

PERCOBAAN 4

Tujuan :

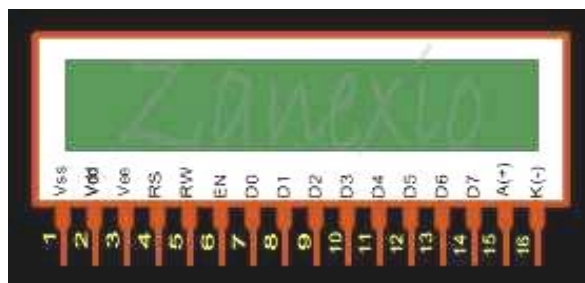
1. Mahasiswa memahami Penggunaan LCD ke Mikrokontroler AT89S51/52

Teori Dasar :

Antarmuka LCD

1. Tampilan LCD

LCD (Liquid Crystal Display) merupakan suatu alat yang dapat menampilkan karakter ASCII sehingga dapat menampilkan campuran huruf dan angka sekaligus. LCD diproduksi khusus untuk digunakan dengan mikrokontroler, yang berarti LCD tidak dapat diaktifkan atau digunakan dengan standar sirkuit IC. LCD digunakan untuk menampilkan pesan yang berbeda dalam sebuah miniatur.



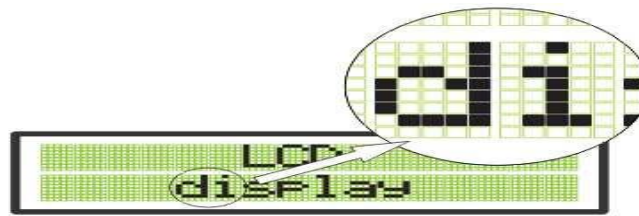
Model yang dijelaskan kali ini didasarkan pada harganya yang murah dan kapabilitasnya yang hebat yang paling sering digunakan dalam aplikasi. Didasarkan dari mikrokontroler HD44780 (Hitachi) dan dapat menampilkan pesan dalam dua baris dengan 16 karakter pada tiap barisnya. LCD ini dapat menampilkan semua huruf dalam alfabet, huruf Yunani, tanda baca, simbol – simbol matematika, dll. Dalam penggunaannya, LCD ini dapat memungkinkan penampilan simbol yang diciptakan oleh pengguna. Fitur yang berguna lainnya termasuk penggeseran pesan secara otomatis (kiri dan kanan), penampilan kursor, lampu latar LED, dll.

2. Pin – pin LCD

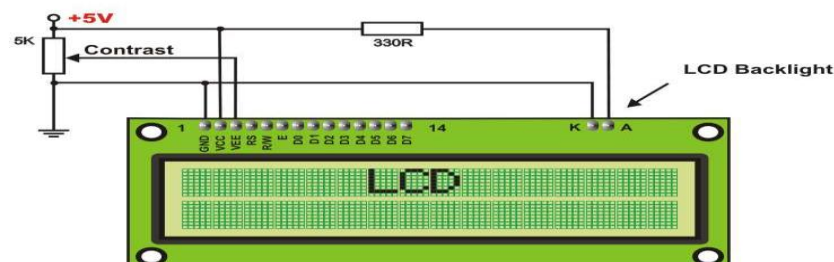
Dalam sepanjang satu sisi PCB LCD yang kecil terdapat pin – pin. Pin – pin ini digunakan untuk menghubungkan ke mikrokontroler. Total terdapat 14 pin disimbolkan dengan angka (16 pin jika terdapat lampu latar). Fungsi dari pin – pin ini dijelaskan pada table di bawah ini:

