.....a total solution for

Educational Lab Trainers

8051 MICROCONTROLLER
TRAINER KIT

M51-02

Kitek
Technologies Pvt. Ltd.

B-4,Lotus C.H.S., Near P&MC. Bank, Plot No. 8,
Sector-7, Airoli, Navi Mumbai - 400 708.
Tel.: 022-65116548 / Telefax : 27694323
Email: sales@kitektechnologies.com
Website: www.kitektechnologies.com

M51-02

8051 Microcontroller Trainer

Table of Contents

System Introduction

General Description :

M51-02 is a single board Microcontroller Training/Development Kit configured around the most popular Intel’s 8051, 8 bit single-chip micro controller. M51-02 is about the architecture, instruction set and capabilities of the 8051 chip and also for actually controlling any industrial process.

The Kit communicates with the outside world through a 101 ASCII Keyboard and 20x2 Liquid Crystal Display (LCD). The kit has the capability of interfacing with an IBM-PC compatible PC system through Serial Port.

M51-02 provides 32K bytes of user's RAM and 16K bytes of EPROM. The total on board program & data memory can be very easily expanded to 64K bytes in an appropriate combination of RAM.

The Input/Output structure of M51-02 provides 48 programmable I/O lines using two nos. of 8255. It has got 16 bit programmable Timer/Counter for generating any type of counting using 8253.

The Kit has RS-232C interface using Rx & Tx of 8051. It also has Optional AUX Serial Port using 8251 for any serial communication.

This Kit has Optional 12 Bit ADC using AD574, 8 bit DAC using DAC0800, Real Time Clock & Printer Interface.

The on board resident system monitor software is very powerful and provides various software utilities. The kit provides various powerful software commands like Block Move, Fill, Set./CLR Breakpoints, Single Stepping, Modify Memory/Register, Line Assembler/Dissembler etc. which are very helpful in debugging/developing the software.

The system also provides a serial monitor covering most of the commands available in keyboard mode. A help menu makes the monitor more user friendly.

M51-02 is configured around the internationally adopted Bus, which is the most popular bus for process control and real time applications. All the address, data and control lines are available at 40 pin & 10 pin FRC connector. The Kit is fully expandable for any kind of application.

System Specifications

CPU	**:**	8 bit Microcontroller using 8051/89C51/89C52 operating @ 11.0592 MHz (Kit configured as 89C52).
Monitor EPROM	**:**	16K bytes of EPROM loaded with powerful monitor program.
RAM	**:**	32K bytes of user's RAM using 62256 expandable upto 64K.
Battery Backup	**:**	3.6V Ni-Cd battery for user RAM.
Timer	**:**	16 bit programmable timer/counter using 8253 & External Timer T0, T1 from 8051.
I/O	**:**	48 I/O lines using 8255 PPI chips & 14 I/O lines provided by 8051.
Interrupts	**:**	Two external interrupts INT0 & INT1 from 8051.
Keyboard	**:**	101 ASCII Keyboard.
Display	**:**	20x2 Backlite LCD Display.
Optional Interface	**:**	1 Channel 12bit ADC using AD574
	:	1 Channel 8bit DAC using DAC0800
	:	Printer Interface
	:	Real Time Clock (RTC) using 6242
Serial Interface	**:**	1) One serial UART interface provided by 8051.
		2) One AUX RS-232C through 8251 with a programmable baud rate(Optional)
BUS	**:**	All data, address and control signals (TTL compatible) available at FRC connector
Power Supply	**:**	In-Built +5V/1A & ±12V/250mA.
Physical Size	**:**	24cm x 15.5cm.
Operating Temp.	**:**	0 to 50°C.

System Capabilities

101 ASCII Keyboard Mode :

1. Examine the contents of any memory location.

2. Modify the contents of any of the RAM location.

3. Move a block of data memory to program memory.

4. Examine/Modify the contents of Register

5. Move a block of program memory to program memory.

6. Fill a particular memory area with a constant.

7. Transmit an Intel Hex file from memory to an IBM PC Compatible computer.

8. Receive the Intel Hex file into memory from an IBM PC Compatible computer.

9. Execute a program at full clock speed.

10. Execute a program in single step i.e. instruction by instruction.

11. Set a Break Point.

12. Clear a Break Point.

13. Enable a Break Point.

14. Disable a Break Point.

15. Display a Break Point.

Note: All the above commands can be operated through Serial mode provided.

System Installation

Keyboard Mode :

To install M51-02 in keyboard mode, the following additional things are required.

1. Connect the AC power supply through AC power chord provided to the M51-02 kit.

2. Switch on the power supply at the rear end.

3. The below message will come on LCD screen:

8051 TRAINER KIT

After some time the operating commands will be displayed as below:

COMMAND : A, B, C, D, E, F,
G, I, L, M, N, P, R, S, T, Y

4. Now M51-02 Kit is ready for the user's experiments for keyboard mode commands.

Serial Mode :

To install M51-02 in serial mode, the following additional things are required.

1. Steps 1, 2 and 3 of Keyboard Mode will remain same for Serial Mode.

2. Connect RS-232C Cable from PC Serial Port (Com-1 OR Com-2) to the connector CN8 of the M51-02 Kit.

3. Switch ON the Hyper Terminal of Win95/98/2000/NT/XP and set the Baudrate at 9600bps.

4. Press **`Y'** Key on ASCII Keyboard of M51-02 Kit. The below message will display on LCD Screen. Now M51-02 Kit is ready for the user's experiments for Serial Mode commands.

SERIAL PC COMUNICATN

Hardware Description

CPU :

The system has got 8051 as the Micro-controller operating at a crystal frequency of 11.0592 MHz. The 8051 Micro controller is an 8 bit CPU with on chip oscillator and clock circuitry, 8 input/output lines, Two 16 bit timer/counters, a fire source interrupt structure, full duplex serial port and built in Boolean processor. The chip has the capacity to address 64K bytes of external data memory and 64K bytes of external program memory.

Memory :

M51-02 provides 32K bytes of CMOS RAM using 62256 chip and 16K bytes of Powerful Monitor EPROM. The total on board program/data memory can be expanded up to 64K bytes.

I/O Devices :

The various I/O chips used in M51-02 are 8255, 8253, 8251, AD574, DAC0800 & RTC6242. The functional role of all these chips are given below:

8255 :

8255 is a programmable peripheral interface (PPI) designed to use with Microprocessor/microcontrollers. This basically acts as a general purpose I/O device to interface peripheral equipments to the system bus. It is not necessary to have an external logic to interface with peripheral devices since the functional configuration of 8255 is programmed by the system software. It has got three Input/Output ports of 8 lines each (Port-A, Port-B & Port-C). Port-C can be divided into two ports of 4 lines each named as Port-C upper and Port-C lower. Any Input/Output combination of Port-A, Port-B, Port-C upper and lower can be defined using the appropriate software commands. M51-02 provides 48 Input/Output lines using two 8255 chips. All I/O lines are brought out at 26pin FRC Connector CN3 & CN4.

8253 :

This chip is a programmable interval Timer/Counter and can be used for the generation of accurate time delays under software control. Various other functions that can be implemented with this chip are Programmable rate generator, Even counter, Binary rate multiplier, Real time clock etc. This chip has got three independent 16 bit counters each having a count rate of up to 2MHz. The first Timer/Counter (i.e. Counter 0) is being used for baud rate generation for RS-232C interface. The other timer/counters are available to the user at 10pin FRC connector CN5.

8251 :

This chip is a programmable communication interface and is used as a peripheral device. This device accepts data characters from the CPU in parallel format and then converts them into serial data characters for the CPU. This chip will signal the CPU whenever it can accept a new character for the CPU. The CPU can read the complete status of it at any time. 8251 has been utilized in M51-01 for AUX RS-232C interface are available 9 pin D-type Connector CN9.

Optional Interface :

M51-02 provides Additional optional interfaces of Single Channel 12 bt ADC using ADC547, Single Channel 8 bit DAC using DAC0800, Real Time Clock using RTC 6242. Using this interface one can configure for any Process Control Applications.

Battery Backup :

The M51-02 provides a battery back up for the onboard RAM area using 3.6V Ni-Cd Rechargeable battery. M51-02 has facility for connecting +5V to the RAM area if the Ni-Cd battery fails. The selection for +5V or Battery supply are available at Jumper (JP1).

Display :

This display contains 2 lines and each line consists of 20 words (20x2). This is a cursor LCD display modular. The CPU receives each 8 bits letter which is locked into the internal display data of RAM (data display of RAM 80 bytes (D.D.RAM) allows 80 characters to be stored), and transfer to 5x7 dot of array word and appear on the displayed.

This LCD modular contains the word generator ROM that will supply 160 different 5x7 dot of array word and also a 64 bytes word generator RAM. Users can define 8 types 5x7 dot of array word.

The position of word display goes into the LCD Modular through the data bus in CPU. Next through the instruction register and finally write the words into the data register to display on a specific location. The LCD Modular will automatically increase or decrease the words in order to move to different addresses. The user can therefore continue sending in word code. The cursor as to moved around or moved in the right of left direction.

Specification of Display :

1. Display data RAM : 80x8 BLT (80 words)

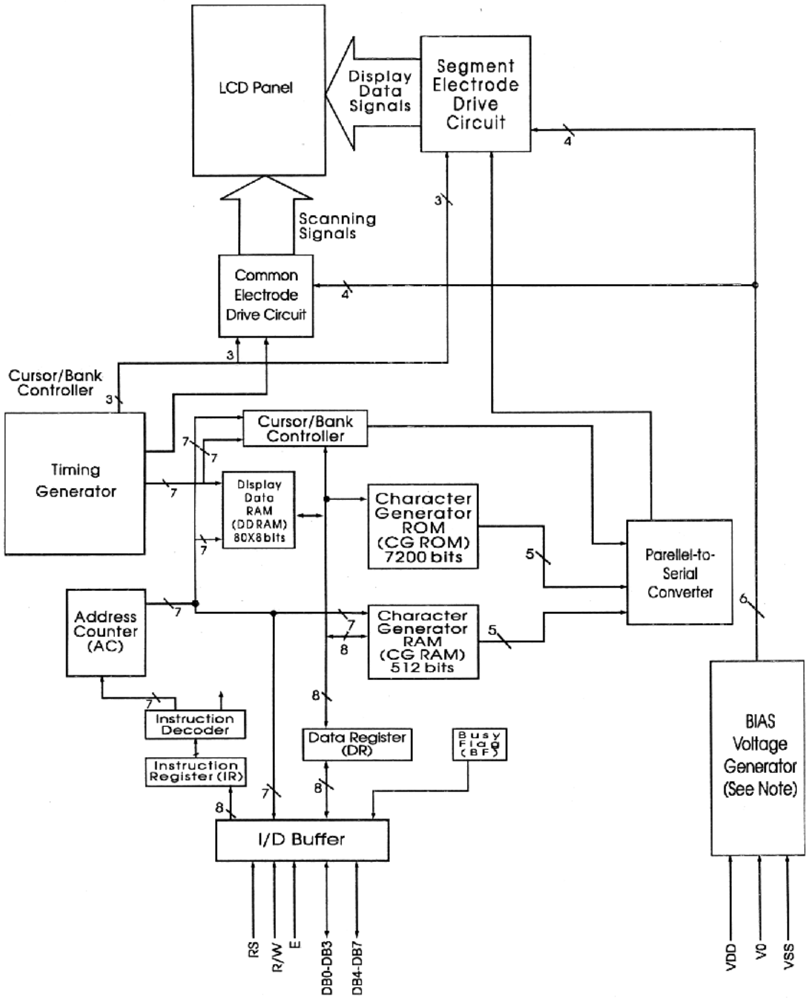
2. Character generator ROM : 160 of 5x7 dot of array word

3. Character generator RAM : 8 different users programmed 5x7 dot of array

4. Kinds of instructions : Clear the display, send cursor home (HOME), ON/OFF display. Cursor ON/OFF, character blinking cursor move to another position, display change position.

5. When the internal power is on, the circuit is reset.

6. Internal circuit vibrator.


LCD Panel
Display Data Signals
Segment Electrode Drive Circuit
4
3
Scanning Signals
Common Electrode Drive Circuit
4
3
Cursor/Bank Controller
3
Timing Generator
7
7
7
Cursor/Bank Controller
Display Data RAM (DDRAM) 80X8bits
7
Character Generator ROM (CG ROM) 7200 bits
5
Parellel-to-Serial Converter
6
Address Counter (AC)
7
Character Generator RAM (CG RAM) 512 bits
7
8
5
7
Instruction Decoder
Instruction Register (IR)
8
Data Register (DR)
Busy Flag (BF)
8
7
8
I/D Buffer
BIAS Voltage Generator (See Note)
RS
R/W
E
DB0-DB3
DB4-DB7
VDD
V0
VSS

Functional Block Diagram

Fig. 1

Note :

Some models incorporate a temperature compensation circuit within the bias voltage generator

The LCD modular has 2 8-bits register-one instruction register (IR) and one data register (DR).

The instruction register stores the instruction code and address information, which contains display data RAM and address of character generator RAM. However, the content of IR is only for read-in but not read-out.

The data register can only temporary store data, the input data first goes through LCD and is stored in the data register. It will then automatically be transferred to display data RAM or character generator RAM. When the CPU read the data from the displayed RAM or from the character generator RAM, it will also temporary store the data in the data register. When the address information is input into the instruction register, the relative data will be moved from display register RAM or character generator RAM to the data register. Then the data can be read from data register by using the output instruction of CPU.

One way to select the two registers is to select the register signal (RS) like follow:

RS	R/W	Function
0	0	Data Bus —> instruction Register
0	1	Read out busy flags (BUSY FLAG DB7) and address counter (DB0-DB6)
1	0	Input into data register and execute the inner instruction: (D.R.RAM—> D.R. OR C.G.RAM — D.R.)
1	1	Get the data out form register, and execute the inner instruction: (D.D.RAM—> D.R. OR C.G.RAM—> D.R.)

