# PERCOBAAN 4 LCD KARAKTER

# **TUJUAN:**

- 1. Memahami rangkaian interface mikrokontroller dengan LCD Karakter 2 x 16
- 2. Dapat memahami program assembly untuk menampilkan data ke LCD Karakter 2 x 16
- 3. Memahami beberapa instruksi assembly dasar, MOV, Setb, Clr, dan waktu tunda.
- 4. Memahami mencetak karakter pada posisi baris dan kolom tertentu



Gambar 1. Rangkaian interface ke LCD Karakter 2 x16

Modul LCD Character dapat dengan mudah dihubungkan dengan mikrokontroller seperti AT89S52. LCD yang akan kita praktikumkan ini mempunyai lebar display 2 baris 16 kolom atau biasa disebut sebagai LCD Character 2x16, dengan 16 pin konektor, yang didifinisikan sebagai berikut:



Gambar 2. Modul LCD Karakter 2x16

Tabel 1.1 Pin dan Fungsi

PIN	Name	Function
1	Vss	Ground voltage
2	Vcc	+5V
3	VEE	Contrast voltage
4	RS	Register Select
		0 = Instruction Register
		1 = Data Register
5	R/W	Read/ Write, to choose write or read mode
		0 = write mode
		1 = read mode
6	E	Enable
		0 = start to lacht data to LCD character
		1= disable
7	DB0	LSB
8	DB1	-
9	DB2	-
10	DB3	-
11	DB4	-
12	DB5	-
13	DB6	-
14	DB7	MSB
15	BPL	Back Plane Light
16	GND	Ground voltage

Display karakter pada LCD diatur oleh pin EN,RS dan RW: jalur EN dinamakan Enable. Jalur ini digunakan untuk memberitahu LCD bahwa anda sedang mengirimkan sebuah data. Untuk mengirimkan data ke LCD, maka melalui program EN harus dibuat logika low "0" dan set pada dua jalur kontrol yang lain RS dan RW. Ketika dua jalur yang lain telah siap, set EN dengan logika "1" dan tunggu untuk sejumlah waktu tertentu ( sesuai dengan datasheet dari LCD tersebut ) dan berikutnya set EN ke logika low "0" lagi.

Jalur RS adalah jalur Register Select. Ketika RS berlogika low "0", data akan dianggap sebagai sebuah perintah atau instruksi khusus ( seperti clear screen, posisi kursor dll ). Ketika RS berlogika high "1", data yang dikirim adalah data text yang akan ditampilkan pada display LCD. Sebagai contoh, untuk menampilkan huruf "T" pada layar LCD maka RS harus diset logika high "1".

Jalur RW adalah jalur kontrol Read/ Write. Ketika RW berlogika low (0), maka informasi pada bus data akan dituliskan pada layar LCD. Ketika RW berlogika high "1", maka program akan melakukan pembacaan memori dari LCD. Sedangkan pada aplikasi umum pin RW selalu diberi logika low "0".

Pada akhirnya, bus data terdiri dari 4 atau 8 jalur ( bergantung pada mode operasi yang dipilih oleh user ). Pada kasus bus data 8 bit, jalur diacukan sebagai DB0 s/d DB7 Beberapa perintah dasar yang harus dipahami adalah inisialisasi LCD Character,

#### **Function Set**

Mengatur interface lebar data, jumlah dari baris dan ukuran font karakter

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	X	X

#### CATATAN:

X : Don't care **DL: Mengatur lebar data** DL=1, Lebar data interface 8 bit ( DB7 s/d DB0) DL=0, Lebar data interface 4 bit ( DB7 s/d DB4) Ketika menggunakan lebar data 4 bit, data harus dikirimkan dua kali **N: Pengaktifan baris** N=0, 1 baris N=1, 2 baris **F: Penentuan ukuran font karakter** F=0, 5x7 F=1, 5x8

# **Entry Mode Set**

Mengatur increment/ decrement dan mode geser

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

# Catatan:

**I/D:** Increment/ decrement dari alamat DDRAM dengan 1 ketika kode karakter dituliskan ke DDRAM.

I/D = "0", decrement

I/D= "1", increment

S: Geser keseluruhan display kekanan dan kekiri

S=1, geser kekiri atau kekanan bergantung pada I/D

S=0, display tidak bergeser

# **Display On/ Off Cursor**

Mengatur status display ON atau OFF, cursor ON/ OFF dan fungsi Cursor Blink

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	В

**D** : Mengatur display

D = 1, Display is ON

D = 0, Display is OFF

Pada kasus ini data display masih tetap berada di DDRAM, dan dapat ditampilkan kembali secara langsung dengan mengatur D=1.