Fungsi	Nomor Pin	Nama	Posisi Logik	Penjelasan
Ground	1	Vss	-	0V
Power Supply	2	Vdd	-	+5V
Contrast	3	Vee	-	0 – Vdd
Control of Operating	4	RS	0	D0 – D7 diinterpretasikan sebagai perintah
			1	D0 – D7 diinterpretasikan sebagai data
	5	R/W	0	Menulis data (dari kontroler ke LCD)
			1	Membaca data (dari LCD ke kontroler)
	6	E	0	Akses ke LCD tidak diaktifkan
			1	Operasi normal
			Dari 1 ke 0	Data/Perintah ditransfer ke LCD
Data / Command	7	D0	0/1	Bit 0 LSB
	8	D1	0/1	Bit 1
	9	D2	0/1	Bit 2
	10	D3	0/1	Bit 3
	11	D4	0/1	Bit 4
	12	D5	0/1	Bit 5
	13	D6	0/1	Bit 6
	14	D7	0/1	Bit 7 MSB
Backlight VCC (5V)	15	Led +	-	Backlight VCC (5V)
Backlight GND (0V)	16	Led -	-	Backlight GND (0V)

3. Layar LCD



Sebuah layar LCD terdiri dari 2 baris, tiap baris terdiri dari 16 karakter. Setiap karakter terdiri dari 5x8 atau 5x11 dot matriks. Percobaan kali ini menggunakan display yang biasa digunakan, sebagai contoh 5x8 display karakter. Kontras display tergantung dari voltase power supply dan pesan keduanya ditampilkan dalam 1 atau 2 baris. Untuk alasan ini, berbagai jumlah voltase dari 0 – Vdd dihubungkan dengan pin yang bersimbol Vee. Potensio trimmer biasa digunakan untuk tujuan ini. Beberapa tampilan LCD mempunyai lampu latar terpasang tetap (LED biru atau hijau). Ketika digunakan selama operasi, resistor pembatas arus harus diserikan ke satu dari beberapa pin dari power supply lampu latar (serupa dengan LED).



4. LCD Memory

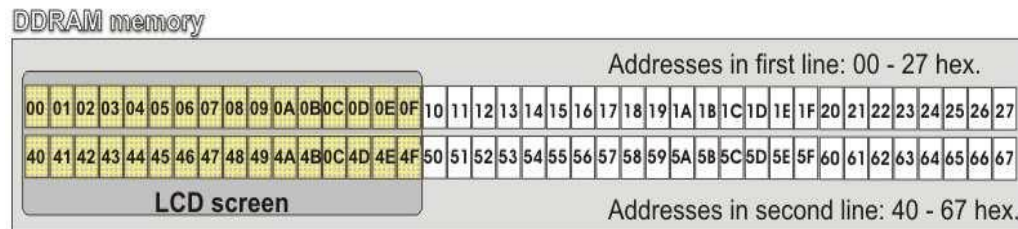
Tampilan LCD terdiri dari 3 blok memori :

- DDRAM Display Data RAM;
- CGRAM Character Generator RAM; and
- CGROM Character Generator ROM.

5. Memory DDRAM

Memori DDRAM digunakan untuk menyimpan karakter untuk selanjutnya ditampilkan. Besar dari memori ini cukup untuk menampung 80 karakter. Beberapa

lokasi memori terhubung langsung ke karakter pada tampilan.



Memori ini berkerja secara sederhana; Memori ini cukup untuk mengkonfigurasi tampilan sehingga dapat menaikkan alamat secara otomatis (geser kanan) dan men-set alamat awal untuk pesan yang harus ditampilkan (contoh 00 hex).

Setelah itu, semua karakter kemudian yang terkirim melalui jalur D0 – D7 akan ditampilkan dalam format pesan yang kita gunakan dari kiri ke kanan. Dalam kasus ini, penampilan dimulai dari bidang pertama dari baris pertama sejak alamat yang digunakan adalah 00 hex. Jika terdapat lebih dari 16 karakter yang terkirim, lalu semua pesan tersebut akan diingat, tapi hanya 16 karakter awal yang dapat terlihat. Agar dapat menampilkan karakter sisanya, sebuah perintah geser harus digunakan. Secara virtual, semua nampak layaknya tampilan LCD adalah sebuah “jendela” yang bergerak dari kiri ke kanan melalui lokasi memori yang berisi karakter yang berbeda. Inilah bagaimana cara kerja dari pesan yang “bergerak” dalam layar diciptakan.