Busy Flag (B.F.)

When busy flag is “1”, it indicates that the LCD Modular is executing the inner instruction and no other instruction can be accepted. The LCD Modular can only accept information when BF is lower to “0”.

Address Counter (A.C.)

The address counter is used to count the display data RAM, or address of character generator RAM. When the address setting instruction address will be sent into the address counter.

When the data is sent into or read out from display register RAM or from the character generator RAM, the address counter will automatically add or subtract 1.

When the content of address counter is in RS = 0 and R/W = 1, the output data line is DB0 DB6.

Display Data RAM (D.D. RAM)

This is a 80x8 bit RAM, which can store 80 8-bit character code as the display data, it can be sent to CPU as the RAM data section without going through RAM section.

Address setting of data display RAM is as followed:

High level bus	Low level bus
AC6 AC5 AC4	AC3 AC2 AC1 AC0

Data display RAM and display position of LCD is as followed:

Character Position (decimal)	:	1	2	3	4	5	6	7	8	9	10	11	......	19	20
First Line (hexadecimal)	:	**00**	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**0A**	**......**	**16**	**17**
Second Line (hexadecimal)	:	**40**	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**4A**	**......**	**56**	**57**

Character Generator ROM (C.G. ROM)

This ROM generates 5x7 dot of array character has 160 different 8-bit character code. The shape and code are shown in Table 2 and 3.

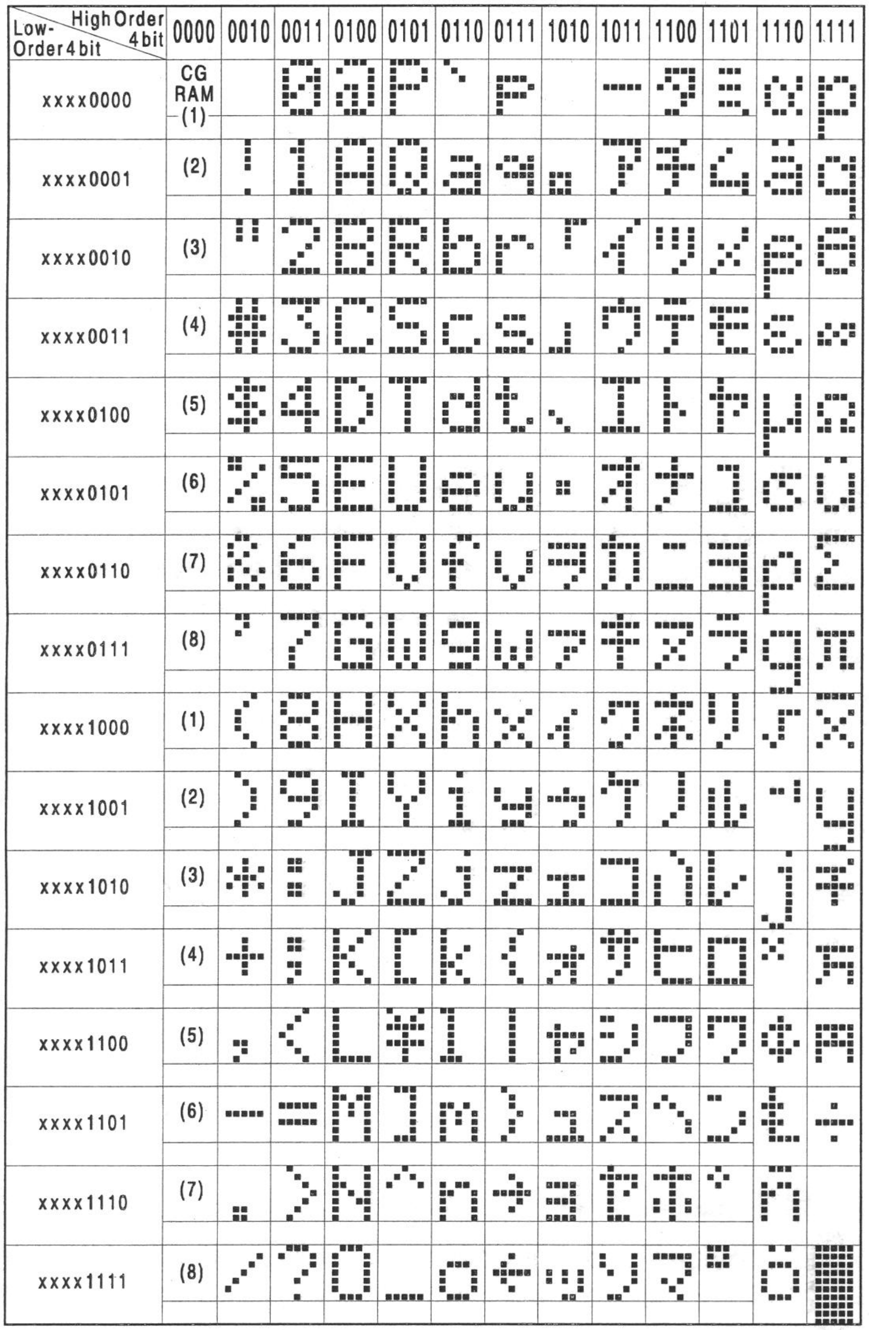
Character Generator RAM (C.G.RAM)

This RAM stores 8 different 5x7 dot of array character which allows the user to design the program. When the character codes is stored in the C.G.RAM, which are the same as the characters in Table 2 and 3, they will be sent to display data RAM. The display data and characters are shown in table.

Timing Generator

Sending signals into the inner register during generating process.

Character codes :



Low-Order 4 bit \ High Order 4 bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	@	P	`	p		ー	タ	ミ	α	p
xxxx0001	(2)	!	1	A	Q	a	q	。	ア	チ	ム	ä	q
xxxx0010	(3)	"	2	B	R	b	r	「	イ	ツ	メ	β	θ
xxxx0011	(4)	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
xxxx0100	(5)	$	4	D	T	d	t	、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)	%	5	E	U	e	u	・	オ	ナ	ユ	σ	ü
xxxx0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	'	7	G	W	g	w	ァ	キ	ヌ	ラ	g	π
xxxx1000	(1)	(	8	H	X	h	x	ィ	ク	ネ	リ	√	x̄
xxxx1001	(2)	)	9	I	Y	i	y	ゥ	ケ	ノ	ル	⁻¹	y
xxxx1010	(3)	*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
xxxx1011	(4)	+	;	K	[	k	{	ォ	サ	ヒ	ロ	ˣ	万
xxxx1100	(5)	,	<	L	¥	l	|	ャ	シ	フ	ワ	¢	円
xxxx1101	(6)	-	=	M	]	m	}	ュ	ス	ヘ	ン	£	÷
xxxx1110	(7)	.	>	N	^	n	→	ョ	セ	ホ	゛	ñ	
xxxx1111	(8)	/	?	O	_	o	←	ッ	ソ	マ	゜	ö	█

Note :

1. The CG RAM generates character patterns in accordance with the user's program.
2. Shaded areas indicate 5x10 dot character patterns.

Character code :

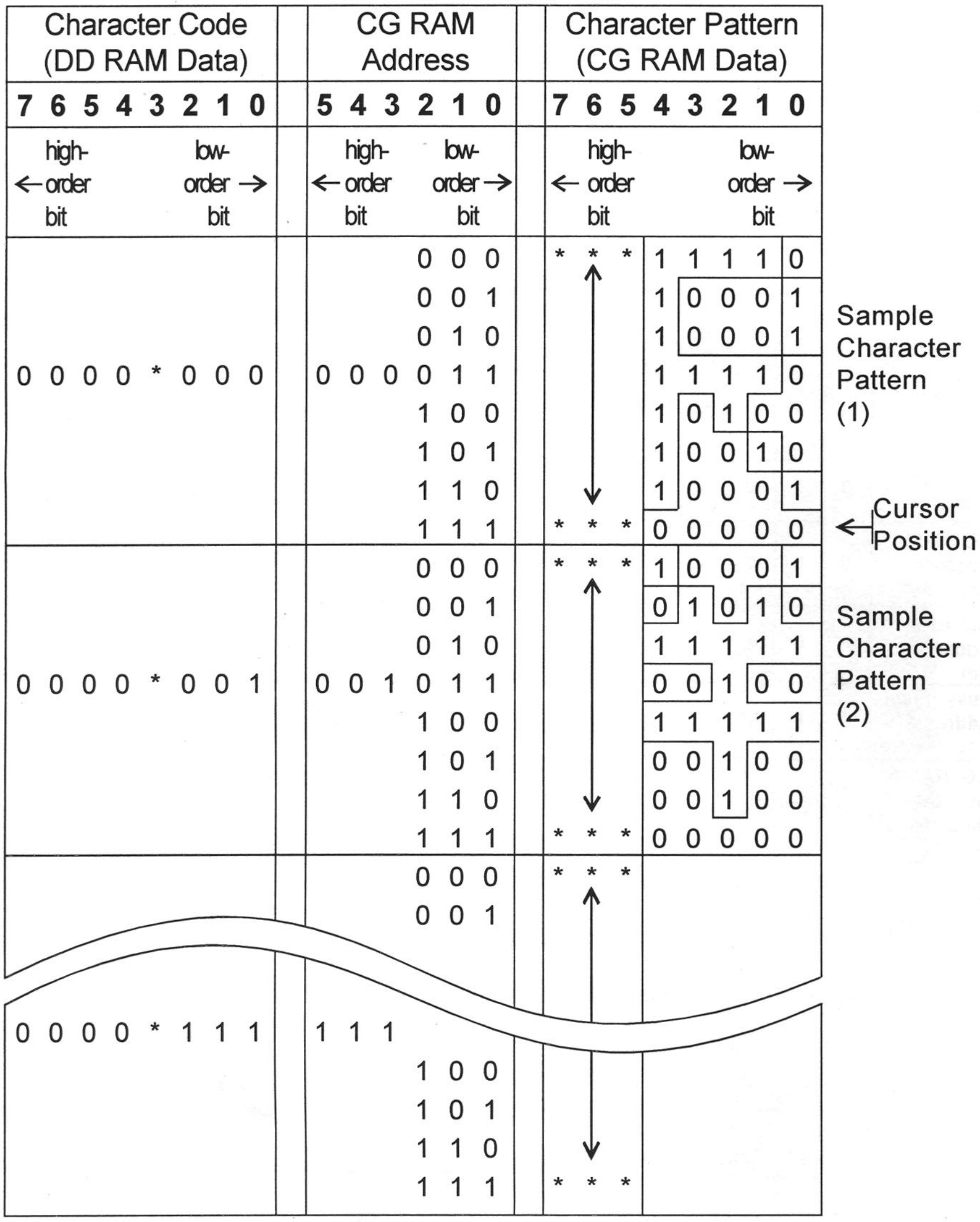
Low-Order 4 bit \ High Order 4 bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	@	P	\	p		-	タ	ミ	α	p
xxxx0001	(2)	!	1	A	Q	a	q	。	ア	チ	ム	ä	q
xxxx0010	(3)	"	2	B	R	b	r	「	イ	ツ	メ	β	θ
xxxx0011	(4)	#	3	C	S	c	s	」	ウ	テ	モ	∈	∞
xxxx0100	(5)	$	4	D	T	d	t	`	エ	ト	ヤ	μ	Ω
xxxx0101	(6)	%	5	E	U	e	u	•	オ	ナ	ユ	σ	ü
xxxx0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	’	7	G	W	g	w	ァ	キ	ヌ	ラ	ϑ	π
xxxx1000	(1)	(	8	H	X	h	x	ィ	ク	ネ	リ	√‾	x̄
xxxx1001	(2)	)	9	I	Y	i	y	ゥ	ケ	ノ	ル	−1	y
xxxx1010	(3)	*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
xxxx1011	(4)	+	;	K	[	k	{	ォ	サ	ヒ	ロ	x	万
xxxx1100	(5)	,	<	L	γ	l	|	ャ	シ	フ	ワ	¢	円
xxxx1101	(6)	_	=	M	]	m	}	ュ	ス	ヘ	ン	£	÷
xxxx1110	(7)	.	>	N	^	n	→	ョ	セ	ホ	゛	n̄	
xxxx1111	(8)	/	?	O	_	o	←	ッ	ソ	マ	゜	ö	

Note :

1. The CG RAM generates character patterns in accordance with the user’s program.
2. Shaded areas indicate 5x10 dot character patterns.

Relationship among character code :

(DD RAM), CG RAM Address, and Character Pattern (CG RAM) Character Pattern for 5x7 Foat



Character Code (DD RAM Data)	CG RAM Address		Character Pattern (CG RAM Data)		
7 6 5 4 3 2 1 0	5 4 3	2 1 0	7 6 5	4 3 2 1 0	
← high-order bit / low-order bit →	← high-order bit	low-order bit →	← high-order bit	low-order bit →	
		0 0 0	* * *	1 1 1 1 0	
		0 0 1		1 0 0 0 1	Sample Character Pattern (1)
		0 1 0		1 0 0 0 1	
0 0 0 0 * 0 0 0	0 0 0	0 1 1		1 1 1 1 0	
		1 0 0		1 0 1 0 0	
		1 0 1		1 0 0 1 0	
		1 1 0		1 0 0 0 1	
		1 1 1	* * *	0 0 0 0 0	← Cursor Position
		0 0 0	* * *	1 0 0 0 1	
		0 0 1		0 1 0 1 0	Sample Character Pattern (2)
		0 1 0		1 1 1 1 1	
0 0 0 0 * 0 0 1	0 0 1	0 1 1		0 0 1 0 0	
		1 0 0		1 1 1 1 1	
		1 0 1		0 0 1 0 0	
		1 1 0		0 0 1 0 0	
		1 1 1	* * *	0 0 0 0 0	
		0 0 0	* * *		
		0 0 1			
0 0 0 0 * 1 1 1	1 1 1				
		1 0 0			
		1 0 1			
		1 1 0			
		1 1 1	* * *		

* Signifies a “don’t care” bit.

