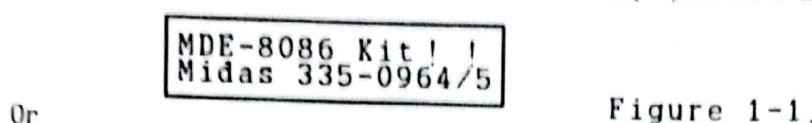
+ Segment & Offset +1 increment.
Register display increment.

- Segment & Offset -1 decrement.
Register display decrement.

2-2 BASIC OPERATION 6

2-2. BASIC OPERATION

On a power-up, following message will be displayed on a LCD.



Or

Figure 1-1.

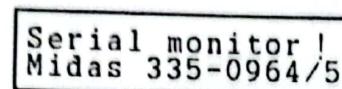
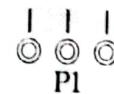


Figure 1-2.

So as to use serial monitor, move jumper P1 which located on the PCB like this.



KEYBOARD

Machine code

KEYBOARD

Serial monitor

► **RES** System Reset Key

Whenever RES is pressed, the display becomes FIGURE 1-1 and user can operate keyboard only in this situation.

⑥ 2 OPERATION INTRODUCTION

▶ **[AD]**, **[:]**, HEXA-DIGIT KEY : Substitute to segment & offset address.

EXAMPLE 1) Check the contents in memory.

KEY

[AD]

LCD

Seg.	Oset	data
0000	1000	FF

↓ ↓ ↓
Input data Offset [The contents of memory 0000:1000
(It may be different)]

F

Seg.	Oset	data
000F	1000	FF

↓ ↓ ↓
Input data Offset [The contents of memory 000F:1000
(It may be different)]

0

Seg.	Oset	data
00F0	1000	FF

↓ ↓ ↓
Input data Offset [The contents of memory 00F0:1000
(It may be different)]

0

Seg.	Oset	data
0F00	1000	FF

↓ ↓ ↓
Input data Offset [The contents of memory 0F00:1000
(It may be different)]

0

Seg.	Oset	data
F000	1000	FF

↓ ↓ ↓
Input data Offset [The contents of memory F000:1000
(It may be different)]

2-2. BASIC OPERATION 7

:

Seg.	Oset	data
F000	1000	FF

↓ ↓ ↓
Segment Offset data
[The contents of memory F000:1000
(It may be different)]

0

Seg.	Oset	data
F000	0000	EB

↓ ↓ ↓
Input data Offset data
[The contents of memory F000:0000]

▷ , , KEY : Increment and decrement to segment & offset address.

If on a power-up or pressing RES key, following message will be displayed on LCD.

MDE-8086 Kit!!
Midas 335-0964/5

If on a AD key,

KEY

LCD

Seg.	Oset	data
0000	1000	FF

↓ ↓ ↓
Segment Offset data
[The contents of memory 0000:1000
(It may be different)]

Seg.	Oset	data
0001	1000	FF

↓ ↓
Segment +1 increment data
[The contents of memory 0001:1000
(It may be different)]

④ 2. OPERATION INTRODUCTION

[+]

Seg 0002	Oset 1000	data FF
-------------	--------------	------------

↓
Segment +1 increment
[The contents of memory 0002:1000
(It may be different)]

[-]

Seg 0001	Oset 1000	data FF
-------------	--------------	------------

↓
Segment -1 decrement
[The contents of memory 0001:1000
(It may be different)]

▷ [AD], [:], HEXA-DIGIT KEY : Update to memory contents.

EXAMPLE 2) Let's store the following like to 01000H ~ 01003H
contents.

< ADDRESS DATA >

01000	AB
01001	CD
01002	EF
01003	34

KEY

LCD

[RES]

MDE-8086 Kit!
Midas 335-0964/5

[AD]

Seg. 0000	Oset 1000	data FF
--------------	--------------	------------

↓
Segment Offset
[The contents of memory 0000:1000]

[DA]

Seg. 0000	Oset 1000	data FF
--------------	--------------	------------

↓
Segment Offset
[The contents of memory 0000:1000
(It may be different)]

2-2. BASIC OPERATION 9

A B

Seg.	Oset	data
0000	1000	AB

+

Seg.	Oset	data
0000	1001	FF

↓
Offset increment

C D

Seg.	Oset	data
0000	1001	CD

+

Seg.	Oset	data
0000	1002	FF

↓
Offset increment

E F

Seg.	Oset	data
0000	1002	EF

+

Seg.	Oset	data
0000	1003	FF

↓
Offset increment

3 4

Seg.	Oset	data
0000	1003	34

+

Seg.	Oset	data
0000	1004	FF

↓
Offset increment

▷ REG , + , - KEY : Display to register contents.