Jika kursor dinyalakan, kursor tersebut muncul di lokasi dimana saat ini ditujukan. Dengan kata lain, saat karakter muncul pada posisi kursor, kursor akan secara otomatis pindah ke alamat berikutnya. Sejak memori ini semacam memori RAM, data dapat ditulis ke dan dibaca dari memori tersebut, namun isi memori tersebut hilang tidak dapat dipulihkan kembali saat power dimatikan.

6. Memori CGROM

Memori CGROM terdiri dari peta karakter dasar dengan semua karakter yang dapat ditampilkan di layar. Setiap karakter ditempatkan ke satu lokasi memori.

		4 higher bits in address															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
4 lower bits in address	xxxx0000	CG RAM (1)		0	@	P	`	P				-	9	≡	α	p	
	xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	△	ä	q
	xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	×	β	θ
	xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
	xxxx0100	(5)		\$	4	D	T	d	t			、	エ	ト	ト	μ	Ω
	xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	1	℃	Ü
	xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
	xxxx0111	(8)		'	7	G	W	g	w			ア	キ	ヌ	ラ	g	π
	xxxx1000	(1)		(8	H	X	h	x			ィ	ク	ネ	リ	フ	×
	xxxx1001	(2))	9	I	Y	i	y			ッ	ケ	ル	ル	´	4
	xxxx1010	(3)		*	:	J	Z	j	z			エ	コ	ハ	レ	j	〒
	xxxx1011	(4)		+	;	K	[k	{			オ	サ	ヒ	ロ	×	π
	xxxx1100	(5)		,	<	L	¥	l	l			ハ	シ	フ	ワ	¢	円
	xxxx1101	(6)		-	=	M]	m	}			ユ	ズ	ハ	ン	£	÷
	xxxx1110	(7)		.	>	N	^	n	÷			ヨ	セ	ホ	°	ñ	
	xxxx1111	(8)		/	?	O	_	o	€			ッ	リ	マ	°	ö	■

Alamat dari lokasi memori CGROM sesuai dengan karakter ASCII. Jika program saat ini sedang dieksekusi menemui perintah “kirim karakter P ke port”, maka seketika itu biner dengan nilai 0101 0000 muncul pada port. Nilai ini setara dengan kode ASCII karakter P. Hal ini kemudian dituliskan ke LCD, dimana hasilnya adalah menampilkan simbol dari lokasi 0101 0000 CGROM. Dengan kata lain, karakter “P”

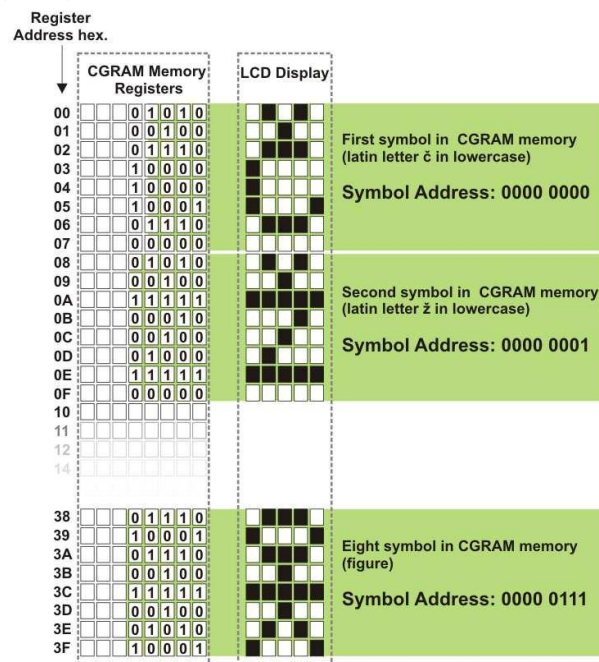
ditampilkan.