Notes :

1. Character code bits 0-2 correspond to CG RAM address bits 3-5. Each of the 8 unique bit strings designated one of the 8 character patterns.

2. CG RAM address bits 0-2 designates the row position of each character pattern. The 8th row is the cursor position. CG RAM data in the 8th row is OR'ed with the display cursor. Any "1" bits in the 8th row will result in the displayed dot regardless of the cursor status (ON/OFF). Accordingly, if the cursor is to be used, CG RAM data for the 8th row should be set to "0".

3. CG RAM data bits 0-4 correspond to the column position of each character pattern bit 4 corresponding to the leftmost column of the character pattern CG RAM data bus are not used for displaying character patterns, but may be used as a general.

4. As shown in tables 2 and 3, character patterns in the CG RAM are accessed by character codes with bits 4-7 equal to "0". For example, the character code "00" (HEX) or "80" (HEX), since bit 3 of the character code is a don't care bit (i.e. can take either value "0" or "1").

5. CG RAM data "1" produces a dark dot, and data "0" produces a light dot in the corresponding position on the display panel.

Functions of Reset

Using the Internal Reset Circuit To Start

LCD Modular internal has an automatic power supply to be used to RESET when the power rises. During RESET, the busy flag is set. When the voltage is raised to 4.5V in about 10ms, it is in the busy stage. The following instructions are then used to set the beginning stage of LCD.

1. Clear display

2. Function set

DL	=	1	8-bit data length interface
N	=	0	(single line display)
F	=	0	The source of 5x7 dot of array character

3. Display ON/OFF control

D	=	1	Display OFF
C	=	0	Cursor OFF
B	=	0	Character flashing function OFF

4. Entry mode set

I/D	=	1	Increase mode
S	=	0	Display OFF

Note : If the time for the power to increases from 0.2V to 4.5V is greater than 0.1ms but less than 10ms, the current cut-off will drop to 0.2V before it rises again. If it takes more than 1ms, the LCD modular will automatically RESET. Otherwise, it has to depend on an external software instruction to RESET (As describe below).

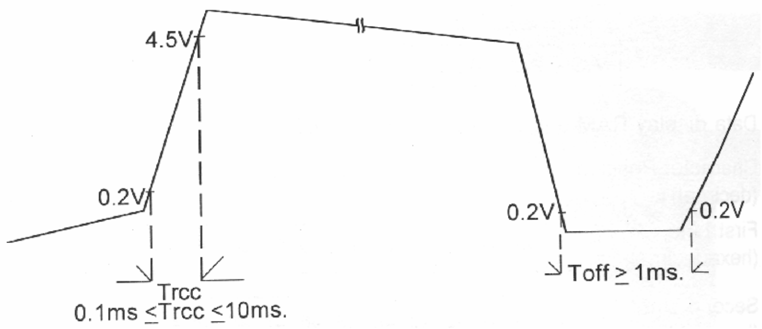
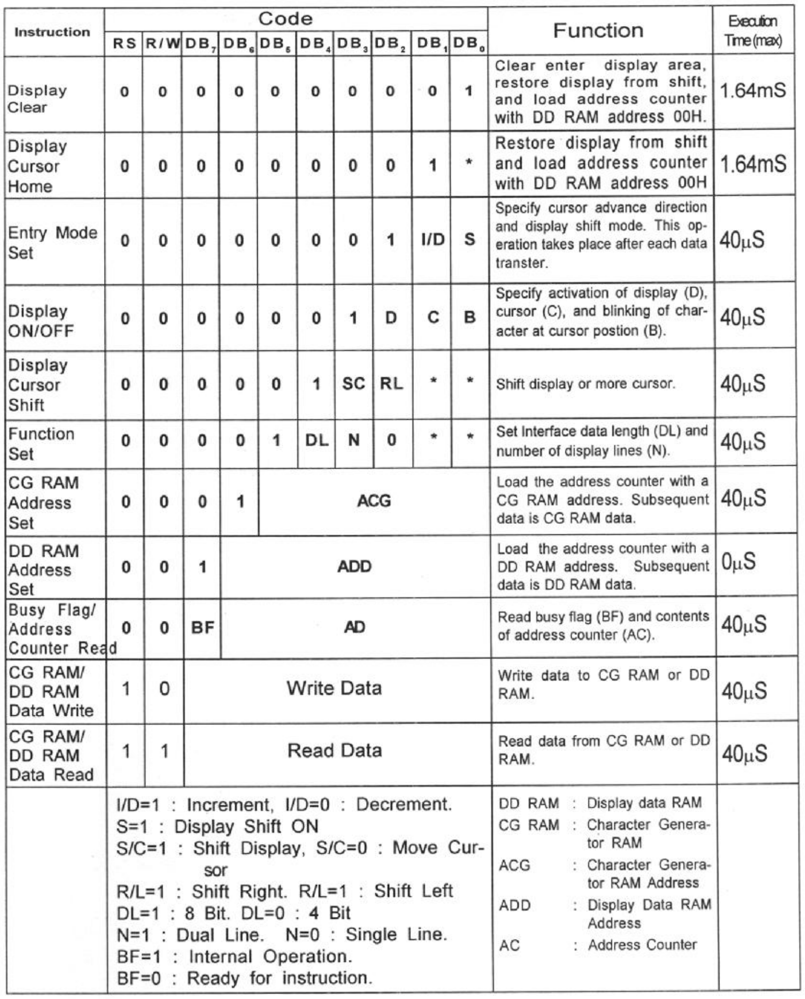


Diagram of module RESET power

Fig. 2

Instruction set :



Instruction	Code										Function	Execution Time (max)
	RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0		
Display Clear	0	0	0	0	0	0	0	0	0	1	Clear enter display area, restore display from shift, and load address counter with DD RAM address 00H.	1.64mS
Display Cursor Home	0	0	0	0	0	0	0	0	1	*	Restore display from shift and load address counter with DD RAM address 00H	1.64mS
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Specify cursor advance direction and display shift mode. This operation takes place after each data transter.	40μS
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	Specify activation of display (D), cursor (C), and blinking of character at cursor postion (B).	40μS
Display Cursor Shift	0	0	0	0	0	1	SC	RL	*	*	Shift display or more cursor.	40μS
Function Set	0	0	0	0	1	DL	N	0	*	*	Set Interface data length (DL) and number of display lines (N).	40μS
CG RAM Address Set	0	0	0	1	ACG						Load the address counter with a CG RAM address. Subsequent data is CG RAM data.	40μS
DD RAM Address Set	0	0	1	ADD							Load the address counter with a DD RAM address. Subsequent data is DD RAM data.	0μS
Busy Flag/ Address Counter Read	0	0	BF	AD							Read busy flag (BF) and contents of address counter (AC).	40μS
CG RAM/ DD RAM Data Write	1	0	Write Data								Write data to CG RAM or DD RAM.	40μS
CG RAM/ DD RAM Data Read	1	1	Read Data								Read data from CG RAM or DD RAM.	40μS
	I/D=1 : Increment, I/D=0 : Decrement. S=1 : Display Shift ON S/C=1 : Shift Display, S/C=0 : Move Cursor R/L=1 : Shift Right. R/L=1 : Shift Left DL=1 : 8 Bit. DL=0 : 4 Bit N=1 : Dual Line. N=0 : Single Line. BF=1 : Internal Operation. BF=0 : Ready for instruction.										DD RAM : Display data RAM CG RAM : Character Generator RAM ACG : Character Generator RAM Address ADD : Display Data RAM Address AC : Address Counter	

Note :

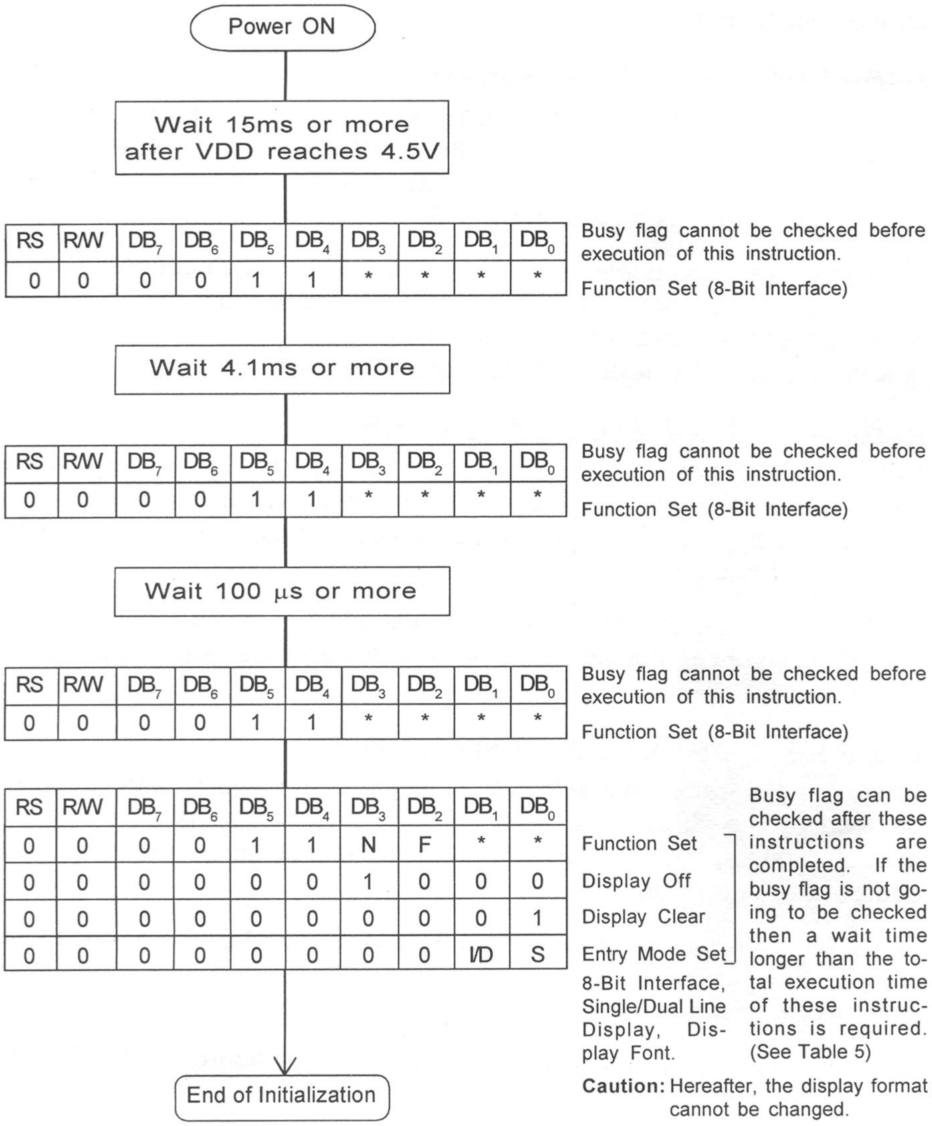
1. Symbol “*” signifies a “don’t care” bit

2. Correct input value for “N” is predetermined for each model

Initialization by Instructions :

If the power conditions for the normal operation of the internal reset circuit are not satisfied.

LCD unit must be initialized by executing a sense of the instructions. The procedure fro this initialization process is as follows.



Power ON

Wait 15ms or more after VDD reaches 4.5V

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0
0	0	0	0	1	1	*	*	*	*

Busy flag cannot be checked before execution of this instruction.

Function Set (8-Bit Interface)

Wait 4.1ms or more

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0
0	0	0	0	1	1	*	*	*	*

Busy flag cannot be checked before execution of this instruction.

Function Set (8-Bit Interface)

Wait 100 μs or more

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0
0	0	0	0	1	1	*	*	*	*

Busy flag cannot be checked before execution of this instruction.

Function Set (8-Bit Interface)

RS	R/W	DB_7	DB_6	DB_5	DB_4	DB_3	DB_2	DB_1	DB_0	
0	0	0	0	1	1	N	F	*	*	Function Set
0	0	0	0	0	0	1	0	0	0	Display Off
0	0	0	0	0	0	0	0	0	1	Display Clear
0	0	0	0	0	0	0	0	I/D	S	Entry Mode Set

8-Bit Interface, Single/Dual Line Display, Display Font.

Busy flag can be checked after these instructions are completed. If the busy flag is not going to be checked then a wait time longer than the total execution time of these instructions is required. (See Table 5)

End of Initialization

Caution: Hereafter, the display format cannot be changed.

Instruction Description

When the LCD is controlled by the CPU, only the instruction register (IR) and the data register (DR) can be read directly by the CPU. The commands from outside the modular can decide the internal operation of LCD. These commands include the register selection (RS) signals, read/write (R/W) signals, and data buffering signals (DB0-DB6).

Table 5 lists all the useful commands in the LCD modular and the execution time, these commands are divided into the following group:

1. Commands of set LCD module

2. Commands of internal set address RAM

3. Commands of data transfer in or out from the internal RAM

4. Other commands

When the LCD modular is executing a command it will reject other commands. Except the "busy flag/read address counter, the internal counting period of busy flag is set to as "1". If the CPU wants to send in other commands it will have to check the busy flag first, until it is cleared to "0" before it send in. The explanation is as followed:

Display Clear Command

This command will put the display data into a empty space" code (20H), address counter will be cleared to 0. When executing this command, display OFF, the cursor or the character blinking function will be moved to the most left side if it is in the set condition.

Display/Cursor Home

The address counter will be cleared to 0, content of D.D. RAM will not be influenced; but if the cursor or the character blinking function is in the set condition, it will be moved to the most left side position.

Entry Mode Set

I/D bit = "1", "1" is added in the address counter after each time it read/write a display data RAM character code, so that the cursor or the character blinking function will move one place to the left and vise-versa when I/D=0. The read/write (R/W) character generator also has the same function.