KEY

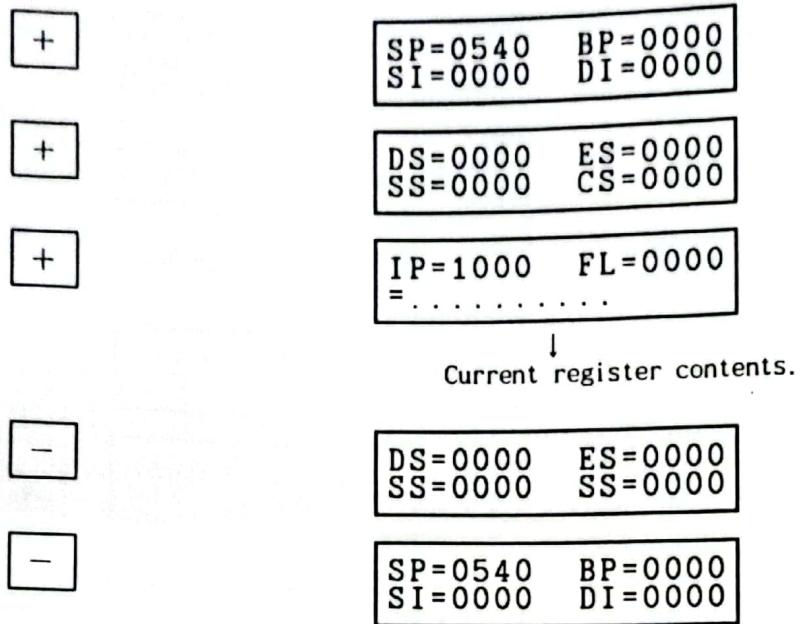
REG

LCD

AX=0000	BX=0000
CX=0000	DX=0000

↓
Current register contents.

10 3. EXAMPLE PROGRAM



3. EXAMPLE PROGRAM

♣ STP Single Step

Store a following program in RAM and execute it by single steps.

<u>ADDRESS</u>	<u>MACHINE CODE</u>	<u>MNEMONIC</u>	
1000	B8 0000	MOV	AX, 0
1003	9E	SAHF	
1004	05 8947	ADD	AX, 4789H
1007	15 8864	ADC	AX, 6488H
100A	04 88	ADD	AL, 88H
100C	80 D4 33	ADC	AH, 33H
		:	
100F	2D 6735	SUB	AX, 3567H
1012	1D 0080	SBB	AX, 8000H
1015	2C 45	SUB	AL, 45H
1017	80 DC 78	SBB	AH, 78H
		:	
101A	B0 FF	MOV	AL, FFH
101C	FE C0	INC	AL
101E	FE C8	DEC	AL
1020	98	CBW	
1021	F6 D8	NEG	AL
		:	
1023	B0 F0	MOV	AL, F0H
1025	B3 11	MOV	BL, 11H

3. EXAMPLE PROGRAM

1027	F6 F3	MUL	BX
1029	B8 0000	MOV	AX, 0000H
1030	B8 3412	MOV	BX, 1234H
1031	F7 FB	IMUL	BX
1031	B8 F000	MOV	AX, 00F0H
1034	B3 10	MOV	BL, 10H
1036	F6 F3	DIV	BL
1038	BA FFFF	MOV	BX, -1
1038	B8 FFFF	MOV	AX, -1
103F	B8 0100	MOV	BX, 1
1041	F7 FB	IDIV	BX
1043	CC	INT	3

① Again, using with machine code input program from 1000H

② It is valid only when the display is in current Flag form. Pressing "STP" key causes the CPU to execute one instruction point according to the user's IP. After execution, the monitor regains control and displays the new PC and its contents. The user may examine and modify registers and memory contents after each step.

KEY	LCD
RES	MDE-8086 Kit !! Midas 335-0964/5
AD	Seg 0000 Oset 1000 data B8 ↓ ↓ ↓ Segment Offset [The contents of memory 0000:1000]
STP	(Next address) IP=1003 FL=0100 t..... ↓ Current Flag content (It means single step)

12.3. EXAMPLE PROGRAM

- Result verify ! + AX=0000 BX=0000
 CX=0000 DX=0000
- ↓
Current Register content
- (Next address)
- 2) SHAF STP IP=1004 FL=0100
 =...t.....
- (Next address)
- 3) ADD AX, 4789H STP IP=1007 FL=0100
 =...t.....
- Result verify ! + AX=4789 BX=0000
 CX=0000 DX=0000
- 4) ADC AX, 6488H STP IP=100A FL=0994
 =o..ts.ap.
- (overflag set, alternate carry set,
sign flag set, parity flag set)
- Result verify ! + AX=AC11 BX=0000
 CX=0000 DX=0000
- 5) ADD AL, 88H STP IP=100C FL=0184
 =o..ts..p.
- (sign flag set, parity flag set)
- Result verify ! + AX=AC99 BX=0000
 CX=0000 DX=0000
- 6) ADC AH, 33H STP IP=100F FL=0180
 =...ts....

Result verify !

AX=DF99	BX=0000
CX=0000	DX=0000

7) SUB AX, 3567H

 STP

(Next address)

IP=1012	FL=0180
=...ts...	

Result verify !

AX=AA32	BX=0000
CX=0000	DX=0000

8) SBB AX, 8000H

 STP

(Next address)

IP=1015	FL=0100
=...t....	

Result verify !

AX=2A32	BX=0000
CX=0000	DX=0000

9) SUB AL, 45H

 STP

(Next address)

IP=1017	FL=0195
=...ts.apc	

Result verify !

AX=2AED	BX=0000
CX=0000	DX=0000

10) SBB AH, 78H

 STP

(Next address)

IP=101A	FL=0185
=...ts..pc	

Result verify !

AX=B1ED	BX=0000
CX=0000	DX=0000

11) MOV AL, FFH

 STP

(Next address)

IP=101C	FL=0185
=...ts..pc	

14 3. EXAMPLE PROGRAM

Result verify !

+

AX=B1FF BX=0000
CX=0000 DX=0000

12) INC AL

STP

(Next address)
IP=101E FL=0155
=...t.zapc

Result verify !