C : Menampilkan kursor

C = 1, kursor ditampilkan

C = 0, kursor tidak ditampilkan

B : Karakter ditunjukkan dengan kursor yang berkedip

B=1, kursor blink

# **Clear Display**

Perintah ini hapus layar

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

# Geser Kursor dan Display

Geser posisi kursor atau display ke kanan atau kekiri tanpa menulis atau baca data display. Fungsi ini digunakan untuk koreksi atau pencarian display

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	X	X

#### Catatan : x = Dont care

S/C	R/L	Note
0	0	Shift cursor position to the left
0	1	Shift cursor position to the right
1	0	Shift the entire display to the left
1	1	Shift the entire display to the right

# POSISI KURSOR

Modul LCD terdiri dari sejumlah memory yang digunakan untuk display. Semua teks yang kita tuliskan ke modul LCD adalah disimpan didalam memory ini, dan modul LCD secara berturutan membaca memory ini untuk menampilkan teks ke modul LCD itu sendiri.

Display	00 01 02	03 04 05	06 07 08 09	10 11 12 13 14	15 16
Line 1	00 01 02	03 04 05	06 07 08 09	ØR ØB ØC ØD ØE	ØF 10 11 12 13 14 15
Line 2	40 41 42	43 44 45	46 47 48 49	4A 4B 4C 4D 4E	4F 50 51 52 53 54 55

Pada peta memori tersebut, daerah yang berwarna biru ( 00 s/d 0F dan 40 s/d 4F ) adalah display yang tampak. Sebagaimanan yang anda lihat, jumlahnya sebanyak 16 karakter per baris dengan dua baris. Angka pada setiap kotak adalah alamat memori yang bersesuaian dengan posisi dari layar. Demikianlah karakter pertama di sudut kiri atas adalah menempati alamah 00h. Posisi karakter berikutnya adalah alamat 01h dan seterusnya. Akan tetapi, karakter pertama dari baris 2 sebagaimana yang ditunjukkan pada peta memori adalah pada alamat 40h. Dimikianlah kita perlu untuk mengirim sebuah perintah ke LCD untuk mangatur letak posisi kursor pada baris dan kolom tertentu. Instruksi Set Posisi Kursor adalah 80h. Untuk ini kita perlu menambahkan alamat lokasi dimana kita berharap untuk menempatkan kursor.Sebagai contoh, kita ingin menampilkan kata "World" pada baris ke dua pada posisi kolom ke sepuluh. Sesuai peta memori, posisi karakter pada kolom 11 dari baris ke dua, mempunyai alamat 4Ah, sehingga sebelum kita tulis kata "World" pada LCD, kita harus mengirim instruksi set posisi kursor, dan perintah untuk instruksi ini adalah 80h ditambah dengan alamat 80h+4Ah =0CAh. Sehingga dengan mengirim perintah CAh ke LCD, akan menempatkan kursor pada baris kedua dan kolom ke 11 dari DDRAM.

Set Alamat Memori DDRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	A	A	A	A	A	A	A

#### Catatan:

A : Alamat RAM yang akan dipilih Sehingga alamat RAM LCD adalah 000 0000 S/D 111 1111 b atau 00 s/d 7Fh

#### INISIALISASI

Sebelum kita dapat menggunakan modul LCD, kita harus melakukan inisialisasi dan mengkonfigurasikannya. Hal ini dijalankan dengan mengirimkan sejumlah instruksi ke LCD. Antara lain: pengaturan lebar data interface 8 bit atau 4 bit data bus, pemilihan ukuran font karakter 5x8 atau 5x7 dan lain-lain, dengan instruksi sebagai berikut.