S bit = 1, but each time it read/write a display data RAM code, it will move to the display direction and move one space to the left (I/D=0) or one space

to the right (I/D=1). When S=0, the display will not move.

When data enters the character generator RAM, the display will not move.

Display On/Off

D: D=1 - Display ON

D=0 - Display OFF

C: C=1 - Cursor display on the display address of the display counter

C=0 - Cursor does not display

B: B=1 - Character blinking of cursor position at feq or fosc=250KHz freq, therefore all black points and character display will exchange with each other. Each character display and overshadow 409.6ms.

Display/Cursor Shift

S/C	R/L		
0	0	-	Cursor move to the left (AC <—AC-1)
0	0	-	Cursor move to the right (AC <—AC+1)
1	0	-	All the characters and cursor move to the left
1	1	-	All the characters and cursor move to the right

Note : When the display moves, the address counter will not move.

Function Set

DL : Select data length for the interface circuit.

DL=1 - Using the 8 bits data length.

DL=0 - Using the 4 bits data length.

N : Select the display format (one or two lines)

C.G. RAM Address Set

Address counter and character generator RAM have address which are driven by the binary 6-bit. when this instruction is driven in, data can be sent into the CPU and character generator RAM.

D.D. RAM Address Set

Address counter and display data RAM have addresses which are driven by the binary 7-bit. When this instruction is driven in, data can be sent into the CPU and the display data RAM. When N=0 (a single line display), binary code ADD between 00H and 4FH; when N=1 (a two lines display), the binary code ADD from 00H until 27H as the first line of from 40H until 67H as the second line.

Read Busy Flag/Address Counter

The busy flag (BF) in LCD can be read from the CPU, using the instruction of LCD modular is the execution of the internal instruction BF = 1 represents the busy stage (execution of the internal instruction), it will not accept any instruction at this time until BF = 0.

Content of address counter and the busy flag will be read out at the same time, it is a 7-bit binary, the address counter will instruct one of the address, either the character generator RAM or display data RAM. This is determined by the final input address set instruction.

Send Data Into C.G. RAM/D.D. RAM

Data with 8-bit in length can be sent into the character generator RAM or the display data RAM. The address of the input data is instructed by the address counter, however, the address of address counter is influenced by the final input address set instruction.

After data input whether the address counter add 1 or minus 1 is determined by the design of the module. It can also be designed as location movement of the display.

Read Data Out of C.G. RAM/D.D. RAM

Character generator RAM with 8-bit in length or the display data RAM can be read by the CPU. The read out data address is instructed by the address counter. The address counter is instructed by the final input address set instruction.

This instruction has to be set in C.G. RAM/D.D. RAM address, once shift cursor instruction of the C.G. RAM/D.D. RAM data is read out, no other instruction can be read out.

The address setting instruction will read the data address into address counter.

Shift cursor command will allow the previous address setting to be used again in order to read the D.D. RAM data. the data can be read from the C.G. RAM/D.D. RAM after the cursor shift.

After the execution of data address counter add 1 or minus 1 will be set in the LCD modular.

After the execution of data read out, the display will not shift.

Keyboard Commands

The operation of this device is simpler by using 101 ASCII Keyboard and 20x2 LCD Display. For convenience, the operation instructions will be displayed when the device is being switched on or RESET.

After power ON the system, it will display as follows:

```
  8051 TRAINER KIT

```

After some time, the operating commands will be displayed:

```
COMMAND : A, B, C, D, E, F
  G, I, L, M, N, P, R, S, T, Y
```

Command Details

A	- Line Assembler	:	for writing program in mnemonics.
B	- Block Move	:	allow user to Move a Block of memory to any RAM area.
C	- Clear Break Point	:	to Clear a set Break Point address.
D	- Dis-assembler	:	to Disassemble the RAM Contents.
E	- Enable Breakpoint	:	to Enable a set Break Point address.
F	- Block Fill	:	to Fill RAM area with a constant data.
G	- Execution	:	to Execute Program in full clock speed.

I	-	View Breakpoint	:	to View a set Break Point address.
L	-	Disable Breakpoint	:	to Disable a set Break Point address.
M	-	Examine/Modify Memory	:	to Examine/Modify any RAM location.
N	-	Single Stepping	:	to execute the program in Single Step Mode.
R	-	Examine/Modify Register	:	to Examine/Modify internal register.
T	-	Set Breakpoint	:	to set a Breakpoint at any RAM location.
Y	-	Serial Communication	:	to go in Serial Mode for PC operation.

Command Description

The following are some useful keys used to move the cursor around:

Back Space ....Wrong Data or Character can be Deleted.

Shift ....To access the upper Character of any Key.

Space Bar ....To give space between two Characters.

A - Line Assembler

This command is used to convert the input Assemble Language to the Machine Language in the memory.

COMMAND : A, B, C, D, E, F
G, I, L, M, N, P, R, S, T, Y

Simply Press the key ‘A’. After the command, the LCD screen will be blank.

Now type the below syntax in the LCD screen.

Syntax :

ASM ORG 3000H

ASM ORG 3000H

Now press `ENTER' key on the ASCII Keyboard. The LCD screen will change as follows.

3000 : _

Now one can write any Assemble instructions in this kit.

Example :

Addition of two bytes.

3000 MOV A,#02H

3002 MOV R0,#03H

3004 ADD A,R0

3005 SJMP 3005H

Now enter the above example in the kit. Pick the first instruction in the LCD Kit using ASCII Keyboard.

```
3000 : MOV A,#02H
```

As user press the `**ENTER**' key the fist instruction is accepted by kit and has converted into the corresponding HEX code. Now Kit LCD screen will change as follows.

```
3002 : _
```

If the instruction is wrong the kit will not accepted the instruction and the address will remain same as follows.

```
3000 : _
```

In this way user can enter whole program as below using `**ENTER**' key.

```
3002 : MOV R0,#03H
```

```
3004 : ADD A,R0
```

```
3005 : SJMO 3005H
```

D - Dissembler

The Disassemble command decodes the value of group memory location mnemonics and display on the LCD screen.

```
COMMAND : A, B, C, D, E, F
 G, I, L, M, N, P, R, S, T, Y
```

Simply Press the key '**D**'. After the command, the LCD screen will be blank.

```

```

Now type the below syntax in the LCD screen.

Syntax :

<RAM Starting Address> <Space> TO <Space> <RAM Ending Address>

```
3000 TO 3006
```

Now press `**ENTER**' key on the ASCII Keyboard. The LCD screen will change as follows.

```
3000 = 74 02
MOV A,#02H
```

By pressing further `**ENTER**' Key the address will be automatically incremented with mnemonics and corresponding code. To terminated this command press `**ESC**' Key.

M - Examine/Modify Memory

The Examine/Modify memory command use to examine OR Modify any memory contents in the Kit.

```
COMMAND : A, B, C, D, E, F
  G, I, L, M, N, P, R, S, T, Y
```

Simply Press the key '**M**'. After the command, the LCD screen will be displayed as below.

```
ENTER ADDRESS
```

Now type the below syntax in the LCD screen.

Syntax :

<RAM/ROM Address> <Space Bar>

```
ENTER ADDRESS
3000 -
```

Now press `**SPACE BAR**' key on the ASCII Keyboard to Examine the RAM/ROM location.

```
ENTER ADDRESS
3000 - 74
```

Now one want to examine the RAM/ROM location simply press `**SPACE BAR**' Key.

To Modify the RAM location with a new data, write the new data in the same RAM location as follows.

Example :

Change Data 75 in the place of old data 74.

Type data 75 with the help of ASCII Keyboard on the LCD screen.

ENTER ADDRESS

3000 - 75

Now press `**SPACE BAR**' key on the ASCII Keyboard to Modify the RAM location. Now LCD screen display as follows.

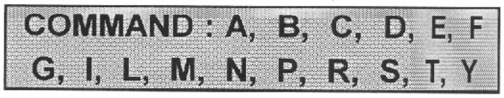
ENTER ADDRESS

3001 - 02

To terminate this command press `**ENTER**' Key.

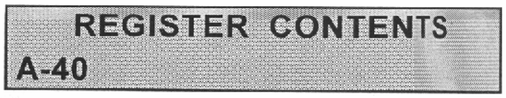
R - Examine/Modify Register

The Examine/Modify Register command use to examine OR Modify any Register contents of 8051 in the Kit.



COMMAND : A, B, C, D, E, F
G, I, L, M, N, P, R, S, T, Y

Simply Press the key ‘**R**’. After the command, the LCD screen will be display as below.



REGISTER CONTENTS
A-40

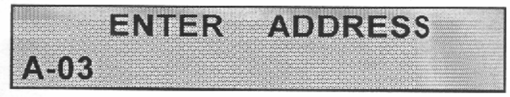
Now press <space Bar> Key to display further register.

To Modify the Register Contents with a new data, write the new data in the same Register as follows.

Example :

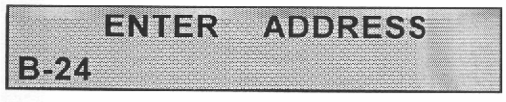
Change the Accumulator Data with 03 in the place of old data 40.

Type data 03 with the help of ASCII Keyboard on the LCD screen.



ENTER ADDRESS
A-03

Now press `**SPACE BAR**' key on the ASCII Keyboard to Modify the Register. Now LCD screen display as follows.

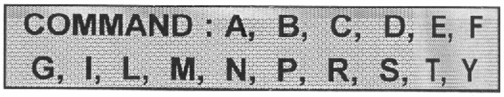


ENTER ADDRESS
B-24

To terminate this command press `**ENTER**' Key.

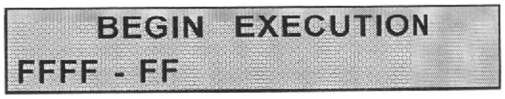
G - Execution

The Execution command use to execute the entered program in full clock speed.



```
COMMAND : A, B, C, D, E, F
 G, I, L, M, N, P, R, S, T, Y
```

Simply Press the key ‘**G**’. After the command, the LCD screen will be display as below.

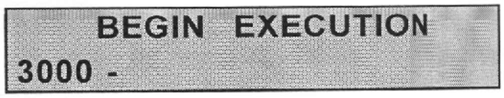


```
      BEGIN  EXECUTION
FFFF - FF
```

Syntax :

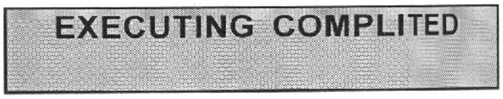
<RAM Starting Address> <ENTER>

If the Program starting address is 3000 location then type RAM starting address 3000.



```
      BEGIN  EXECUTION
3000 -
```

Now press `**ENTER**' key on the ASCII Keyboard to Execute the program from 3000 location. Now the LCD screen will display as below.

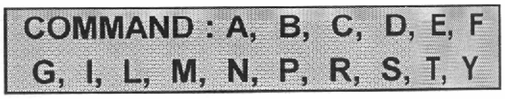


```
   EXECUTING COMPLITED
```

To terminate this command press `**RESET KEY**' on the Kit.

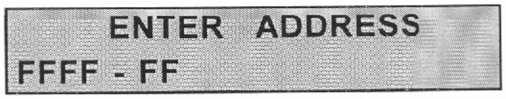
N - Single Stepping

The Single Stepping command use to execute the entered program in single stepping mode.



```
COMMAND : A, B, C, D, E, F
 G, I, L, M, N, P, R, S, T, Y
```

Simply Press the key ‘**N**’. After the command, the LCD screen will be display as below.

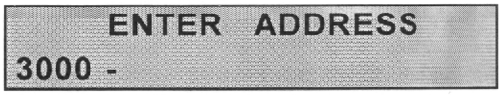


```
      ENTER  ADDRESS
FFFF - FF
```

Syntax :

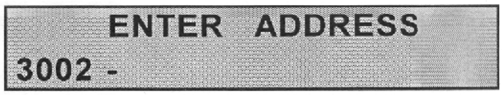
<RAM Starting Address> <Space Bar>

If the Program starting address is 3000 location as per example given in the (manual) then type RAM starting address 3000.



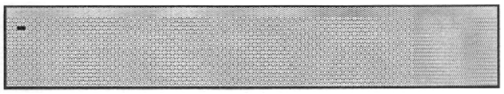
ENTER ADDRESS
3000 -

Now press `**SPACE BAR**' key on the ASCII Keyboard to Single Step the first instruction. Now the LCD screen will display as below.



ENTER ADDRESS
3002 -

To do further single stepping press `**SPACE BAR**' Key. To terminate this command press `**ENTER**' key. The display on the LCD screen as below.



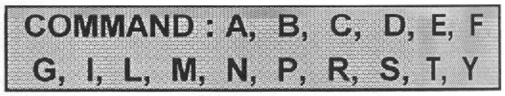
-

Note :

To Examine Registers while doing single stepping don't press `**RESET KEY**'.

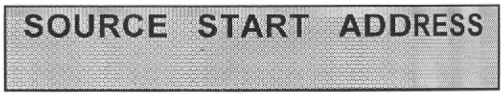
B - Block Move

The Block Move command use to move the block from one memory area block location to another memory area block location.



COMMAND : A, B, C, D, E, F
G, I, L, M, N, P, R, S, T, Y

Simply Press the key '**B**'. After the command, the LCD screen will be display as below.

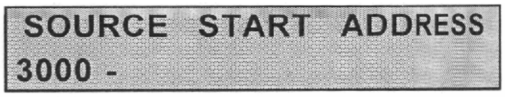


SOURCE START ADDRESS

Syntax :

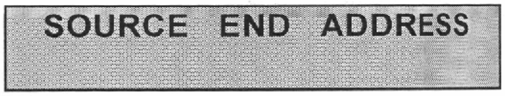
<Source Start Address> <Space> <Source End Address> <Space> <Destination Start Address> <ENTER>

Block Move the Program as per example given in the (manual) i.e. 3000 - 3006 to 4000 destination location.