+

AX=B100 BX=0000
CX=0000 DX=0000

13) DEC AL

STP

(Next address)
IP=1020 FL=0195
=...ts.apc

Result verify !

+

AX=B1FF BX=0000
CX=0000 DX=0000

14) CBW

STP

(Next address)
IP=1021 FL=0195
=...ts.apc

Result verify !

+

AX=FFFF BX=0000
CX=0000 DX=0000

15) NEG AL

STP

(Next address)
IP=1023 FL=0111
=...t..a.c

Result verify !

+

AX=FF01 BX=0000
CX=0000 DX=0000

16) MOV AL,FOH

STP

(Next address)
IP=1025 FL=0111
=...t..a.c

3. EXAMPLE PROGRAM 16

Result verify !



AX=FFF0	BX=0000
CX=0000	DX=0000

(Next address)

IP=1027	FL=0111
=...t...a.c	

17) MOV BL, 11H

STP

Result verify !



AX=FFF0	BX=0011
CX=0000	DX=0000

18) MUL BL

STP

(Next address)

IP=1029	FL=0905
=o...t...pc	

Result verify !



AX=0FF0	BX=0011
CX=0000	DX=0000

19) MOV AX, F000H

STP

(Next address)

IP=102C	FL=0905
=o...t...pc	

Result verify !



AX=F000	BX=0011
CX=0000	DX=0000

20) MOV BX, 1234H

STP

(Next address)

IP=102F	FL=0905
=o...t...pc	

Result verify !



AX=F000	BX=1234
CX=0000	DX=0000

21) IMUL BX

STP

(Next address)

IP=1031	FL=0985
=o...ts...pc	

16.3. EXAMPLE PROGRAM

Result verify !	<input type="checkbox"/>	<table border="1"><tr><td>AX=0000</td><td>BX=1234</td></tr><tr><td>CX=0000</td><td>DX=FEDC</td></tr></table>	AX=0000	BX=1234	CX=0000	DX=FEDC
AX=0000	BX=1234					
CX=0000	DX=FEDC					
22) MOV AX, 00FOH	<input type="checkbox"/>	(Next address)				
	STP	<table border="1"><tr><td>IP=1034</td><td>FL=0985</td></tr><tr><td>=... t s .. pc</td><td></td></tr></table>	IP=1034	FL=0985	=... t s .. pc	
IP=1034	FL=0985					
=... t s .. pc						
Result verify !	<input type="checkbox"/>	<table border="1"><tr><td>AX=00F0</td><td>BX=1234</td></tr><tr><td>CX=0000</td><td>DX=FEDC</td></tr></table>	AX=00F0	BX=1234	CX=0000	DX=FEDC
AX=00F0	BX=1234					
CX=0000	DX=FEDC					
23) MOV BL, 10H	<input type="checkbox"/>	(Next address)				
	STP	<table border="1"><tr><td>IP=1036</td><td>FL=0985</td></tr><tr><td>=... t s .. pc</td><td></td></tr></table>	IP=1036	FL=0985	=... t s .. pc	
IP=1036	FL=0985					
=... t s .. pc						
Result verify !	<input type="checkbox"/>	<table border="1"><tr><td>AX=00F0</td><td>BX=1210</td></tr><tr><td>CX=0000</td><td>DX=FEDC</td></tr></table>	AX=00F0	BX=1210	CX=0000	DX=FEDC
AX=00F0	BX=1210					
CX=0000	DX=FEDC					
24) DIV BL	<input type="checkbox"/>	(Next address)				
	STP	<table border="1"><tr><td>IP=1038</td><td>FL=0145</td></tr><tr><td>=... t z .. pc</td><td></td></tr></table>	IP=1038	FL=0145	=... t z .. pc	
IP=1038	FL=0145					
=... t z .. pc						
Result verify !	<input type="checkbox"/>	<table border="1"><tr><td>AX=000F</td><td>BX=1210</td></tr><tr><td>CX=0000</td><td>DX=FEDC</td></tr></table>	AX=000F	BX=1210	CX=0000	DX=FEDC
AX=000F	BX=1210					
CX=0000	DX=FEDC					
25) MOV DX, -1	<input type="checkbox"/>	(Next address)				
	STP	<table border="1"><tr><td>IP=103B</td><td>FL=0145</td></tr><tr><td>=... t z .. pc</td><td></td></tr></table>	IP=103B	FL=0145	=... t z .. pc	
IP=103B	FL=0145					
=... t z .. pc						
Result verify !	<input type="checkbox"/>	<table border="1"><tr><td>AX=000F</td><td>BX=1210</td></tr><tr><td>CX=0000</td><td>DX=FFFF</td></tr></table>	AX=000F	BX=1210	CX=0000	DX=FFFF
AX=000F	BX=1210					
CX=0000	DX=FFFF					
26) MOV AX, -1	<input type="checkbox"/>	(Next address)				
	STP	<table border="1"><tr><td>IP=103E</td><td>FL=0145</td></tr><tr><td>=... t z .. pc</td><td></td></tr></table>	IP=103E	FL=0145	=... t z .. pc	
IP=103E	FL=0145					
=... t z .. pc						

3. EXAMPLE PROGRAM 17

Result verify !



AX=FFFF	BX=1210
CX=0000	DX=FFFF

27) MOV BX, 1

STP

(Next address)

IP=1041	FL=0145
=... t. z. pc	

Result verify !



