

JOBSHEET V

ANTARMUKA MIKROKONTROLER DENGAN PENAMPIL LCD (Liquid Crystal Display)

TUJUAN

- Mengetahui dan memahami cara mengantarmukakan mikrokontroler dengan modul penampil LCD.
- Mengetahui dan memahami bagaimana memrogram mikrokontroler untuk menampilkan karakter ke penampil LCD.

LCD M1632

Modul LCD M1632 seperti dalam Gambar 5.1 merupakan modul LCD dengan tampilan 16 x 2 baris dengan konsumsi daya yang rendah. Modul ini dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LCD. Mikrokontroler HD44780 buatan Hitachi yang berfungsi sebagai pengendali LCD ini mempunyai CGROM (Character Generator Read Only Memory), CGRAM (Character Generator Random Access Memory), dan DDRAM (Display Data Random Access Memory).



Gambar 5.1 Modul LCD M1632

LCD ini memiliki 16 kaki, sebagaimana ditunjukkan dalam Tabel 6.1.

Tabel 5.1 Konfigurasi Pin LCD M1632

No	Kaki	Deskripsi
1	Vss	0V (GND)
2	Vcc	+5V
3	VLC	LCD Contrast Voltage
4	RS	Register Select; H: Data Input; L: Instruction Input
5	RD	H:Read; L: Write
6	EN	Enable Signal
7	D0	Data Bus 0
8	D1	Data Bus 1
9	D2	Data Bus 2
10	D3	Data Bus 3
11	D4	Data Bus 4
12	D5	Data Bus 5
13	D6	Data Bus 6
14	D7	Data Bus 7
15	V+BL	Positive Backlight Voltage
16	V-BL	Negative Backlight Voltage

DDRAM

DDRAM adalah merupakan memori tempat karakter yang ditampilkan berada. Contoh, untuk karakter 'A' atau 41H yang ditulis pada alamat 00, maka karakter tersebut akan tampil pada baris pertama dan kolom pertama dari LCD. Apabila karakter tersebut ditulis di alamat 40, maka karakter tersebut akan tampil pada baris kedua kolom pertama dari LCD. Posisi ini ditunjukkan dalam Gambar 5.2.

Posisi	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Addr	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

Gambar 5.2 Posisi DDRAM

CGRAM

CGRAM adalah merupakan memori untuk menggambarkan pola sebuah karakter di mana bentuk dari karakter dapat diubah-ubah sesuai keinginan. Namun memori ini akan hilang saat power supply tidak aktif, sehingga pola karakter akan hilang.

CGROM

CGROM adalah merupakan memori untuk menggambarkan pola sebuah karakter di mana pola tersebut sudah ditentukan secara permanen dari HD44780 sehingga pengguna tidak dapat mengubah lagi. Namun karena ROM bersifat permanen, maka pola karakter tersebut tidak akan hilang walaupun power supply tidak aktif. Pada Gambar 6.3, tampak terlihat pola-pola karakter yang tersimpan dalam lokasi-lokasi tertentu dalam CGROM. Pada saat HD44780 akan menampilkan data 41H ke DDRAM, maka HD44780 akan mengambil data di alamat 41H (0100 0001) yang ada pada CGROM yaitu pola karakter A.

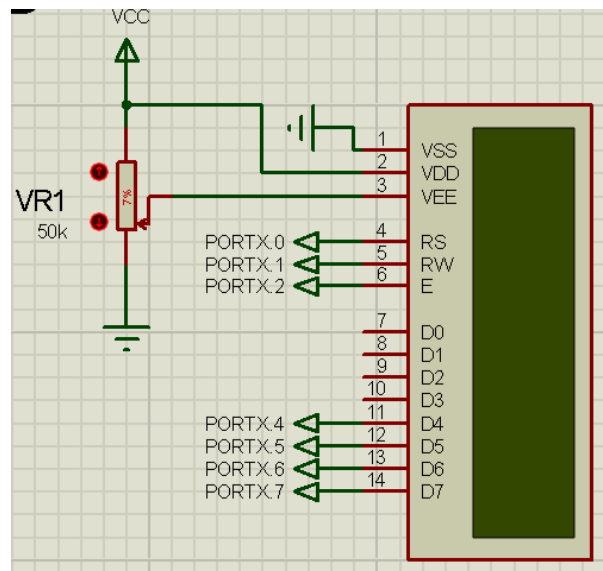
Dalam Gambar 5.4 modul LCD dihubungkan ke PORTB mikrokontroler dimana kaki RS, RD, EN terhubung ke PORTB.0-PORTB.2 dan D4, D5, D6, D7 terhubung ke PORTB.4-PORTB.7.