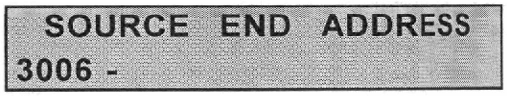
SOURCE START ADDRESS
3000 -

Now press `**SPACE BAR**' key on the ASCII Keyboard and the LCD screen will display as below.



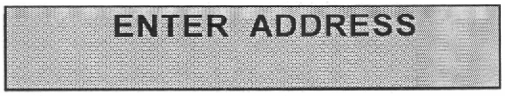
SOURCE END ADDRESS

Now enter the End Address i.e. 3006 and the LCD screen will display as below.



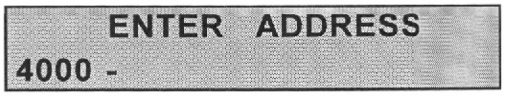
SOURCE END ADDRESS
3006 -

Press `**SPACE BAR**' key on the ASCII Keyboard and the LCD screen will display as below.



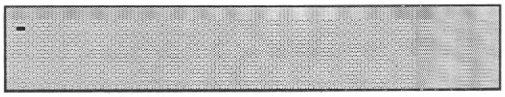
ENTER ADDRESS

Now enter the Destination Starting Address i.e. 4000 and the LCD screen will display as below.



ENTER ADDRESS
4000 -

Press `**ENTER**' key on the ASCII Keyboard and the LCD screen will display as below.

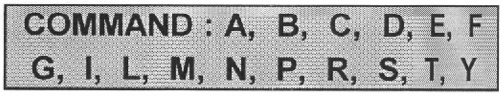


-

By using Examine/Modify Memory i.e. `**M**' command, one can verify the Block Move data in 4000 location.

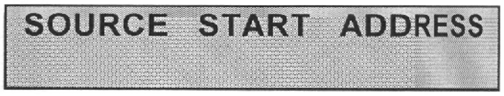
F - Block Fill

The Block Fill command use to fill the block of memory with a constant data.



COMMAND : A, B, C, D, E, F
G, I, L, M, N, P, R, S, T, Y

Simply Press the key '**F**'. After the command, the LCD screen will be display as below.

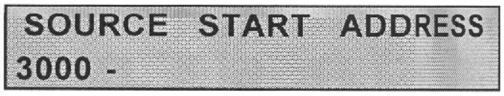


SOURCE START ADDRESS

Syntax :

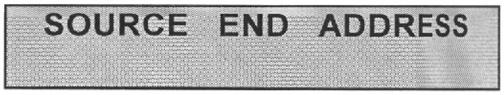
<Source Start Address> <Space> <Source End Address> <Space> <DATA> <ENTER>

Fill the Block with a constant data 55 from 3000 starting location to 30FF ending location.



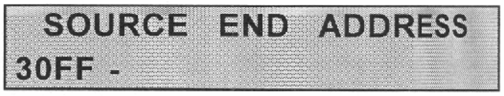
SOURCE START ADDRESS
3000 -

Now press `**SPACE BAR**' key on the ASCII Keyboard and the LCD screen will display as below.



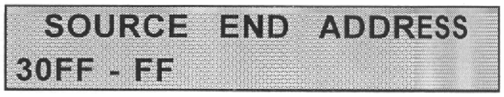
SOURCE END ADDRESS

Now enter the End Address i.e. 30FF and the LCD screen will display as below.



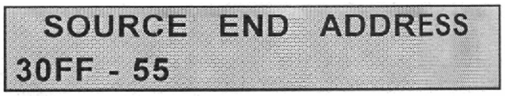
SOURCE END ADDRESS
30FF -

Press `**SPACE BAR**' key on the ASCII Keyboard and the LCD screen will display as below.



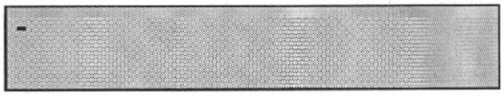
SOURCE END ADDRESS
30FF - FF

Now enter the Data to fill the memory block and the LCD screen will display as below.



SOURCE END ADDRESS
30FF - 55

Press `**ENTER**' key on the ASCII Keyboard and the LCD screen will display as below.

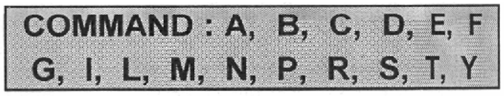


-

By using Examine/Modify Memory i.e. `**M**' command, one can verify the Block Fill data in 3000 location.

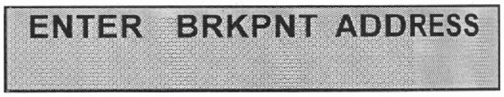
T - Set Break Point

The Set Break Point command use to Set the Break Point at desired address.



COMMAND : A, B, C, D, E, F
G, I, L, M, N, P, R, S, T, Y

Simply Press the key '**T**'. After the command, the LCD screen will be display as below.

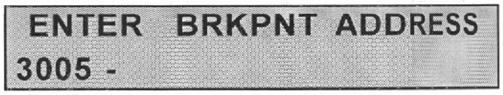


ENTER BRKPNT ADDRESS

Syntax :

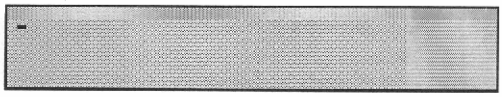
<Address of the Break Point to be set> <ENTER>

As per the Example given in the (manual) one want to set a Break Point at 3005 location. The Break Point address should always at the end of instruction.



```
ENTER  BRKPNT ADDRESS
3005 -
```

Press `**ENTER**' key on the ASCII Keyboard and the LCD screen will display as below.

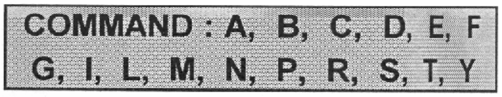


```
-
```

Now your Break Point has set at address 3005 location.

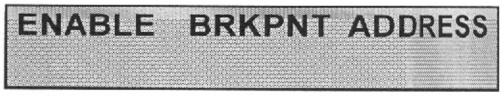
E - Enable Break Point

The Enable Break Point command use to enable the Set Break Point address.



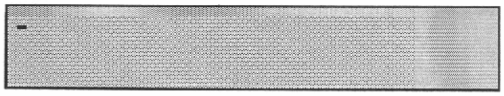
```
COMMAND : A, B, C, D, E, F
 G, I, L, M, N, P, R, S, T, Y
```

Simply Press the key ‘**E**’. After the command, the LCD screen will be display as below.



```
ENABLE  BRKPNT ADDRESS
```

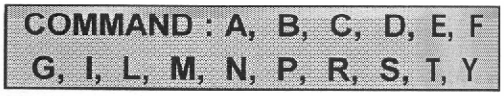
Press `**ENTER**' key on the ASCII Keyboard and the LCD screen will display as below.



```
-
```

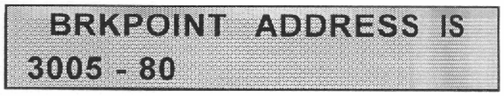
I - View Break Point

The View Break Point command use to View the Set Break Point address.



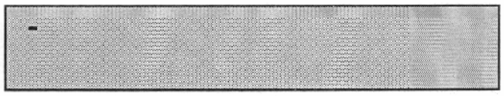
```
COMMAND : A, B, C, D, E, F
 G, I, L, M, N, P, R, S, T, Y
```

Simply Press the key ‘**I**’. After the command, the LCD screen will be display as below.



```
  BRKPOINT  ADDRESS IS
 3005 - 80
```

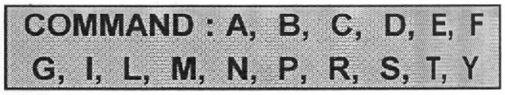
After viewing the Break Point press `**ENTER**' Key and the LCD screen will display as below.



```
 -
```

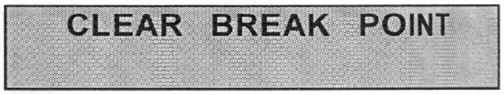
C - Clear Break Point

The Clear Break Point command use to Clear the Set Break Point address.



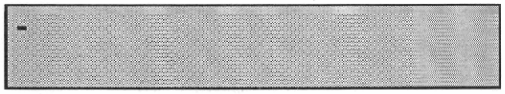
```
COMMAND : A, B, C, D, E, F
 G, I, L, M, N, P, R, S, T, Y
```

Simply Press the key ‘**C**’. After the command, the LCD screen will be display as below.



```
CLEAR BREAK POINT
```

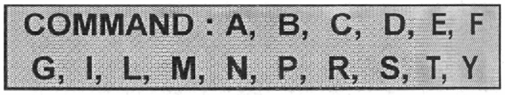
Press `**ENTER**' key on the ASCII Keyboard and the LCD screen will display as below.



```
-
```

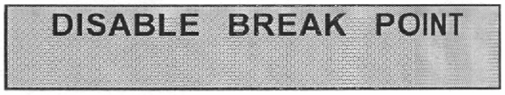
L - Disable Break Point

The Disable Break Point command use to Disable the Set Break Point address.



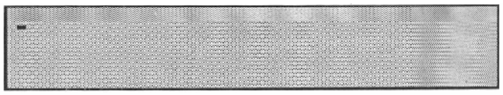
```
COMMAND : A, B, C, D, E, F
 G, I, L, M, N, P, R, S, T, Y
```

Simply Press the key ‘**L**’. After the command, the LCD screen will be display as below.



```
DISABLE BREAK POINT
```

Press `**ENTER**' key on the ASCII Keyboard and the LCD screen will display as below.



```
-
```

How to Use the Break Point Commands

To set the Break Point follow the below steps. While working in Break Point don't press `**RESET KEY**' in between.

Step 1

Set the Break Point address at 3005 using **Set Break Point `T'** command illustrated in (Keyboard Command section).

Step 2

Now Enable the Set Break Point address 3005 using **Enable Break Point `E'** command illustrated in (Keyboard Command section).

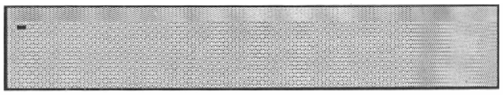
Step 3

View the Set Break Point address using **View Break Point `I'** command illustrated in (Keyboard Command section).

Step 4

Execute the program from 3000 location using **Executing `G'** command illustrated in (Keyboard Command section).

After that, LCD screen will display as follows.



Step 4

Now you can examine all registers using **Examine/Modify Register `R'** command illustrated in (Keyboard Command section).

Serial Command

M51-02 provides the following on-board Serial interfaces.

1. RS-232C interface through UART of the 8051 Micro controller chip.

2. AUX RS-232C interface through 8251 USART chip.

RS-232C Interface

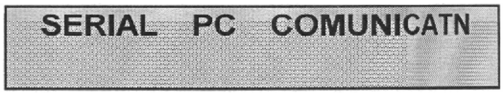
The RS-232C interface provided by 8051 chip is a three line interface i.e. the handshake lines are not provided. The three signals namely GND, TX & RX are brought out at Connector CN7. Through this RS-232 interface user can communicate with PC by selecting baudrate of 2400 bps.

The user can communicate 8051 Kit with PC as below procedure mentioned.

1. Connect Serial cable at CN8 of the Kit to any COM-1 OR COM-2 of the PC.

2. Run **Windows Hyper Terminal** Software using baudrate of 2400 bps.

3. Press `**Y**' key on M51-02 Kit ASCII keyboard & the LCD screen will display.



SERIAL PC COMUNICATN

4. Now you will see Command Prompt message in the PC Screen.

M51-02> SERIAL COMM.

COMMAND>

5. Now you can use all serial commands.

AUX RS-232C Interface (Optional)

The AUX RS-232C interface is provided on M51-02 through Intel's USART chip 8251 (Universal Synchronous Asynchronous Receiver Transmitter). The 8251 uses Timer 0 of 8253 for baud rate generation. It is a seven line interface with all seven signal lines being brought out at 9 pin D-type Connector CN9.

Using Hyper Terminal WIN98 / 2000/NT/XP With M51-02 KIT

Introduction :

HYPER TERMINAL is a simple terminal emulator Windows based software for IBM-PC/AT compatible computers to allows the user to communicate with the computer through serial port with the facility of downloading & uploading of the data between the computer and the other serial device. The various communication parameters like baud rate (speed), number of data bits, stop bits, parity etc. can be changed. The package communicates through COM1: as well as COM2: port of the IBM-PC Systems.

Installation :

If your computer doesn't have Hyper Terminal software please follow the below procedure.

1. First go to Control Panel and Click Add/Remove Program.

2. Click Windows Setup.

3. Select Communication and Click Details.

4. Select Hyper Terminal then press OK and Click Apply.

5. Computer will ask for Window CD for Installing Hyper Terminal.

6. After Installing Hyper Terminal Click cursor on **<Start>** **<Programs>** **<Accessories>** **<Communications>** **<Hyper Terminal>** then open it.

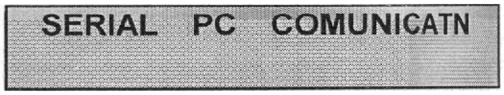
7. Click on **<Hypertrm>** icon give **ABC** in name Block and Click **OK.**

8. Select Communication Port either **<Direct to COM1**> or **<Direct to COM2>** and click **OK.**

9. For connecting the kit of 8051 TX/RX at connector CN8 select following COM1/COM2 properties:

Bits per second	**9600**
Data bits	**8**
Parity	**None**
Stop bit	**1**
Flow Control	**XON/XOFF**

a. Press **`Y**' key on M51-02 Kit ASCII keyboard & the LCD screen will display.



SERIAL PC COMUNICATN

b. Now you will see Command Prompt message in the PC Screen.

```
M51-02> SERIAL COMM.
COMMAND>
```

c. Now you can use all serial commands.

The following commands are available through Serial Mode. Use only **CAPITAL** letters for executing the commands.