Proses ini sama untuk semua karakter dalam alfabet (kapital dan kecil), tapi tidak dengan angka. Seperti terlihat pada “peta” sebelumnya, alamat – alamat dari semua digit terdorong ke depan ke 48 tergantung dari nilainya (digit 0 dengan alamat 48, digit 1 dengan alamat 49, digit 2 dengan alamat 50, dst.). Karena itu, untuk menampilkan digit – digit secara benar, tiap digit harus ditambahkan sebelumnya dengan sebuah angka desimal 48 untuk dikirim ke LCD.

7. Memori CGRAM

Terlepas dari standar karakter, tampilan LCD dapat menampilkan simbol yang telah ditetapkan oleh user sendiri. Simbol tersebut dapat berupa simbol dengan ukuran 5x8 pixel. Memori RAM yang disebut CGRAM dengan ukuran 64 byte yang mengaktifkannya.

Register memori dengan lebar 8 bit, namun hanya 5 bit ke bawah yang digunakan. Logik satu (1) dalam setiap register merepresentasikan sebuah titik redup, sementara 8 lokasi dikelompokkan bersama merepresentasikan 1 karakter. Hal ini lebih baik diilustrasikan dengan gambar dibawah:



Simbol biasanya didefinisikan pada awal program dengan secara sederhana menuliskan beberapa nol dan satu ke register memory CGRAM sehingga bit – bit tersebut membentuk bentuk yang diinginkan. Untuk menampilkan bit – bit tersebut

perlu sebelumnya menentukan alamat bit – bit tersebut. Lihatlah kolom pertama dari peta karakter CGROM. CGROM tidak terdiri alamat – alamat memori RAM, tetapi simbol yang sedang dibahas di sini. Dalam contohnya, “tampilan 0” berarti menampilkan “c”, “tampilan 1” berarti menampilkan “z” dan seterusnya.

8. Perintah Dasar LCD

Semua data yang ditransfer ke LCD melalui output D0 – D7 akan diinterpretasikan sebagai perintah atau data, dimana tergantung dari kondisi logik PIN RS:

RS = 1 – Bit – bit D0 – D7 berisi alamat dari karakter – karakter yang akan ditampilkan. Prosesor LCD mengalamatkan satu karakter dari peta karakter dan menampilkannya. Memori DDRAM menentukan lokasi dimana karakter akan ditampilkan. Alamat ini ditentukan sebelum karakter ditransfer atau alamat dari karakter yang sebelumnya ditransfer dinaikkan secara otomatis.

Perintah	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	Waktu Eksekusi
Clear display	0	0	0	0	0	0	0	0	0	1	1.64ms
Cursor home	0	0	0	0	0	0	0	0	1	x	1.64ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	40us
Display on/off control	0	0	0	0	0	0	1	D	U	B	40us
Cursor/display shift	0	0	0	0	0	1	D/C	R/L	x	x	40us
Function set	0	0	0	0	1	DL	N	F	x	x	40us
Set CGRAM address	0	0	0	1	Alamat CGRAM						40us
Set DDRAM address	0	0	1	Alamat DDRAM							40us
Read "BUSY" flag (BF)	0	1	BF	Alamat DDRAM							-

RS = 0 – Bit – bit D0 – D7 merupakan perintah – perintah yang menentukan mode display. Perintah – perintah tersebut dikenali oleh LCD diperlihatkan pada table dibawah :

Write to CGRAM or DDRAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	40us
--------------------------------	---	---	----	----	----	----	----	----	----	----	------

Read from CGRAM or DDRAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	40us
---	---	---	----	----	----	----	----	----	----	----	------

I/D	1 = Increment (by 1)	R/L	1 = Shift right
	0 = Decrement (by 1)		0 = Shift left
S	1 = Display shift on	DL	1 = 8-bit interface
	0 = Display shift off		0 = 4-bit interface
D	1 = Display on	N	1 = Display in two lines
	0 = Display off		0 = Display in one line
U	1 = Cursor on	F	1 = Character format 5x10
	0 = Cursor off		0 = Character format 5x7 dots
B	1 = Cursor blink on	D/C	1 = Display shift
	0 = Cursor blink off		0 = Cursor shift