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Upper Case Code	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
xxxx0000	CG P001 (1)			0	a	P	`	P					-	9	3	α	p
xxxx0001	(2)	!	1	A	Q	a	q					.	7	4	ä	q	
xxxx0010	(3)	"	2	B	R	b	r					「	イ	ツ	×	β	θ
xxxx0011	(4)	#	3	C	S	c	s					」	ウ	テ	ε	ε	*
xxxx0100	(5)	\$	4	D	T	d	t					、	エ	ト	φ	μ	ω
xxxx0101	(6)	%	5	E	U	e	u					.	オ	ナ	1	σ	ü
xxxx0110	(7)	&	6	F	V	f	v					ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	'	7	G	W	g	w					フ	キ	ヲ	ラ	g	π
xxxx1000	(1)	(8	H	X	h	x					イ	ク	ネ	リ	J	×
xxxx1001	(2))	9	I	Y	i	y					ウ	ケ	ル	ル	Y	Y
xxxx1010	(3)	*	:	J	Z	j	z					エ	コ	ハ	レ	j	キ
xxxx1011	(4)	+	;	K	[k	[オ	サ	ヒ	ロ	*	π
xxxx1100	(5)	,	<	L	¥	l	¥					オ	シ	フ	ワ	φ	π
xxxx1101	(6)	-	=	M]	m)					ユ	ズ	ハ	ン	も	÷
xxxx1110	(7)	.	>	N	^	n	^					ヨ	セ	ホ	°	ñ	
xxxx1111	(8)	/	?	O	_	o	_					ツ	リ	マ	°	ö	■

Note: The user can specify any pattern for character-generator RAM.

Gambar 5.3 pola Karakter dalam CGROM



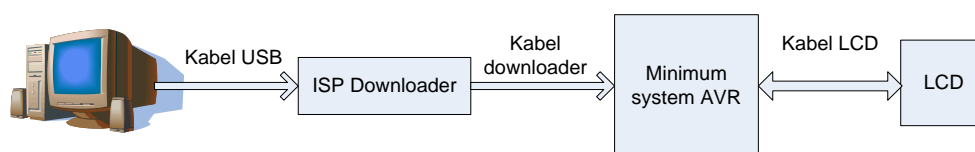
Gambar 5.4 Rangkaian LCD

ALAT DAN BAHAN YANG DIGUNAKAN

- 1 set PC/Laptop yang sudah berisi program Code Vision dan Khazama
- 1 buah catu daya DC +5V
- 1 buah multimeter
- 1 buah ISP Downloader AVR
- 1 buah sistem minimum AVR
- 1 buah LCD
- 1 buah kabel printer USB

PROSEDUR

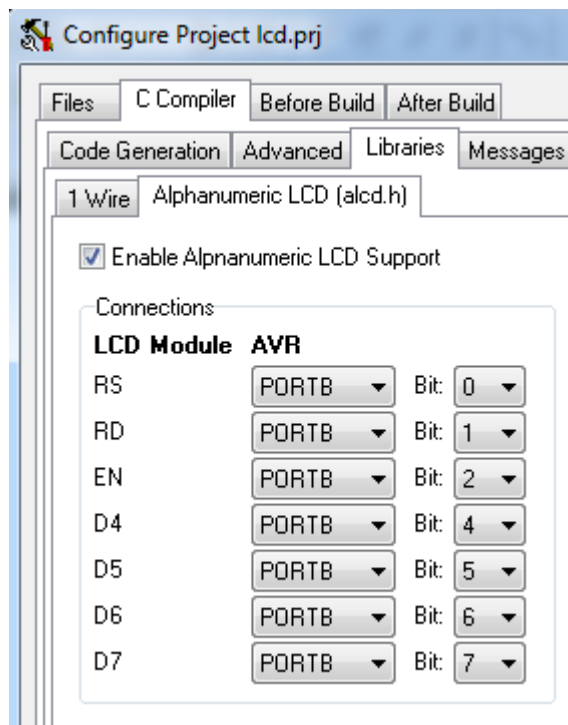
1. Rangkailah peralatan yang diperlukan seperti dalam Gambar 6.5. Hubungkan soket jumper PORTB pada minimum system dengan soket jumper pada LCD.