AX=FFFF	BX=0001
CX=0000	DX=FFFF

28) IDIV BX

STP

(Next address)

IP=1043	FL=0144
=... t. z. p.	

Result verify !



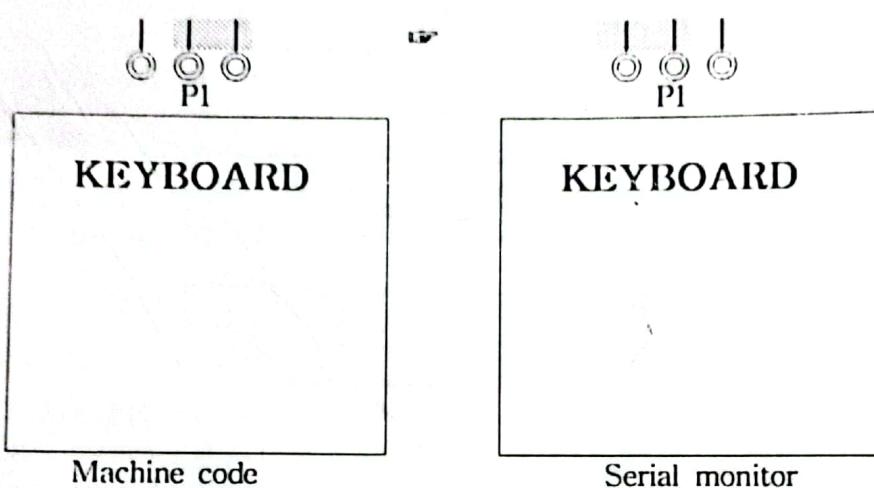
AX=FFFF	BX=0001
CX=0000	DX=0000

4. Serial Monitor

Serial monitor is the basic monitor program to do data communicate between MIDAS-8086 and computer.

4-1. How to use serial monitor?

So as to use serial monitor, move jumper P1 which located on the PCB like this,



4-2. How to install diskette to Hard-disk ?

- ④ Insert diskette to A: driver and input like follows.

A:>INSTALL A: C:

↑ Destination drive

- ⑤ Then the screen will be clear and will be displayed like follow, later install will be completed. If there is a same directory name(8086 in here) at destination drive installation is impossible so change existing directory to other name.

INSTALLING A: TO C:\8086

MIDAS Eng. 1995year

copy now..

4-3. IF PRESSING THE RES KEY ? 19

Wait...

(Later the screen will be clear and following message will be appeared)
Install OK.

⑤ Following message will be displayed at destination drive.(In case of C: drive)

C:\\$086>

⑥ Check total 10 files likes COMM.EXE, MASM.EXE, LOD186.EXE, Etc.

4-3. If pressing the RES Key ?

Move JP1 to serial monitor status and if on a power-up or pressing RES key, following message will be displayed on LCD and data communication is possible with computer.

Serial Monitor !
Midas 335-0964/5

4-4. How to connect computer ?

① The connector of computer RS-232C is 25 pin and RS-232C of MDE-8086 is 9 pin, must be connect like Figure 4-1.