#### Init\_lcd:

```
Mov r1,#0000001b ;Display clear
call write_inst ;EN=pulse dan RS=0
mov r1,#00111000b ;Function set,Data 8 bit,2 line font 5x7
call write_inst ;Set bit EN=pulse dan RS=0
mov r1,#00001100b ;Display on,cursor off,cursor blink off
call write_inst ;EN=pulse dan RS=0
mov r1,#00000110b ;Entry mode, Set increment
call write_inst
ret
```

#### Percobaan 4.1.

# Tulis Sebuah Karakter pada LCD Karakter

Pada percobaan ini, karakter 'A' akan ditampilkan pada LCD Display

Untuk melakukan percobaan ini lakukan beberapa langkah sebagai berikut:

- 1. Pasang jumper LCD\_EN, yang berfungsi untuk memberikan power supply +5V
- 2. Hubungkan modul Microcontroller Trainer dengan power supply +5V
- 3. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
- 4. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
- 5. Ketik program berikut ini:

```
org Oh
        call init_LCD
 start: mov R1,#80h
                           ; Lokasi Display RAM, Row=1 Col=1
       call write_inst
       mov R1,#'A'
                           ; Cetak Karakter A
        call write_data
        stop: sjmp stop
Init_lcd:
       mov r1,#0000001b ;Display clear
        call write_inst
       mov r1,#00111000b ;Function set,Data8 bit,2 line font 5x7
        call write_inst
       mov r1,#00001100b ;Display on,cursor off,cursor blink off
        call write inst
       mov r1,#00000110b ;Entry mode, Set increment
        call write inst
       ret
Write inst:
        clr P3.6
                   ; P3.6 = RS =0
       mov P0,R1 ; P0 = D7 s/d D0 = R1
setb P3.7 ; P3.7 = EN = 1
        call delay
                        ; call delay time
        clr P3.7
                        ; P3.7 = EN = 0
        ret
        ;
Write_data:
        setb P3.6
                        ; P3.6 = RS =1
                        ; P0 = D7 s/d D0 = R1
       mov P0,R1
                        ; P3.7 =EN = 1
        setb P3.7
                      ; call delay time
        call delay
        clr p3.7
                        ; P3.7 = EN = 0
        ret
```

```
;
delay: mov R0,#0
delay1:mov R7,#0fh
djnz R7,$
djnz R0,delay1
ret
;
end
```

- 6. Simpanlah program yang anda ketik dan beri nama : prog41a.asm
- 7. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari \*.asm ke \*.hex.
- 8. Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software ( Lihat Petunjuk Penggunaan)
- 9. Lakukan pengamatan pada LED.
- 10. Lakukan modifikasi pada program tersebut untuk mencetak karakter lain, sesuai tabel:

Nama Karakter	Posisi Display
1	Row 1, Col 2
Z	Row 1, Col 8
&	Row 2, Col 12

#### Percobaan 4.2.

#### Tulis Delapan Karakter pada LCD Karakter

Pada percobaan ini, karakter 'A' akan ditampilkan pada LCD display

Untuk melakukan percobaan ini lakukan beberapa langkah sebagai berikut:

- 1. Pasang jumper LCD\_EN, yang berfungsi untuk memberikan power supply +5V
- 2. Hubungkan modul Microcontroller Trainer dengan power supply +5V
- 3. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
- 4. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
- 5. Ketik program berikut ini:

```
org Oh
       call init_LCD
                            ; Lokasi Display RAM, Row=1 Col=1
start: mov R1,#80h
       call write_inst
       mov R1,#'W'
       call write_data
       mov R1,#'e'
       call write_data
       mov R1,#'l'
       call write data
       mov R1,#'c'
       call write_data
       mov R1,#'o'
       call write_data
       mov R1,#'m'
       call write data
       mov R1,#'e'
       call write_data
       stop: sjmp stop
Init_lcd:
       mov r1,#0000001b ;Display clear
       call write inst
       mov r1,#00111000b ;Function set,Data 8 bit,2 line font 5x7
       call write_inst
       mov r1,#00001100b ;Display on, cursor off, cursor blink off
       call write_inst
       mov r1,#00000110b ;Entry mode, Set increment
       call write inst
       ret
       ;
Write inst:
       clr P3.6 ; P3.6 = RS =0
       mov P0,R1 ; P0 = D7 s/d D0 = R1
```

```
setb P3.7 ; P3.7 =EN = 1
     call delay ; call delay time
     clr P3.7; P3.7 = EN = 0
     ret
      ;
Write data:
     ret
      ;
delay: mov R0,#0
delay1:mov R7,#0fh
     djnz R7,$
     djnz R0,delay1
     ret
      ;
      end
```