1.	Dump Data Memory	**:**	DD
2.	Dump Program Memory	**:**	DP
3.	Enter Data Memory	**:**	ED
4.	Enter Program Memory	**:**	EP
5.	Fill Memory	**:**	FL
6.	Data To Data Block Move	**:**	BD
7.	Program To Program Block Move	**:**	BP
8.	Data To Program Block Move	**:**	BM
9.	Single Instruction	**:**	SI
10.	Go (Execute)	**:**	GO
11.	Register Display	**:**	RG
12.	Set Breakpoint	**:**	SB
13.	Display Breakpoint	**:**	DI
14.	Clear Breakpoint	**:**	CB
15.	Enable Breakpoint	**:**	EB
16.	Disable Breakpoint	**:**	DB
17.	Download	**:**	DL
18.	Upload	**:**	UP
19.	Exit	**:**	EX

Dump Data Memory (DD)

This command dumps the data memory between and including two specified address separated by a space.

Syntax

DD <starting address> <end address> <CR>

The starting address should not be higher than the end address.

Dump Program Memory (DP)

This command dumps the program memory between and including two specified addresses separated by a space.

Syntax

DP <starting address> <end address> <CR>

The starting address should not be greater than the end address.

Enter Data Memory (ED)

This command is used for entering data bytes in to data memory area.

Syntax

ED <starting address> <CR>

On pressing <CR> the PC content is incremented and on pressing <Space bar> the PC content is decremented which enables the user to examine/modify the data of the location shown. For coming out of this command press Esc.

Enter Program Memory (EP)

This command is used for entering program in to the program memory area

Syntax

EP <starting address> <CR>

For coming out of this command press Esc.

Fill Memory (FL)

This command is used to fill a memory area with a constant.

Syntax

FL <starting address> <end address> <byte to be filled> <CR>

Data To Data Block Move (BD)

This command is used for moving block of data memory area to another destination address of data memory area.

Syntax

BD <starting address of block> <end address of block>

<destination address> <CR>

Program To Program Block Move (BP)

This command is used for moving a block of program memory area to another destination address within program memory area.

Syntax

BP <starting address of block> <end address of block>

<destination address> <CR>

Data To Program Memory Block Move (BM)

This command is used for moving a block of data memory area to program memory area.

Syntax

BM <starting address of data block> <end address of data block>

<destination address of program memory block> <CR>

Single Instruction (SI)

This command is used for running a program in single instruction mode i.e. instruction by instruction.

Syntax

SI <starting address> <CR>

After the execution of an instruction, the contents of all the registers are displayed and system waits for a Carriage Return to be pressed for the execution of the next instruction. For coming out of this command press Esc key.

Execution In Full Speed (GO)

This command is used to execute a program in full clock speed mode.

Syntax

GO <starting address> <CR>

Register Display (RG)

This command displays the contents of all the registers of 8031 controller.

Syntax

RG <CR>

Set Breakpoint (SB)

This command is used to set a breakpoint within a program.

Syntax

SB <breakpoint address> <CR>

This breakpoint will be active only if it is enabled by using enable command.

Display Breakpoint (DI)

This command displays the breakpoint address.

Syntax

DI <CR>

Clear Breakpoint (CB)

This command clears the breakpoint set earlier and restores the program at the breakpoint address (if changed).

Syntax

CB <CR>

Enable Breakpoint (EB)

This command enables a breakpoint. The program stops at break point address only if it is enabled.

Syntax

EB <CR>

Disable Breakpoint (DB)

This command disables a breakpoint. Breakpoint address will not be useful if it is disabled by this command.

Syntax

DB <CR>

Downloading

Downloading the ABC.HEX from Computer to Trainer

The following procedure is to be adopted for downloading the file from PC to Trainer Kit.

The `DL' command loads the data from your diskette/PC to the memory of the Kit.

1. Type `DL' on PC screen & Press 'Enter' Key on the PC keyboard.
2. Select **<Transfe**r> **<Send Text File..**.> Choose File Name i.e. **ABC.HEX** and Click it.
3. The program stored in the file will be loaded in memory of Trainer at the address specified in the program.

Note :

If the file is not Downloading please Change the Line Delay & Character Delay in the Properties of the Hyper Terminal. To Change the Line Delay & Character Delay select **<Properties> <Setting> <ASCII Setup>**.

Uploading

Uploading the ABC.HEX from Trainer to Computer

The following procedure is to be adopted for uploading the file from Trainer to PC.

1. Type `**UP**’ on the PC Screen through PC keyboard

2. Write Start Address and End Address

3. Then Go in Menu bar. Select **<Transfe**r> **<Capture Text..**.> from PC keyboard give any name with **HEX** Extension i. e. **ABC.HEX** (You can Browse where you want to save it) Click on **<Star**t>

4. Press **‘Enter’** Key of PC. Address and Data will seen on PC Screen.

5. Then again Go in Menu bar. Select **<Transfe**r> **<Capture Tex**t> click on **<Sto**p>

6. Data captured will be stored in the file as defined by the file name.

Memory/IO Mapping

Memory Mapping :

The 8051 chip supports 64K bytes of program memory and 64K bytes of data memory. Since in a practical situation, the user may need the same memory area for Data as well as programs, the system M51-02 has been designed such a way that the same memory chip can be used as program/data memory. There are three memory sockets namely MEM0, MEM1 & M3M2.

MEM0 is defined as Program Memory which is used for Monitor Program. 16K Bytes of Powerful Monitor Program is Embedded in the 27C512 EPROM. The Program Memory address is from `0000 to 3FFF'.

MEM1 is defined as user Program Memory / Data Memory. 32K bytes of RAM (62256) can be used for Program Memory / Data Memory for the user RAM. The address is from `3000 to AFFF' for the user RAM.

MEM2 is defined as expenction for Program Memory / Data Memory. 16K bytes of RAM can be used for Program Memory / Data Memory. The address is from `B000 to EFFF'.

I/O Mapping :

Various chips used in M51-02 are 8255, 8253, RTC ADC, DAC and 8251. The addresses for these I/O devices are given below:

Active range Port Addresses	Port Numbers		Selected Device
(FF00-FF03)	8255-I		Programmable Peripheral Interface
	FF00	Port A	
	FF01	Port B	
	FF02	Port C	
	FF03	Control Word	
(FF04-FF07)	8255-II		Programmable Peripheral Interface
	FF04	Port A	
	FF05	Port B	
	FF06	Port C	
	FF07	Control Word	
(FF0C-FF0F)	8253		Programmable Interval Timer
	FF0C	Counter 0	
	FF0D	Counter 1	
	FF0E	Counter 2	
	FF0F	Control Word	

Subroutines

M51-02 Monitor uses certain subroutines for its operation, which can also be used by the user of his programs. The addresses of these routines and their descriptions are given here:

	Address	**Label**	**Description**
1.	0AC1	RDADR:	this routine reads address a has terminator, no other register is destroyed
2.	16A5	CPRMPT:	this routine displays prompt TERM : on CRT no register is destroyed
3.	19D3	SERDISP:	this routine displays message on the PC screen in serial terminal emulation mode Input : DPTR has message table address 24H indicates end of message
4.	19DF	TRNSM:	this routine outputs the byte on RS-232C serial port Input: accumulator has byte to be outputted no other register affected.
5.	1A14	RECV	this routine inputs character from serial port into accumulator, no other register affected.
6.	1A6B	TRANS	this routine unpacks the hex byte into two hex nibbles then transmits ASCII, equivalents of the two one RS-232C serial port Input: a has the hex byte no register is affected.
7.	0B77	GETCDE	this routine searches the table Input: a has byte to be searched in table, DPTR has starting address of table on-chip ram location, 4E has number enters in table Output: a has position number of character in table if found else carry is set indicating character not found.
8.	1558	LCDINT	this routine initialize LCD.
9.	1581	SWRITE	this routine accepts data from accumulator and treat it as command word for LCD.
10.	1591	WRITE	this routine accepts data from accumulator and treat it as data to be displayed on LCD.
11.	0AA1	GETKEY	this routine wait for key to be pressed and put corresponding ASCII codes in accumulator.
12.	0A3C	DISPLAY	this routine displays DPTR on ROW 1 of LCD

	Address	Label	Description
13.	0A71	DISPLAYT	this routine displays DPTR on ROW 2 of LCD
14.	154CH	HF80H	this routine called as a command word to write characters starting from ROW-1 COLOUM-1
15.	1552H	HFC0H	this routine called as a command word to write characters starting from ROW-2 COLOUM-2
16.	5A40	ASCINREG	this routine prints the character whose ASCII code is stored in R7
17.	159CH	DELAY1	this is delay to be called after giving any command to LCD module, destroys register R3 & R6
18.	5A47	PRNSTR	this routine display ASCII string in first ROW on LCD
19.	5A50	RAMCON	this routine displays RAM contents on LCD screen
20.		ASCINREG	this routine prints the character whose ASCII code is stored in R7 Syntax : MOV R7,#DATA LCALL 5A40H Input Register:-R7 Output Register:-NONE
21.		PRNSTR	display ASCII string in a first row of LCD press space to RESET Syntax: MOV DPTR,#STARTING ADDR OF STR LCALL 5A47H 5A47H:-routine used are display & GETKEY "PRESS SPACE TO CONTINUE" Input Register:-DPTR Output Register:-NONE
22.		RAMCON2	displays contents of RAM location on LCD screen Syntax: MOV DPTR,#ADDR OF RAM whose data to be printed LCALL 5A50H

Address	Label	Description
		5A50H:- displays contents of DPTR register given on LCD screen "PRESS SPACE TO CONTINUE" destroys register R1 & R2 Input Register:-DPTR Output Register:-NONE
23.	RAMCON3	display ASCII code on the LCD screen from any RAM location Syntax : LCALL 547BH MOV DPTR,#ADDR of RAM whose data to be print ASCII LCALL 5A85H 5A7BH: routine used for initialization of LCD 5A50H: this routine converts HEX to ASCII of given DPTR register's ADDR, routine used are display & GETKEY "PRESS SPACE TO CONTINUE" Input Register:-DPTR Output Register:-NONE.
24.	RAMCON1	displays contents of RAM location on LCD screen Syntax: MOV DPTR,#ADDR of RAM whose data to be printed LCALL 5AA2H 5A50H:- displays contents of DPTR register given on LCD screen "WITHOUT SPACE TO CONTINUE" destroys register R1 & R2 Input Register:- DPTR Output Register:- NONE

Address	Label	Description
25.	RAMCON4	display ASCII code on the LCD screen from any RAM location Syntax : LCALL 5A7BH MOV DPTR,#ADDR OF RAM whose data to be print ASCII LCALL 5A9EH 5A7BH: routine used fro initialization of LCD "WITHOUT SPACE TO CONTINUE" Input Register:-DPTR Output Register:- NONE

Interrupt

There are two external interrupts called INT0 & INT1 are coming from CPU. Interrupt INT0 is used for keyboard and interrupt INT1 is free for the user. The different interrupts, Timer, Serial Interface addresses are as follows:

			RAM Location
INT0	-	0003	3DF0
INT1	-	0013	3DF6
T0	-	000B	3DF3
T1	-	001B	3DF9
Rx + Tx	-	0023	3DFC

These external interrupts, timer, serial interface can not be used directly because our monitor area starts from 0000 to 1FFF. So we are given jump location at RAM area of corresponding above facility

Sample Programs

Programs are given here to the user to understand the programming techniques of 8051 microcontrollers.

Program-1 :

Flashing Display of `WELCOME M51-02 KIT' On LCD Screen. executing this program from address 3000H, “ **WELCOME M51-02 KIT** ” message flashes on the display of the kit.

Address	Code	Label	Mnemonic	Operand	Comments
3000	90 30 1E	HERE:	MOV	DPTR,#301E	WELCOME M51-02 KIT message
3003	12 0A 3C		LCALL	0A3C	display routine
3006	7B 00		MOV	R3,#0	
3008	7A 00	LOOP2:	MOV	R2,#0	
300A	DA FE	LOOP1:	DJNZ	R2,LOOP1	delay code
300C	DB FA		DJNZ	R3,LOOP2	
300E	90 30 32		MOV	DPTR,#3032	blank message
3011	12 0A 3C		LCALL	0A3C	display routine
3014	7B 00		MOV	R3,#0	
3016	7A 00	LOOP4:	MOV	R2,#0	
3018	DA FE	LOOP3:	DJNZ	R2,LOOP3	delay code
301A	DB FA		DJNZL	R3,LOOP4	
301C	80 E2		SJMP	3000	Display message `WELCOME M51-02 KIT
301E	57 45 4C 43 4F 4D		DFB	57H, 45H, 4CH, 43H, 4FH, 4DH	
	3024 45 20 4D 35 31 2D		DFB	45H, 20H, 4DH, 35H, 31H, 2DH	
302A	30 32 20 4B 49 54		DFB	30H, 32H, 20H, 4BH, 49H, 54H to Blank Display	
3032	20 20 20 20 20 20		DFB	20H, 20H, 20H, 20H, 20H, 20H	
3038	20 20 20 20 20 20		DFB	20H, 20H, 20H, 20H, 20H, 20H	
303E	20 20 20 20 20 20		DFB	20H, 20H, 20H, 20H, 20H, 20H	

Program-2:

Hexadecimal Addition of Two Numbers

On executing the program from 3000H enter the address message displayed. Enter the first two digit operand then press enter, enter next two digit. Operand. sum of two numbers is displayed if sum is greater than two digits.