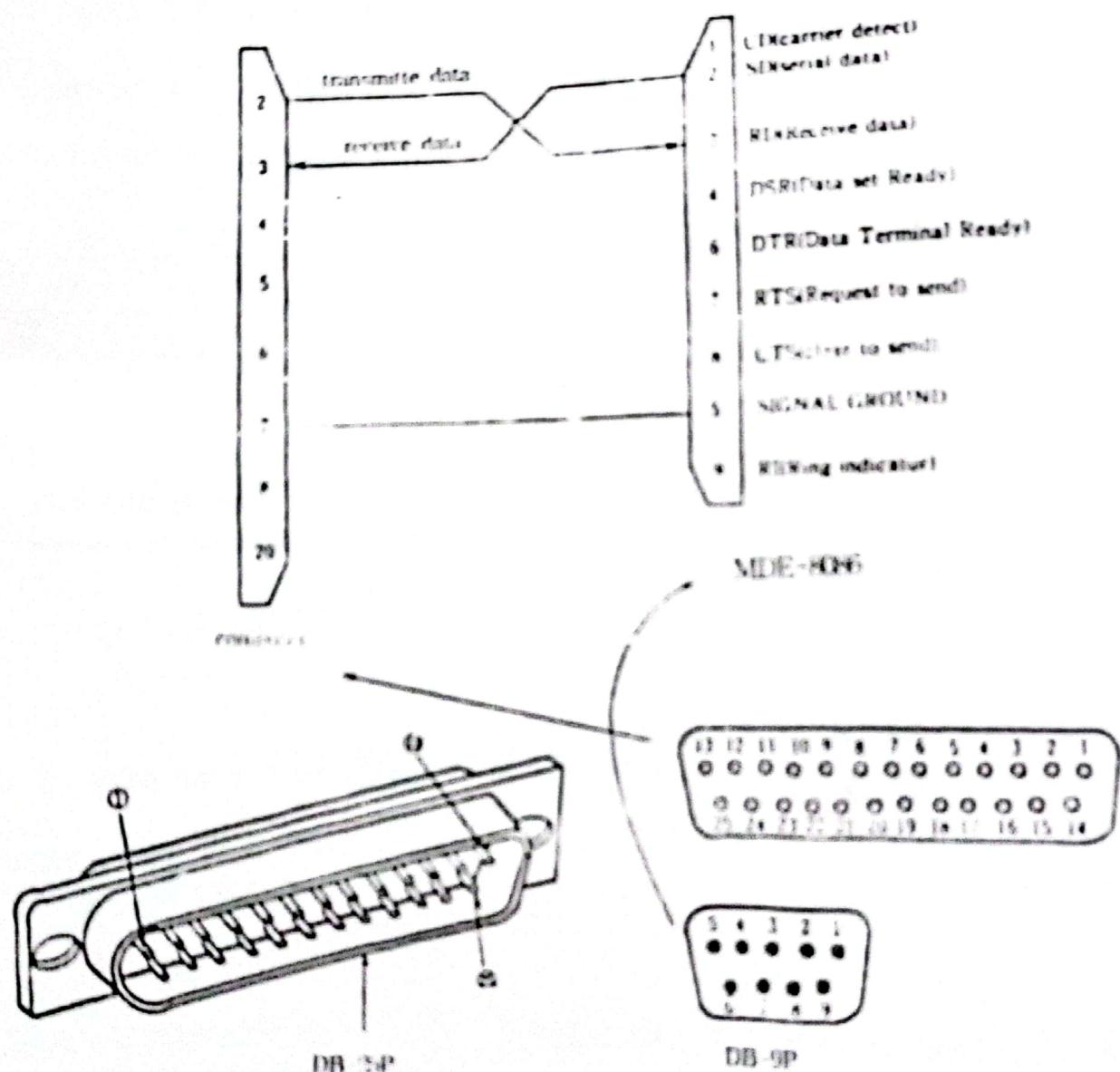


FIGURE 4-1. HOW TO CONNECT CABLE RS-232C

- ② Connect MDE-8086 and RS-232C port of computer like follow Figure 4-2.

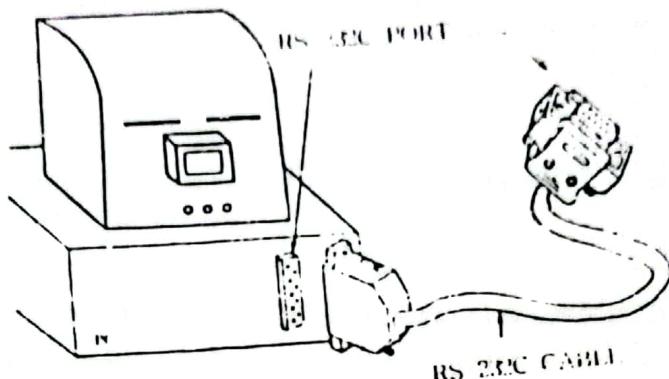


FIGURE 4-2. Connect to MDE-8086 and COMPUTER

- ③ Insert diskette which has COMM.EXE to A: drive of computer, then execute PROCOMM.(Or if you install to hard-drive, execute COMM at 8086 directory)

A:>COMM^②

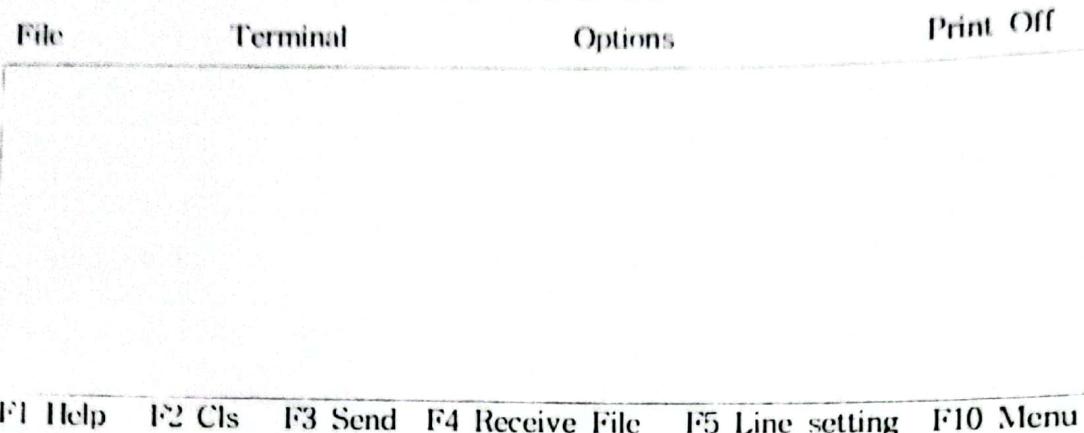
^② : Return Key.