Gambar 5.5 Rangkaian antarmuka mikrokontroler dengan LCD

2. Buka program Code Vision AVR
3. Buatlah file project (.prj) kemudian pilih IC yang digunakan (ATmega8535) dan atur clock 4.000 Mhz. (seperti praktikum sebelumnya)

- Klik sub tab Libraries>>Alphanumeric LCD (alcd.h). Centang Enable Alphanumeric LCD Support kemudian sesuaikan pada gambar di bawah ini. Kemudian klik OK



- Buatlah file source (.c) kemudian hubungkan file project dengan file source seperti pada praktikum sebelumnya.
- Tuliskan file header


```
#include <mega8535.h>
#include <alcd.h>
#include <delay.h>
```
- Buatlah program utama dan tuliskan program berikut di dalam program utama .


```
// LCD module initialization
lcd_init(16);
```
- Tuliskan program berikut dalam program utama tepatnya di dalam while(1).


```
while (1)
{
    lcd_gotoxy(0,0); //menempatkan kursor di kolom 0 baris 0
    lcd_putchar(0x41); //menampilkan string 'A'
}
```
- Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khazama AVR Programmer. Klik auto program, amati tampilan LCD.

10. Hapus program yang ada di dalam while. Tuliskan program berikut di dalam while(1).

```
while (1)
{
    lcd_gotoxy(0,0); //menempatkan kursor di kolom 0 baris 0
    lcd_putchar(0x41);//menampilkan string 'A'
    delay_ms(1000);
    lcd_clear();
    lcd_gotoxy(0,1);
    lcd_putchar(0x30);
    delay_ms(1000);
}
```

11. Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khazama AVR Programmer. Klik auto program, amati tampilan LCD.

12. Hapus program yang ada di dalam while. Tuliskan program berikut di dalam while(1).

```
while (1)
{
    lcd_gotoxy(0,0);//menempatkan kursor di baris 0 kolom 0
    lcd_putsf("Saya belajar");//menampilkan string "Saya belajar"
}
```

13. Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khazama AVR Programmer. Klik auto program, amati tampilan LCD.

14. Hapus program yang ada di dalam while. Tuliskan program berikut di dalam while(1).

```
while (1)
{
    lcd_gotoxy(0,1); //menempatkan kursor di kolom 0 baris 1
    lcd_putsf("Antarmuka LCD");//menampilkan string "Antarmuka LCD"
}
```

15. Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khazama AVR Programmer. Klik auto program, amati tampilan LCD.

16. Hapus program yang ada di dalam while. Tuliskan program berikut di dalam while(1).

```
while (1)
{
    lcd_gotoxy(0,0); //menempatkan kursor di baris 0 kolom 0
    lcd_putsf("Saya belajar");//menampilkan string "Saya belajar"
    lcd_gotoxy(0,1);//menempatkan kursor di baris 1 kolom 0
    lcd_putsf("Antarmuka LCD");//menampilkan string "Antarmuka LCD"
}
```

17. Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khazama AVR Programmer. Klik auto program, amati tampilan LCD.

18. Buatlah tampilan nama anda berjalan dari kiri ke kanan pada baris 0 dan NIM anda berjalan dari kanan ke kiri pada baris 1 (tampilan di baris 0 dan 1 muncul secara bersamaan).

19. Ubahlah nilai dalam lcd_init(16) menjadi lcd_init(40) dan lihat perbedaannya.

DATA HASIL PERCOBAAN

1. Tampilan LCD pada langkah percobaan nomor 6
2. Tampilan LCD pada langkah percobaan nomor 7
3. Tampilan LCD pada langkah percobaan nomor 8
4. Tampilan LCD pada langkah percobaan nomor 9
5. Tampilan LCD pada langkah percobaan nomor 10
6. Source code program pada langkah percobaan nomor 11
7. Tampilan LCD pada langkah percobaan nomor 12

ANALISIS DATA

1. Analisis data hasil percobaan pada langkah percobaan nomor 6 sampai dengan nomor 10!
2. Jelaskan program anda pada langkah percobaan nomor 11!
3. Apakah fungsi sintaks `lcd_init()` dan jelaskan perbedaan antara menggunakan `lcd_init(16)` menjadi `lcd_init(40)`?