Address	Code	Label	Mnemonic	Operand	Comments
3000	12 0A C1	START:	LCALL	0AC1H	read 1st operand
3003	A3		INC	DPTR	
3004	E0		MOVX	A,@DPTR	
3005	F5 0A		MOV	0AH,A	
3007	12 0A C1		LCALL	0AC1H	read 2nd operand
300A	A3		INC	DPTR	
300B	E0		MOVX	A,@DPTR	
300C	25 0A		ADD	A,0AH	ADD Hex
300E	90 4E 00		MOV	DPTR,#4E00H	
3011	F0		MOVX	@DPTR,A	
3012	90 4E 00		MOV	DPTR,#4E00H	
3015	F0		MOVX	@DPTR,A	
3016	54 F0		ANL	A,#0F0H	
3018	C4		SWAP	A	
3019	90 4E 01		MOV	DPTR,#4E01H	
301C	F0		MOVX	@DPTR,A	
301D	90 4E 00		MOV	DPTR,#4E00H	
3020	E0		MOVX	A,@DPTR	
3021	54 0F		ANL	A,#0FH	
3023	90 4E 02		MOV	DPTR,#4E02H	
3026	F0		MOVX	@DPTR,A	
3027	12 15 58		LCALL	1558H	display result
302A	90 4E 01		MOV	DPTR,#4E01H	
302D	E0		MOVX	A,@DPTR	
302E	12 0B 77		LCALL	0B77H	get code
3031	12 15 91		LCALL	1591H	write

Address	Code	Label	Mnemonic	Operand	Comments
3034	90 4E 02		MOV	DPTR,#4E02H	
3037	E0		MOVX	A,@DPTR	
3038	12 0B 77		LCALL	0B77H	get code
303B	12 15 91		LCALL	1591H	write
303E	12 0A A1	LP1:	LCALL	0AA1H	get key space
3041	B4 20 FA		CJNE	A,#20H,LP1	
3044	02 30 00		LJMP	START	

Program-3 :

Decimal Addition of Two Numbers

On executing the program from 3000 `ENTER ADDRESS' appears on LCD. Enter the first operand in decimal and press <ENTER>. `ENTER ADDRESS' appears again. Enter the next operand in decimal and press <ENTER>. The sum of two no. is displayed. If the sum is greater that two digits, the carry is ignored in the result.

Address	Code	Label	Mnemonic	Operand	Comments
3000	12 0A C1	START:	LCALL	0AC1H	read 1st operand
3003	A3		INC	DPTR	
3004	E0		MOVX	A,@DPTR	
3005	F5 0A		MOV	0AH,A	
3007	12 0A C1		LCALL	0AC1H	read 2nd operand
300A	A3		INC	DPTR	
300B	E0		MOVX	A,@DPTR	
300C	25 0A		ADD	A,0AH	add hex
300E	D4		DA	A	decimal adjust acc
300F	90 4E 00		MOV	DPTR,#4E00H	
3012	F0		MOVX	@DPTR,A	
3013	12 15 58		LCALL	1558H	display result
3016	90 4E 00		MOV	DPTR,#4E00H	
3019	E0		MOVX	A,@DPTR	
301A	54 F0		ANL	A,#0F0H	
301C	C4		SWAP	A	
301D	12 0B 77		LCALL	0B77H	
3020	12 15 91		LCALL	1591H	

Address	Code	Label	Mnemonic	Operand	Comments
3023	90 4E 00		MOV	DPTR,#4E00H	
3026	E0		MOVX	A,@DPTR	
3027	54 0F		ANL	A,#0FH	
3029	12 0B 77		LCALL	0B77H	
302C	12 15 91		LCALL	1591H	
302F	12 0A A1	LP1:	LCALL	0AA1H	get key space
3032	B4 20 FA		CJNE	A,#20H,LP1	
3035	02 30 00		LJMP	START	

Program-4 :

This program Reads a hex byte from the keyboard of the kit and split it into two nibbles and stores MSB in 3050H and LSB in 3051H. On executing the program enter the data and press Enter.

Address	Code	Label	Mnemonic	Operand	Comments
3000	12 0A C1	START:	LCALL	0AC1H	read operand
3003	A3		INC	DPTR	
3004	E0		MOVX	A,@DPTR	
3005	FB		MOV	R3,A	
3006	54 0F		ANL	A,#0FH	
3008	C4		SWAP	A	
3009	90 30 50		MOV	DPTR,#3050H	
300C	F0		MOVX	@DPTR,A	
300D	EB		MOV	A,R3	
300E	54 0F		ANL	A,#0FH	
3010	90 30 51		MOV	DPTR,#3051H	
3013	F0		MOVX	@DPTR,A	
3014	02 30 00		LJMP	START	

Program-5 :

This program is to see if the no. is odd or even. On executing program asks for word. Enter the word and press `ENTER'. If the word odd `NUMBER IS ODD' display on LCD screen. If word is even `NUMBER IS EVEN' will display on LCD screen.

Address	Code	Label	Mnemonic	Operand	Comments
3000	12 0A C1	START:	LCALL	0AC1H	read operand
3003	A3	INC	DPTR		
3004	E0	MOVX	A,@DPTR		
3005	44 FE		ORL	A,#0FEH	
3007	B4 FE 05		CJNE	A,#0FEH,ODD	
300A	90 30 2B		MOV	DPTR,#MSG2	
300D	80 03		SJMP	DISP	
300F	90 30 17	ODD:	MOV	DPTR,#MSG1	
3012	12 0A 3C	DISP:	LCALL	0A3CH	display message
3015	80 FE	HERE:	SJMP	HERE	
3017	45 55 4D 42	MSG1:	DFB	45H,55H,4DH,42H	
301B	45 52 20 49		DFB	45H,52H,20H,49H	
301F	53 20 4F 44		DFB	53H,20H,4FH,44H	
3023	44 20 20 20		DFB	44H,20H,20H,20H	
3027	20 20 20 20		DFB	20H,20H,20H,20H	
302B	45 55 4D 42	MSG2:	DFB	45H,55H,4DH,42H	
302F	45 52 20 49		DFB	45H,52H,20H,49H	
3033	53 20 45 56		DFB	53H,20H,45H,56H	
3037	45 4E 20 20		DFB	45H,4EH,20H,20H	
303B	20 20 20 20		DFB	20H,20H,20H,20H	

Program-6 :

Hex multiplication.

Reads two hex bytes from keyboard multiplies them and displays result.

Address	Code	Label	Mnemonic	Operand	Comments
3000	12 0A C1	START:	LCALL	0AC1H	read 1st operand
3003	A3		INC	DPTR	
3004	E0		MOVX	A,@DPTR	
3005	F5 F0		MOV	0F0H,A	

Address	**Code**	**Label**	**Mnemonic**	**Operand**	**Comments**
3007	12 0A C1		LCALL	0AC1H	read 2nd operand
300A	A3		INC	DPTR	
300B	E0		MOVX	A,@DPTR	
300C	A4		MUL	AB	multiply A&B
300D	90 4E 00		MOV	DPTR,#4E00H	
3010	F0		MOVX	@DPTR,A	
3011	12 15 58		LCALL	1558H	display result
3014	90 4E 00		MOV	DPTR,#4E00H	
3017	E0		MOVX	A,@DPTR	
3018	54 F0		ANL	A,#0F0H	
301A	C4		SWAP	A	
301B	12 0B 77		LCALL	0B77H	
301E	12 15 91		LCALL	1591H	
3021	90 4E 00		MOV	DPTR,#4E00H	
3024	E0		MOVX	A,@DPTR	
3025	54 0F		ANL	A,#0FH	
3027	12 0B 77		LCALL	0B77H	
302A	12 15 91		LCALL	1591H	
302D	12 0A A1	LP1:	LCALL	0AA1H	get key space
3030	B4 20 FA		CJNE	A,#20H,LP1	
3033	02 30 00		LJMP	START	

Program 7 :

This program displays a message “GOOD 51” on PC Screen. The following program will work in Serial Mode and message will display in PC Screen monitor.

Address	**Code**	**Label**	**Mnemonic**	**Operand**	**Comments**
3000	12 5A CF		LCALL	5ACFH	
3003	90 30 0C		MOV	DPTR,#300C	
3006	12 19 D3		LCALL	19D3H	serial display
3009	02 16 A5		LJMP	16A5	Cmmd. prmpt.
300C	47 4F 4F 44 20	DFB;	47H, 4FH, 4FH, 44H, 20H		
3011	35 31	DFB;	35H, 31H		

Program-8:

This program reads a key from the 101 ASCII keyboard of PC and split it into two nibbles and stores MSB nibble in 3050H M.L. and LSB in 3051H M.L.

Address	Code	Label	Mnemonic	Operand	Comments
3000	12 5A CF		LCALL	5ACFH	
3003	12 1A 14		LCALL	1A14H	receive data
3006	FB		MOV	R3,A	
3007	54 F0		ANL	A,#F0	
3009	C4		SWAP	A	
300A	F8		MOV	R0,A	
300B	EB		MOV	A,03	
300C	54 0F		ANL	A,#0F	
300E	90 30 50		MOV	DPTR,#3050H	
3011	F0		MOVX	@ DPTR,A	
3012	A3		INC	DPTR	
3013	E8		MOV	A,00	
3014	F0		MOVX	@DPTR,A	
3015	02 16 A5		LJMP	16A5H	

Program-9:

This program displays a character on PC Screen when ASCII code is in accumulator.

Address	Code	Label	Mnemonic	Operand	Comments
3000	12 5A CF		MOV	5ACFH	
3003	74 44		MOV	A,#44	
3005	12 19 DF		LCALL	19DFH	trnsm
3008	02 16 A5		LJMP	16A5H	

Program-10 :

This program displays the no. in accumulator on PC Screen.

Address	Code	Label	Mnemonic	Operand	Comments
3000	12 5A CF		MOV	5ACFH	
3003	74 44		MOV	A,#44	
3005	12 1A 6B		LCALL	1A6BH	trnsm
3008	02 16 A5		LJMP	16A5H	

Connectors

The pin details of the various connectors are given here for your reference.

Bus Connector-CN1

Pin no.	Signal	Pin no.	Signal
1	P1.0	21	P3.1/TXD
2	VCC	22	ALE
3	P1.1	23	P3.2/INT0
4	AD0	24	NC
5	P1.2	25	P3.3/INT1
6	AD1	26	P2.7/A15
7	P1.3	27	P3.4/T0
8	AD2	28	P2.6/A14
9	P1.4	29	P3.5/T1
10	AD3	30	P2.5/A13
11	P1.5	31	P3.6/WR
12	AD4	32	P2.4/A12
13	P1.6	33	P3.7/RD
14	AD5	34	P2.3/A11
15	P1.7	35	NC
16	AD6	36	P2.2/A10
17	RESET	37	NC
18	AD7	38	P2.1/A9
19	P3.0/RXD	39	GND
20	NC	40	P2.0/A8

Bus Connector-CN2

Pin	Signal	Pin	Signal
1	A3	6	A5
2	A7	7	A0
3	A2	8	A4
4	A6	9	CS-IO
5	A1	10	RST

8253 Connector-CN5

Pin	Signal	Pin	Signal
1	CLK-0	6	CLK-1
2	OUT-0	7	GATE-2
3	GATE-0	8	OUT-2
4	OUT-1	9	CLK-2
5	GATE-1	10	GND

8255-II Connector-CN4

Pin no.	Signal	Pin no.	Signal
1	PC4	14	PB1
2	PC5	15	PA6
3	PC2	16	PA7
4	PC3	17	PA4
5	PC0	18	PA5
6	PC1	19	PA2
7	PB6	20	PA3
8	PB7	21	PA0
9	PB4	22	PA1
10	PB5	23	PC6
11	PB2	24	PC7
12	PB3	25	GND
13	PB0	26	VCC

8255-I Connector-CN3

Pin no.	Signal	Pin no.	Signal
1	PC4	9	PB4
2	PC5	10	PB5
3	PC2	11	PB2
4	PC3	12	PB3
5	PC0	13	PB0
6	PC1	14	PB1
7	PB6	15	PA6
8	PB7	16	PA7

Pin no.	Signal	Pin no.	Signal
17	PA5	22	PA1
18	PA5	23	PC6
19	PA2	24	PC7
20	PA3	25	GND
21	PA0	26	VCC

Power Supply Connector-CN7

Pin	Signal
1	+5V
2	GND
3	+12V
4	-12V

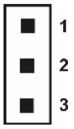
8051 TX/RX Serial Connector-CN8

Pin	Signal
1	RX
2	TX
3	GND
4	GND

8251 RS-232C Serial Connector-CN9

Pin	Signal	Pin	Signal
1	NC	5	GND
2	Rxd	6	DSR
3	Txd	7	RTS
4	DTR	8	CTS
		9	NC

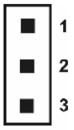
Jumper Details



JP 1

Short 1 & 2 for +5V to RAM

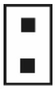
Short 2 & 3 for Battery Supply to RAM



JP 2

Short 1 & 2 for +5V reference to ADC

Short 2 & 3 for +10V reference to ADC



JP 3

Short JP3 to give +5V to 26 pin of CN5

APPENDIX

Instructions That Affect Flag Settings

Instruction		FLAG		
		CY	OV	AC
ADD		x	x	x
ADDC		x	x	x
SUBB		x	x	x
MUL		0	x	
DIV		0	x	
DA		x		
RRC		x		
RLC		x		
CJNE		x		
SETB	C	1		
CLR	C	0		
CPL	C	x		
ANL	C,bit	x		
ANL	C,/bit	x		
ORL	C,bit	x		
ORL	C,/bit	x		
MOV	C,bit	x		

8051 Instruction Set

Arithmetic Operations

Mnemonic		Byte	Cyc
ADD	A,@Ri	1	1
ADD	A,Rn	1	1
ADD	A,direct	2	1
ADD	A,#data	2	1
ADDC	A,@Ri	1	1
ADDC	A,Rn	1	1

Mnemonic		Byte	Cyc
ADDC	A,direct	2	1
ADDC	A,#data	2	1
SUBB	A,@Ri	1	1
SUBB	A,Rn	1	1
SUBB	A,direct	2	1
SUBB	A,#data	2	1
INC	A	1	1
INC	@Ri	1	1
INC	Rn	1	1
INC	DPTR	1	1
INC	direct	2	1
DEC	A	1	1
DEC	@Ri	1	1
DEC	Rn	1	1
DEC	direct	2	1
MUL	AB	1	4
DIV	AB	1	4
DA	A	1	1