- 6. Simpanlah program yang anda ketik dan beri nama : prog42a.asm
- Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari \*.asm ke \*.hex.
- Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software (Lihat Petunjuk Penggunaan)
- 9. Lakukan pengamatan pada LED.
- 10. Lakukan modifikasi pada program tersebut untuk mencetak karakter lain, sesuai tabel:

Nama Karakter	Start Posisi Display
Selamat Datang	Row 1, Col 1
Mikrokontroller	Row 2, Col 1

# Percobaan 4.3.

#### **Tulis Karakter Dengan Look Up Table**

Pada percobaan ini, karakter 'A' akan ditampilkan pada LCD Display

Untuk melakukan percobaan ini lakukan beberapa langkah sebagai berikut:

- 1. Pasang jumper LCD\_EN, yang berfungsi untuk memberikan power supply +5V
- 2. Hubungkan modul Microcontroller Trainer dengan power supply +5V
- 3. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
- 4. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
- 5. Ketik program berikut ini:

```
org Oh
       call init_LCD
start: call write_char
       stop: sjmp stop
write char:
       mov dptr,#word1
                            ;DPTR = [ address word1 ]
       mov r3,#16 ;R3=16,number character to be display
       mov r1,#80h ;R1=80h,address DDRAM start position
       call write_inst
       ;
write1:clr a ; A = 0
                           ; A = [A + DPTR]
       movc a,@a+dptr
                            ; R1 = A
       mov r1,A
       inc dptr
                            ; DPTR = DPTR +1
       inc apt:
call write_data;
_______; R3 = R3-1,
       ret
Init_lcd:
       mov r1,#0000001b ;Display clear
       call write_inst ;
       mov r1,#00111000b ;Function set,Data 8 bit,2 line font 5x7
       call write_inst ;
       mov r1,#00001100b ;Display on, cursor off, cursor blink off
       call write_inst
       mov r1,#00000110b ;Entry mode, Set increment
       call write_inst
       ret
Write_inst:
       clr P3.6
                             ; P3.6 = RS = 0
       mov P0,R1 ; P0 = D7 s/d D0 = R1
setb P3.7 ; P3.7 =EN = 1
       call delay ; call delay time
```

```
; P3.7 = EN = 0
          clr P3.7
          ret
          ;
Write_data:
         setb P3.6 ; P3.6 = RS =1
mov P0,R1 ; P0 = D7 s/d D0 = R1
setb P3.7 ; P3.7 =EN = 1
call delay ; call delay time
call delay ; call delay time
                                        ; P3.7 = EN = 0
          clr p3.7
          ret
delay: mov R0,#0
delay1:mov R7,#0fh
          djnz R7,$
          djnz R0,delay1
          ret
          ;
word1: DB ' Welcome Home '; Karakter yang disimpan di ROM
          ;
          end
```

- 6. Simpanlah program yang anda ketik dan beri nama : prog43a.asm
- 7. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari \*.asm ke \*.hex.
- 8. Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software (Lihat Petunjuk Penggunaan)
- 9. Lakukan pengamatan pada LCD.
- 10. Lakukan modifikasi pada program tersebut untuk mencetak karakter lain, sesuai tabel:

Nama Karakter	Start Posisi Display
Selamat Datang	Row 1, Col 1
Mikrokontroller	Row 2, Col 1

# Percobaan 4.4. Perintah Geser Satu Karakter

Pada percobaan ini, karakter "A" akan ditampilkan pada layar LCD Karakter dan dilakukan proses geser ke kanan.

Untuk melakukan percobaan ini lakukan beberapa langkah sebagai berikut:

- 1. Pasang jumper LCD\_EN, yang berfungsi untuk memberikan power supply +5V
- 2. Hubungkan modul Microcontroller Trainer dengan power supply +5V
- 3. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
- 4. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
- 5. Ketik program berikut ini:

geserL equ 00011011b ; 0 0 0 1 S/C R/L X X geserR equ 00011111b ; 0 0 0 1 S/C R/L X X org Oh call init\_LCD start: mov R1,#80h call write\_inst mov R1,#'A' ; Karakter yang akan digeser call write data ; mov R3,#40 ; 40x geser ke kanan Next: mov R1,#geserR call write\_inst call ldelay djnz R3,next sjmp start Init\_lcd: mov r1,#0000001b ;Display clear call write\_inst ; mov r1,#00111000b ;Function set, ;Data 8 bit,2 line font 5x7 call write\_inst ; mov r1,#00001100b ;Display on, ;cursor off,cursor blink off call write\_inst mov r1,#00000110b ;Entry mode, Set increment call write inst ret ; Write\_inst: clr P3.6 ; RS = 0, write mode instruction ; D7 s/d D0 = P0 = R1 mov P0,R1 setb P3.7 ; EN = 1call delay ; call delay time