Code	Command to LCD Instruction
1	Clear display screen
2	Return home
4	Decrement cursor (shift cursor to
6	Increment cursor (shift cursor to
5	Shift display right
7	Shift display left
8	Display off, cursor off
A	Display off, cursor on
C	Display on, cursor off
E	Display on, cursor blinking
F	Display on, cursor blinking
10	Shift cursor position to left
14	Shift cursor position to right
18	Shift the entire display to the left
1C	Shift the entire display to the right
80	Force cursor to beginning to 1st
C0	Force cursor to beginning to 2nd
38	2 lines and 5x7 matrix

Tabel. KODE PERINTAH LCD

[illegible]

[illegible]

Langkah – Langkah Percobaan LCD Modul 4.1 s/d Modul 4.2

1. Bukalah **M-IDE 51** yang ada di desktop kemudian klik **New** untuk membuat lembaran kerja baru.
2. Ketiklah Program dan setelah itu **Save as** program tersebut dalam bentuk **.asm** ;
(Contoh : ketik : assembly.asm)
3. Setelah selesai, klik **F9** atau klik symbol seperti ini  (Running Program/Build Current File) untuk mengetahui program tersebut terjadi **error** atau **no error**.
4. Setelah program berhasil **tanpa eror** , Bukalah **TOP View Simulator** untuk melihat simulasi program tersebut.
5. Pilih **Select Device : ATMEL 89C51/52** (tidak perlu mengklik Data Memory maupun Program Memory) dengan **Operating Frequency : 12000000 Hz** Kemudian **OK**.
6. Klik File, kemudian arahkan kursor kearah menu **EXTERNAL MODULE SETTING**, Klik **Modul LCD**,pilih **select LCD : 2 Lines x 16 characters** dengan **Data Bus Width: 8 Bit** dan **Ceklist Back Light**.
7. Selanjutnya, Klik **PORT LINE SELECTION**, kemudian Konfigurasi **Control Line** dan **Port Line** tersebut. Setelah itu, klik **OK** dan **klik OK** lagi.
8. (Catatan : Jika didalam program menggunakan penekanan ataupun tampilan led,dll. Maka bukalah kembali **TOP View Simulator** dan **pilih modul ke 2** yang diinginkan. Jika hanya menggunakan 1 modul lewatkan langkah yang ke 8 ini).
9. Kemudian bukalah **View** dan arahkan kursor ke **External Module** kemudian pilih **Module LCD** Untuk menampilkan layar LCD Yang sudah dikonfigurasi tadi.(Jika menggunakan 2 modul maka setelah membuka menu **VIEW** untuk memilih 1 modul, bukalah kembali menu **VIEW** dan pilih modul ke 2 yang telah dikonfigurasi sebelumnya).
10. Setelah itu Klik **File** Kemudian **Load Program**, dan ambil data **.hex** setelah itu **Open** dan **OK**.
11. Kemudian arahkan kursor ke menu **Run** dan Klik **GO** untuk menjalankan Simulasi Tersebut dan Klik **Stop** untuk menghentikan simulasinya. **Tanyakan Kepada Asisten Jika Kurang Jelas !!**
P1.0-P1.7 terhubung dengan pin data D0-D7 di LCD
; P2.0 terhubung pada pin RS
; P2.1 terhubung pada pin R/W
; P2.2 terhubung pada pin E
} Konfigurasi LCD
Ke AT89S51/52
12. Catat hasil keluaran pada rangkaian LCD pada Simulator.

Alat-alat :

- 1 Set IBM PC dengan Sistem Operasi Windows dan Software M-IDE51, dan ISP Flash Programmer, dan TOP View Simulator.