or

C:\8086>COMM^②

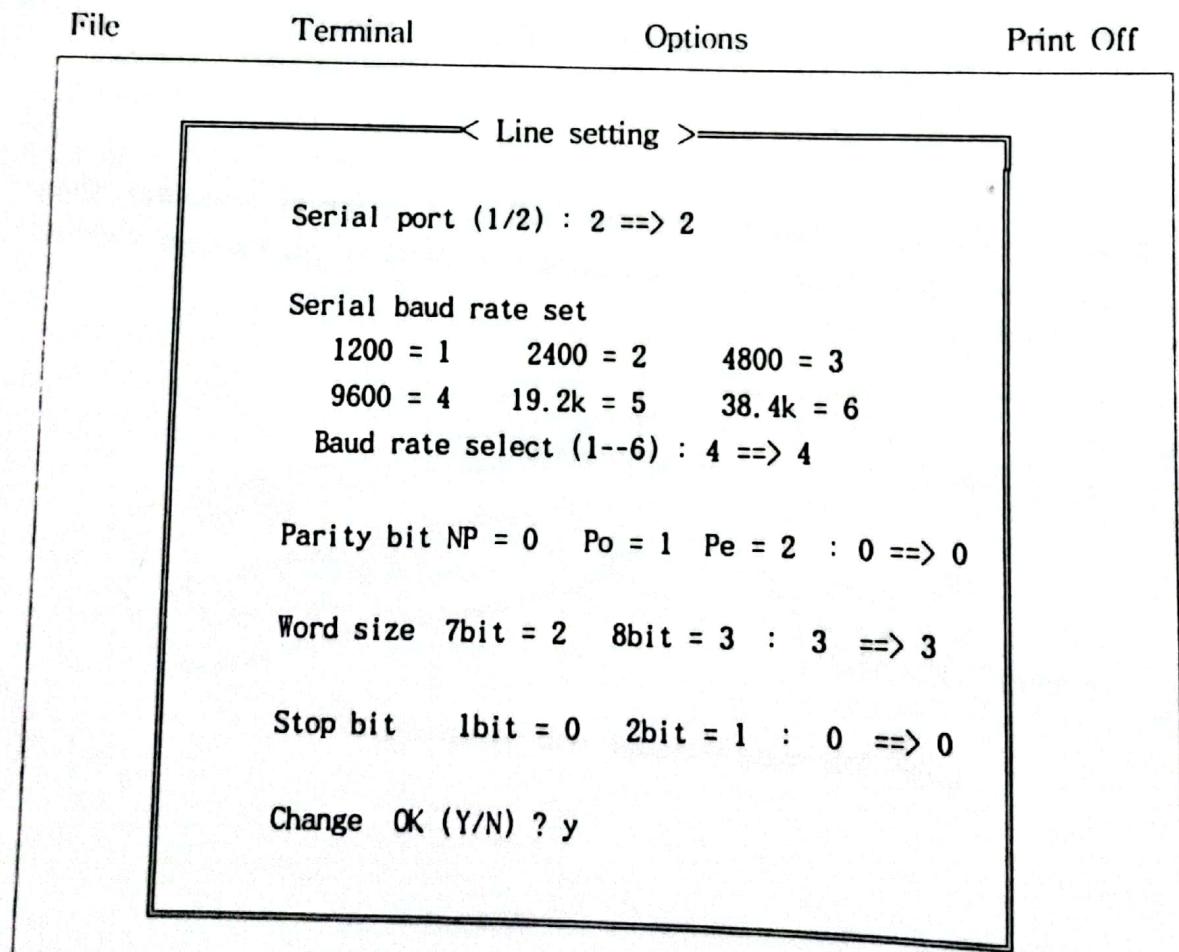
Later following message will be displayed.

22 4. SERIAL MONITOR



F1 Help F2 Cls F3 Send F4 Receive File F5 Line setting F10 Menu

- ④ And, in case data communication between MDE-8086 and computer, need fixing initial of COMM. When you push F5 key, following is displayed and the step of fixing initial is like follow.



F1 Help F2 Cls F3 Send F4 Receive File F5 Line setting F10 Menu

- ⑤ When on a power-up, following message will be displayed. This message is displayed only when on a power-up, and does not display when press RES key.

File	Terminal	Options	Print Off
<pre>** Serial Monitor 1.0 ** ** Midas 335-0964/5 ** 8086 > █ ← MDE-8086 PROMPT</pre>			

F1 Help F2 Cls F3 Send F4 Receive File F5 Line setting F10 Menu

4-5. Operation serial monitor command

User can only use command which stored at serial monitor. Serial monitor can execute to command when user type command and then CR(carriage return) key.

⌘ If there is no any command at serial monitor, error message will be displayed with bell sound and serial monitor prompt will be displayed again.

```
** 8086 Monitor 1.0 **
** Midas 335-0964/5 **
```

```
8086 >█ ← Carriage Return
```

3.4.4. SERIAL MONITOR

8086 >?<

HELP COMMAND	
E segment:offset.....	: Enter Data To Memory
D segment:offset length.....	: Dump Memory Contents
R [register name].....	: Register Display & Change
M address1, length, address2.....	: Move Memory From 1 to 2
F address, length, data.....	: Fill Memory With Any Data
L Return key.....	: Program Down Load
G segment:offset.....	: Execute Program
T.....	: Program 1 step execute

1 Memory modify command.

Segment	Offset
↓	↓
8086 >E 0000:1000	↙
0000:1000 FF ? 11	↙
0000:1001 FF ? 22	↙
0000:1002 FF ? 33	↙
0000:1003 FF ? 44	↙
0000:1004 FF ? 55	↙
0000:1005 FF ? /	↙ ← (Offset decrement)
0000:1004 55 ? /	↙
0000:1003 44 ? .	↙ ← (Escaping command)

2 Memory display command.

3 Memory fill command.

Segment Length Data
 ↓ ↓ ↓
 8086 >F 1000 FF 1234 ↴

☞ Verifying ?

8086 >D 0000:1000 ↴	.4.4.4.4.4.4.4.4
0000:1000 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:1010 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:1020 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:1030 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:1040 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:1050 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:1060 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:1070 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4

8086 >D ↴	.4.4.4.4.4.4.4.4
0000:1080 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:1090 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:10A0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:10B0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:10C0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:10D0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:10E0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:10F0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4

4 Block move command.

The M command is used to move blocks of memory from one area to another.