Logical Operations

Mnemonic		Byte	Cyc
ANL	A,@Ri	1	1
ANL	A,Rn	1	1
ANL	A,direct	2	1
ANL	A,#data	2	1
ANL	direct,A	2	1
ANL	direct,#data	3	2
ORL	A,@Ri	1	1
ORL	A,Rn	1	1
ORL	A,direct	2	1

Mnemonic		Byte	Cyc
ORL	A,#data	2	1
ORL	direct,A	2	1
ORL	direct,#data	3	2
XRL	A,@Ri	1	1
XRL	A,Rn	1	1
XRL	A,direct	2	1
XRL	A,#data	2	1
XRL	direct,A	2	1
XRL	direct,#data	3	2
CLR	A	1	1
CPL	A	1	1
RL	A	1	1
RLC	A	1	1
RR	A	1	1
RRC	A	1	1
SWAP	A	1	1

Data Transfer

Mnemonic		Byte	Cyc
MOV	A,@Ri	1	1
MOV	A,Rn	1	1
MOV	A,direct	2	1
MOV	A,#data	2	1
MOV	@Ri,A	1	1
MOV	@Ri,direct	2	2
MOV	@Ri,#data	2	1
MOV	Rn,A	1	1
MOV	Rn,direct	2	2
MOV	Rn,#data	2	1
MOV	direct,A	2	1
MOV	direct,@Ri	2	2

Mnemonic		Byte	Cyc
MOV	direct,Rn	2	2
MOV	direct,direct	3	2
MOV	direct,#data	3	2
MOV	DPTR,#data16	3	2
MOVC	A,@A+DPTR	1	2
MOVC	A,@A+PC	1	2
MOVX	A,@Ri	1	2
MOVX	A,@DPTR	1	2
MOVX	@Ri,A	1	2
MOVX	@DPTR,A	1	2
PUSH	direct	2	2
POP	direct	2	2
XCH	A,@Ri	1	1
XCH	A,Rn	1	1
XCH	A,direct	2	1
XCHD	A,@Ri	1	1

Boolean Variable Manipulation

Mnemonic		Byte	Cyc
CLR	C	1	1
SETB	C	1	1
CPL	C	1	1
CLR	bit	2	1
SETB	bit	2	1
CPL	bit	2	1
ANL	C,bit	2	2
ANL	C,/bit	2	2
ORL	C,bit	2	2
ORL	C,/bit	2	2
MOV	C,bit	2	1
MOV	bit,C	2	2

Program and Machine Control

Mnemonic		byte	Cyc
NOP	1	1	
RET	1	2	
RETI	1	2	
ACALL	addr11	2	2
AJMP	addr11	2	2
LCALL	addr16	3	2
LJMP	addr16	3	2
SJMP	rel	2	2
JMP	@A+DPTR	1	2
DJNZ	Rn,rel	2	2
DJNZ	direct,rel	3	2
JZ	rel	2	2
JNZ	rel	2	2
JC	rel	2	2
JNC	rel	2	2
JB	bit,rel	3	2
JNB	bit,rel	3	2
JBC	bit,rel	3	2
CJNE	A,direct,rel	3	2
CJNE	A,#data,rel	3	2
CJNE	@Ri,#data,rel	3	2
CJNE	Rn,#data,rel	3	2

Special Function Register

Register Symbol	(MSB) b7	b6	b5	b4	b3	b2	b1	(LSB) b0	Byte Address
P0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	80H(128)
SP									81H(129)
DPL									82H(130)
DPH									83H(131)
PCON	SMOD	----	----	----	GF1	GF0	PD	IDL	87H(135)
*PCON	SMOD	----	----	WLE	GF1	GF0	PD	IDL	87H(135)
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88H(136)
TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0	89H(137)
TL0									8AH(138)
TL1									8BH(139)
TH0									8CH(140)
TH1									8DH(141)
P1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	T2EX	T2	90H(144)
*P1	SDA	SCL	RT2	T2	CT3I	CT2I	CT1I	CT0I	90H(144)
SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	98H(152)
SBUF									99H(153)
P2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	0A0H(208)
IE	EA	-——	ET2	ES	ET1	EX1	ET0	EX0	0A8H(168)
*IEN0	EA	EAD	ES1	ES0	ET1	EX1	ET0	EX0	0A8H(168)
+CML0									0A9H(169)
+CML1									0AAH(170)
+CML2									0ABH(171)
+CTL0									0ACH(172)
+CTL1									0ADH(173)
+CTL2									0AEH(174)
+CTL3									0AFH(175)
P3	RD	WR	T1	T0	INT1	INT0	TXD	RXD	0B0H(176)

Register Symbol	**(MSB) b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**(LSB) b0**	**Byte Address**
IP	--—	--—	PT2	PS	PT1	PX1	PT0	PX0	0B8H(184)
*IP0	--—	PAD	PS1	PS0	PT1	PX1	PT0	PX0	0B8H(184)
+P4	CMT1	CMT0	CMSR5	CMSR4	CMSR3	CMSR2	CMSR1	CMSR0	0C0H(192)
+P5	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	0C4H(196)
+ADCON	ADC.1	ADC.0	ADEX	ADCI	ADCS	AADR2	AADR1	AADR0	0C5H(197)
+ADCH									0C6H(198)
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2	0C8H(200)
*TM2IR	T2OV	CMI2	CMI1	CMI0	CTI3	CTI2	CTI1	CTI0	0C8H(200)
+CMH0									0C9H(201)
RCAP2L									0CAH(202)
*CMH1									0CAH(202)
RCAP2H									0CBH(203)
*CMH2									0CBH(203)
TL2									0CCH(204)
*CTH0									0CCH(204)
TH2									0CDH(205)
*CTH1									0CDH(205)
+CTH2									0CEH(206)
+CTH3									0CFH(207)
PSW	CY	AC	F0	RS1	RS0	OV	F1	P	0D0H(208)
+1S1CON		ENS1	STA	STO	SI	AA	CR1	CR0	0D8H(216)
+S1STA	SC4	SC3	SC2	SC1	SC0	0	0	0	0D9H(217)
+S1DAT									0DAH(218)
+S1ADR 0DBH (219)	--- --- --- Slave Address --- --- ---								GC

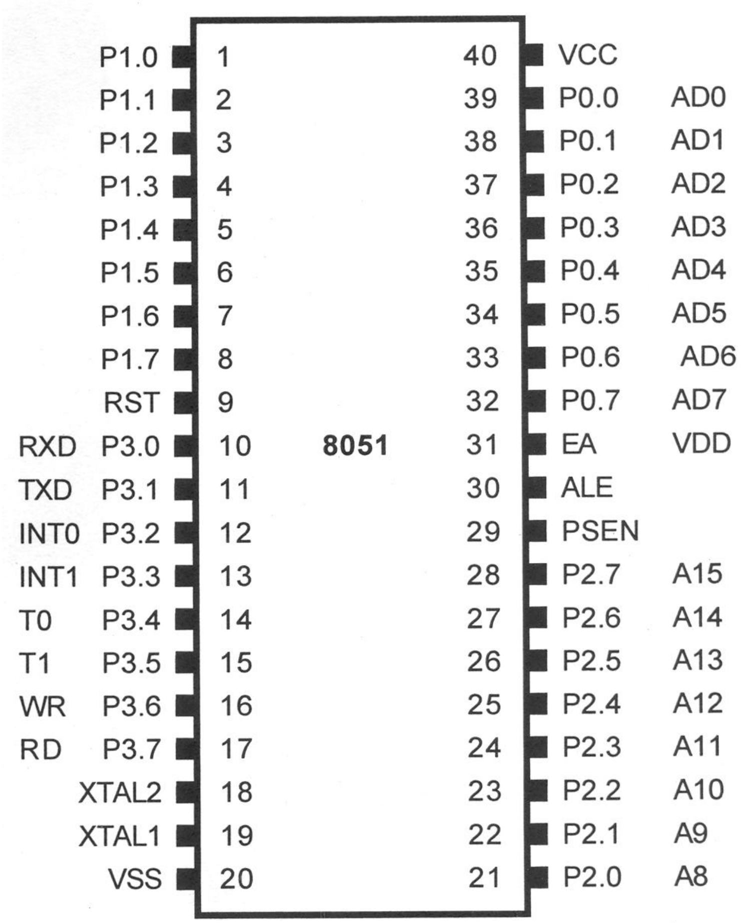
Register Symbol	(MSB) b7	b6	b5	b4	b3	b2	b1	(LSB) b0	Byte Address
ACC	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	0E0H(224)
+IEN1	ET2	ECM2	ECM1	ECM0	ECT3	ECT2	ECT1	ECT0	0E8H(232)
+TM2CON	T2IS1	T2IS0	T2ER	T2B0	T2P1	T2P0	T2MS1	T2MS0	0EAH(234)
+CTCON	CTN3	CTP3	CTN2	CTP2	CTN1	CTP1	CTN0	CTP0	0EBH(235)
+TML2									0ECH(236)
+TMH2									0EDH(237)
+STE	TG47	TG46	SP45	SP44	SP43	SP42	SP41	SP40	0EEH(238)
+RTE	TP47	TP46	RP45	RP44	RP43	RP42	RP41	RP40	0EFH(239)
B	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	0F0H(240)
+IP1	PT2	PCM2	PCM1	PCM0	PCT3	PCT2	PCT1	PCT0	0F8H(248)
+PWM0									0FCH(252)
+PWM1									0FDH(253)
+PWMP									0FEH(254)
+T3									0FFH(255)

Notes :

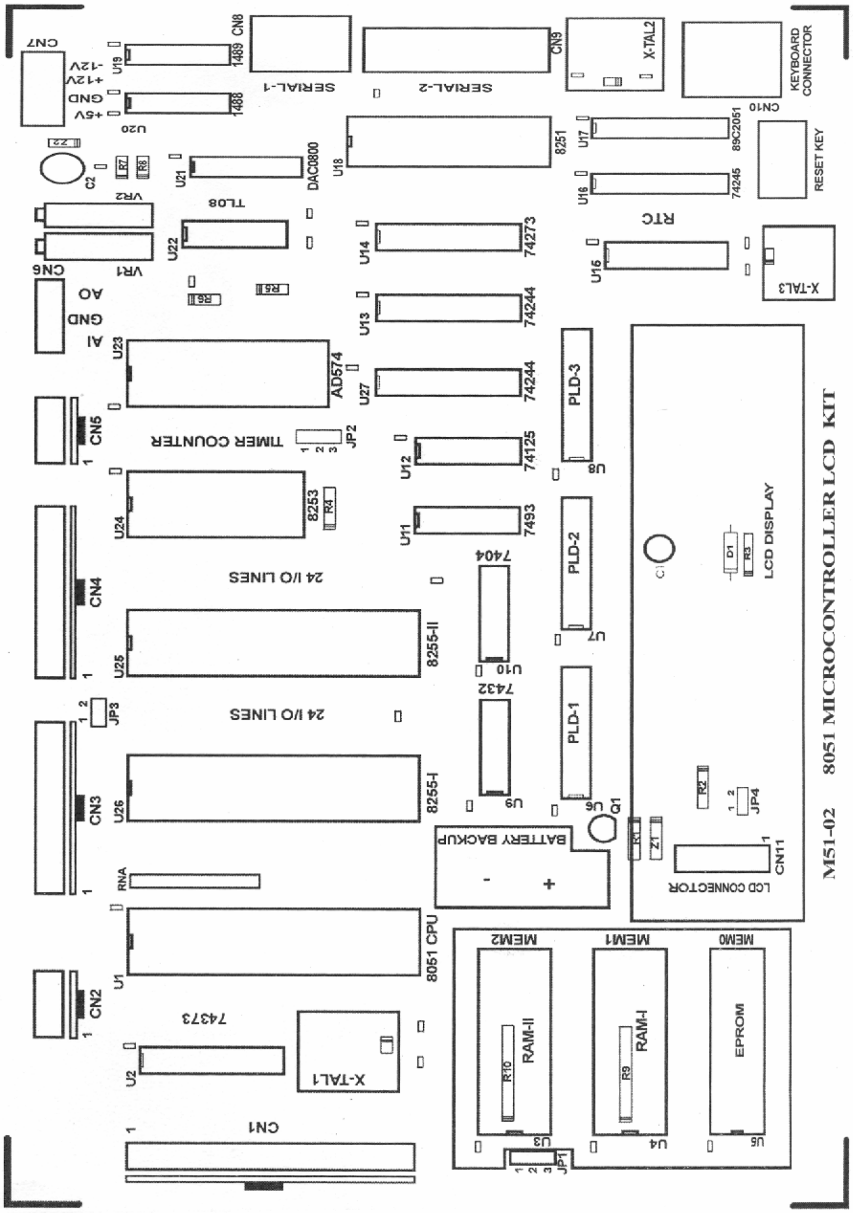
1. denotes the difference between 80C552 and 8051

2. + denotes the addition of 80C552

8051 Pin Detail



8051 LCD Kit Layout



Warranty

1. We guarantee the product against all manufacturing defects for 12 months from the date of sale by us or through our dealers. Consumables like dry cell etc. are not covered under warranty.

2. The guarantee will become void, if

a) The product is not operated as per the instruction given in the operating manual.

b) The agreed payment terms and other conditions of sale are not followed.

c) The customer resells the instrument to another party.

d) Any attempt is made to service and modify the instrument.

3 The non-working of the product is to be communicated to us immediately giving full details of the complaints and defects noticed specifically mentioning the type, serial number of the product and date of purchase etc.

4 The repair work will be carried out, provided the product is dispatched securely packed and insured. The transportation charges shall be borne by the customer.