Bagian 1

Modul 4.1 Program LCD Penggunaan Perintah EQU

PROGRAM	ANALISA PROGRAM
<pre>\$mod51 RS EQU P2.0 RW EQU P2.1 EN EQU P2.2 PIN EQU P0 org 00h ajmp start start: mov a,#38h acall perintah mov a,#0Ch acall perintah mov a,#06h acall perintah mov a,#01h acall perintah acall delay mov a,#86h acall perintah mov a,#'M' acall data1 acall delay</pre>	

<pre>mov a,#'P' acall data1 acall delay mov a,#'S' acall data1 acall delay mov a,#'1' acall data1 acall delay mov a,#0C7h acall perintah mov a,#'O' acall data1 acall delay mov a,#'K' acall data1 acall delay ajmp \$ perintah: mov PIN,a clr RW clr RS nop setb EN clr EN acall delay_lcd ret data1: mov PIN,a clr rw setb rs</pre>	
---	--

<pre>nop setb en clr en acall delay_lcd ret delay_lcd: mov r0,#0fh ulang:djnz r2,ulang djnz r1,ulang djnz r0,ulang ret delay: mov r0,#0AAh lagi:djnz r2,lagi djnz r1,lagi djnz r0,lagi ret end</pre>	
--	--

OUTPUT PADA LCD :

1. Carilah Nilai heksa untuk Tulisan Pertama “Lab” pada baris 1 dan “Mikroprosesor” pada baris 2 ?
Jawab :

.....
.....
.....
.....
.....
.....

BAGIAN 2

**Modul 4.2 Program LCD Untuk menampilkan beberapa karakter
(Pemberian nilai heksa yang ada pada masing – masing karakter
yang diinginkan)**

P1.0-P1.7 terhubung dengan pin data D0-D7 di LCD ; P2.0 terhubung pada pin RS
; P2.1 terhubung pada pin R/W
; P2.2 terhubung pada pin E

PROGRAM	ANALISA
<pre>ORG 0H MOV A,#38H ACALL COMMAND MOV A,#0EH ACALL COMMAND MOV A,#01H ACALL COMMAND MOV A,#06H ACALL COMMAND MOV A,#84H ACALL COMMAND MOV DPTR,#BACA MOV R5,#04H KATA1: CLR A MOVC A,@A+DPTR INC DPTR ACALL DATA_DISPLAY DJNZ R5,KATA1 MOV A,#0C4H ACALL COMMAND MOV DPTR,#LCD MOV R6,#03H KATA2: CLR A MOVC A,@A+DPTR INC DPTR ACALL DATA_DISPLAY DJNZ R6,KATA2 ANDA : SJMP ANDA</pre>	

<pre>COMMAND: ACALL MULAI MOV P1,A CLR P2.0 CLR P2.1 SETB P2.2 CLR P2.2 RET DATA_DISPLAY: ACALL MULAI MOV P1,A SETB P2.0 CLR P2.1 SETB P2.2 CLR P2.2 RET MULAI: SETB P1.7 CLR P2.0 SETB P2.1 BALIK: SETB P2.2 CLR P2.2 JB P1.7,BALIK RET BACA:DB 42H, 41H,43H,41H LCD : DB 4CH,43H,44H END</pre>	
---	--

Output Pada LCD :

1. Tambahkan program diatas pada modul 4.6 dan ubah beberapa program dan bahlah DB nya untuk menampilkan tulisan 'Lab' dan 'Mikroprosesor' dengan ketentuan sebagai berikut :
 - a. Jika ditekan P0.0 maka pada layar LCD akan tampil tulisan " Lab" Pada posisi baris 1 kolom 4.
 - b. Jika ditekan P0.1 maka pada layar LCD akan tampil tulisan " Mikroprosesor" pada posisi baris 2 kolom 2.

Program Anda :

[illegible]

Nama Assisten :		
BAGIAN 1	BAGIAN 2	MANDIRI
Tanggal Periksa :		

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