Segment Length Data
 ↓ ↓ ↓
 8086 >M 1000 100 2000 ↴

☞ Resulting ?

8086 >D 2000 ↴	.4.4.4.4.4.4.4.4
0000:2000 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:2010 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4

PG 4. SERIAL MONITOR

0000:2020 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:2030 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:2040 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:2050 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:2060 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:2070 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
8086 >D	
0000:2080 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:2090 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:20A0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:20B0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:20C0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:20D0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:20E0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4
0000:20F0 12 34 12 34 12 34 12 34 - 12 34 12 34 12 34 12 34	.4.4.4.4.4.4.4.4

5 Display Registers command.

The R command is used to display the i8086 processor registers.

8086 >R

AX=0000	BX=0000	CX=0000	DX=0000
SP=0540	BP=0000	SI=0000	DI=0000
DS=0000	ES=0000	SS=0000	CS=0000
IP=1000	FL=0000	=	

☞ Individual register change

8086 >R AX

AX=0000 1234

BX=0000 4567

CX=0000 7788

DX=0000 1111

SP=0540

23 4. SERIAL MONITOR

8086 >T[■]

AX=1234 BX=4567 CX=7788 DX=1111
SP=0540 BP=0000 SI=0000 DI=0000
DS=0000 ES=0000 SS=0000 CS=0000
IP=1003 FL=0100 = . . . t

↓
Next address

8086 >T[■]

AX=1234 BX=5678 CX=7788 DX=1111
SP=0540 BP=0000 SI=0000 DI=0000
DS=0000 ES=0000 SS=0000 CS=0000
IP=1006 FL=0100 = . . . t

8086 >T[■]

AX=1234 BX=5678 CX=1000 DX=1111
SP=0540 BP=0000 SI=0000 DI=0000
DS=0000 ES=0000 SS=0000 CS=0000
IP=1009 FL=0100 = . . . t

↓
Next address

8086 >T[■]

AX=1234 BX=5678 CX=1000 DX=2000
SP=0540 BP=0000 SI=0000 DI=0000
DS=0000 ES=0000 SS=0000 CS=0000
IP=100C FL=0100 = . . . t

☞ Execute program command

Segment Offset

↓ ↓
8086 >G 0000:1000[■]

Execute Address = 0000:1000

•

•

•

8086 >

8086 >

☞ Resulting ?

8086 >R

```
AX=1234 BX=4567 CX=7788 DX=1111
SP=0540 BP=0000 SI=0000 DI=0000
DS=0000 ES=0000 SS=0000 CS=0000
IP=1000 FL=0000 = . . . . .
```

8086 >R IP

IP=1000

8086 >

6 Program Down load & program execute command.

The L command moves object data in hexa format from an external devices to memory.

8086 >L

Down load start !! ← (Note : See section 5. Serial monitor experiment)

```
:14100000B83412BB7856B90010BA00208BF08BFBB0030BC08
:0910140000408EDA8ED18EC0CCB2
:00
OK Completed !!
```

8086 >

☞ Set IP

8086 >R IP

IP=1000

☞ Executes instructions, one at a time, beginning at the location pointed to by the program counter. After execution of each instruction, the processor registers are displayed.

5. Serial Monitor Experiment.

For develop the program more efficiently, make source file using with editor program of computer then assembling this file and make HEX(Intel file format), down-load to MDE-8086 using with serial monitor. (See the following example program)

II Using with editor program make program like follow.

(Imagine the name of file is EX2.ASM)

```

CODE SEGMENT ← Define the segment
ASSUME CS:CODE, DS:CODE, ES:CODE, SS:CODE
:
SEG_D EQU 0000H : Define the constant
    : Start address
ORG 1000H
MOV AX, SEG_D
MOV DS, AX
MOV ES, AX
:
MOV BX, OFFSET DATA
MOV AX, [BX]
MOV CH, [BX+2]
MOV CL, [BX+3]
:
MOV BP, BX
MOV DX, DS:[BP+4]
:
MOV SI, ES:[BP+6]
MOV DI, [BP+8]
:
MOV BX, WORD PTR DATA+10
INT 3
:
ORG 2000H
DATA DW 3412H

```

30 4. SERIAL MONITOR

```
DW    7856H  
DW    OCDABH  
DW    14FOH  
DW    5368H  
DW    0C4B1H  
DW    2010H  
:  
CODE  ENDS  
END
```

② Set up MASM ASSEMBLER like follows

C:\8086>MASM

Microsoft (R) Macro Assembler Version 5.10

Copyright (C) Misrosoft Corp 1981, 1988. All right reserved.

Source filename [.ASM]:EX2
Object filename [C:EX2.OBJ]:
Source listing [NULL.LST]:EX2
Cross reference [NULL.CRF]:

47838 + 452253 Bytes symbol space free

0 Warning Errors

0 Severe Errors

C:\8086>

③ Make HEX(ABS) file.

C:\8086>LOD186^E

Paragon LOD186 Loader - Version 4.0h
Copyright (C) 1983 - 1986 Microtec Research Inc.
ALL RIGHT RESERVED.