	clr P3.7	;	; $EN = 0$
	ret		
	;		
Write_data:			
	setb P3.6	;	; RS = 1, write mode data
	mov P0,R1	;	; D7 s/d D0 = P0 = R1
	setb P3.7	;	; EN =1
	call delay	;	; call delay time
	clr p3.7	;	; $EN = 0$
	ret		
	;		
Ldelay:	mov R2,#030h		
Ld1:	call delay		
	djnz R2,Ld1		
	ret		
	;		
delay:	mov R0,#0		
delay1:	mov R7,#0fh		
	djnz R7,\$		
	djnz R0,delay1		
	ret		
	;		
	end		

# Percobaan 4.5. Perintah Geser Kalimat

Pada percobaan ini, kalimat "Welcome home" akan ditampilkan pada layar LCD Karakter dengan menggunakan look up table dan dilakukan proses geser ke kiri.Untuk melakukan percobaan ini lakukan beberapa langkah sebagai berikut:

1. Pasang jumper LCD\_EN, yang berfungsi untuk memberikan power supply +5V

- 2. Hubungkan modul Microcontroller Trainer dengan power supply +5V
- 3. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
- 4. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
- 5. Ketik program berikut ini:

```
geserL equ 00011011b
                          ; 0 0 0 1 S/C R/L X X
geserR equ 00011111b
                           ; 0 0 0 1 S/C R/L X X
           org Oh
           call init_LCD
start: call write_char
           call geser
           sjmp start
           ;
geser:
mov R6,#40 ;R7 = 40; geser sebanyak 40 karakter
lagi: mov R1,#geserL ;R1 = GeserLeft
           call write_inst
           call ldelay
           djnz R6,lagi ;R7=R7-1,jika R7?0 maka lompat ke lagi
write_char:
           mov dptr,#word1 ;DPTR = [ address word1 ]
           mov r3,#16
                          ;R3=16,number character to be display
           mov r1,#80h ;R1=80h,address DDRAM start position
           call write_inst
write1:clr a ; A = 0
           movc a,@a+dptr ; A = [A+ DPTR]
          mov r1,A ; R1 = A
           inc dptr
                           ; DPTR = DPTR +1
           call write_data;
           djnz r3, write1 ; R3 = R3-1,
           ret
Init lcd:
           mov r1,#0000001b ;Display clear
           call write_inst ;
           mov r1,#00111000b
                                  ;Function set,
                                   ;Data 8 bit,2 line font 5x7
```

```
call write_inst ;
           mov r1,#00001100b
                                   ;Display on,
                                     ;cursor off,cursor blink off
            call write_inst
           mov r1,#00000110b
                                   ;Entry mode, Set increment
            call write_inst
           ret
            ;
Write_inst:
            clr P3.6
                                   ; RS = P2.0 = 0, write mode
instruction
           mov P0,R1
setb P3.7
call delay
clr P3.7
                                   ; D7 s/d D0 = P0 = R1
                                    ; EN = 1 = P2.1
                                   ; call delay time
            clr P3.7
                                    ; EN = 0 = P2.1
           ret
            ;
           setb P3.6
mov P0,R1
setb P3.7
call delay
Write_data:
                                   ; RS = P2.0 = 1, write mode data
                                   ; D7 s/d D0 = P0 = R1
                                  ; EN = 1 = P2.1
                                  ; call delay time
                                  ; EN = 0 = P2.1
           ret
Ldelay: mov R2,#030h
Ld1: acall delay
           djnz R2,Ld1
           ret
delay: mov R0,#0
delay1: mov R7,#0fh
           djnz R7,$
           djnz R0,delay1
           ret
;
word1: DB ' Welcome Home ' ;here is the data to be look up
            ;
            end
```

- 6. Simpanlah program yang anda ketik dan beri nama : prog44a.asm
- 7. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari \*.asm ke \*.hex.
- 8. Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software