Object/Command File [OBJ]:EX2^EOutput Object File [C:EX2.ABS]:^EMap Filename [C:NUL.MAP]:^E

**LOAD COMPLETE

C:\8086>

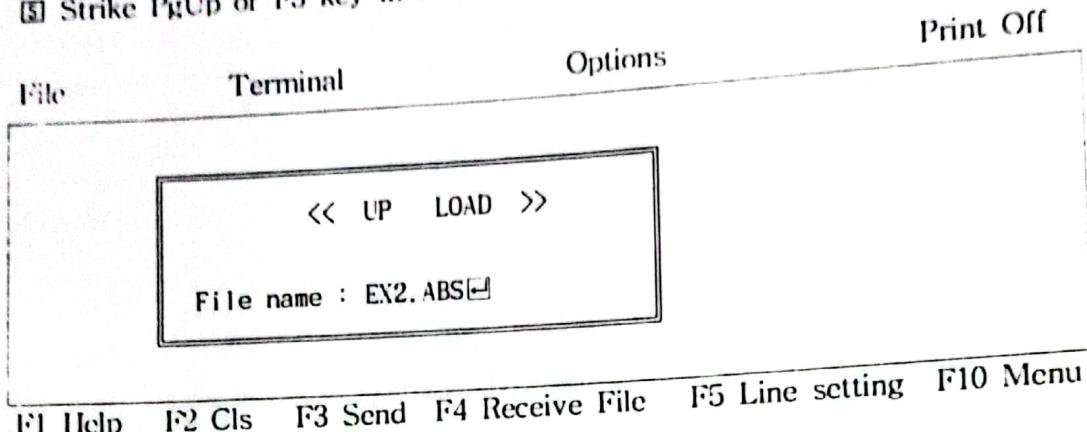
④ Down-load hex file to MDE-8086.

File	Terminal	Options	Print Off
<p>** Serial Monitor 1.0 ** ** Midas 335-0964/5 **</p> <p>8086 >L^E Down load start !!</p>			

F1 Help F2 Cls F3 Send F4 Receive File F5 Line setting F10 Menu

32 4. SERIAL MONITOR

⑤ Strike PgUp or F3 key in computer, and then like following will be displayed.



⑥ Then above message will be disappeared and following message will be displayed.

```
:14100000B800008ED88EC0BB00208B078A6F028A4F038BEBB6  
:101014003E8B5604268B76068B7E088B1E0A20CCCC  
:0E20000012345678ABCDF0146853B1C41020E2  
:00000001FF
```

OK completed

8086 >

⑦ Using T command

8086 >

☞ Set IP

8086 >R IP

IP=1000

8086 >T ← MOV AX,SEG_D

AX=0000 BX=0000 CX=0000 DX=0000

SP=0540 BP=0000 SI=0000 DI=0000

DS=0000 ES=0000 SS=0000 CS=0000

IP=1003 FL=0100 = . . . t

↓
Next address

8086 >T~~1~~ ← MOV DS, AX , MOV ES, AX ,MOV BX,OFFSET DATA

AX=0000 BX=2000 CX=0000 DX=0000
 SP=0540 BP=0000 SI=0000 DI=0000
 DS=0000 ES=0000 SS=0000 CS=0000
 IP=100A FL=0100 = . . . t

8086 >T~~2~~ ← MOV AX, [BX]

AX=1234 BX=2000 CX=0000 DX=0000
 SP=0540 BP=0000 SI=0000 DI=0000
 DS=0000 ES=0000 SS=0000 CS=0000
 IP=100C FL=0100 = . . . t

8086 >T~~3~~ ← MOV CH, [BX+2]

AX=1234 BX=2000 CX=5600 DX=0000
 SP=0540 BP=0000 SI=0000 DI=0000
 DS=0000 ES=0000 SS=0000 CS=0000
 IP=100F FL=0100 = . . . t

8086 >T~~4~~ ← MOV CL, [BX+3]

AX=1234 BX=2000 CX=5678 DX=0000
 SP=0540 BP=0000 SI=0000 DI=0000
 DS=0000 ES=0000 SS=0000 CS=0000
 IP=1012 FL=0100 = . . . t

8086 >T~~5~~ ← MOV BP, BX

AX=1234 BX=2000 CX=5678 DX=0000
 SP=0540 BP=2000 SI=0000 DI=0000
 DS=0000 ES=0000 SS=0000 CS=0000
 IP=1014 FL=0100 = . . . t

8086 >T~~6~~ ← MOV DX, DS:[BP+4]

AX=1234 BX=2000 CX=5678 DX=ABCD
 SP=0540 BP=2000 SI=0000 DI=0000
 DS=0000 ES=0000 SS=0000 CS=0000
 IP=1018 FL=0100 = . . . t

JJ 4 SERIAL MONITOR

8086 >T~~4~~ ← MOV SI, ES:[BP+6]

AX=1234 BX=2000 CX=5678 DX=ABCD
SP=0540 BP=2000 SI=F014 DI=0000
DS=0000 ES=0000 SS=0000 CS=0000
IP=101C FL=0100 = . . . t

8086 >T~~4~~ ← MOV DI, [BP+9]

AX=1234 BX=2000 CX=5678 DX=ABCD
SP=0540 BP=2000 SI=F014 DI=6853
DS=0000 ES=0000 SS=0000 CS=0000
IP=101F FL=0100 = . . . t

8086 >T~~4~~ ← MOV BX, WORD PTR DATA+10

AX=1234 BX=B1C4 CX=5678 DX=ABCD
SP=0540 BP=2000 SI=F014 DI=0000
DS=0000 ES=0000 SS=0000 CS=0000
IP=1023 FL=0100 = . . . t