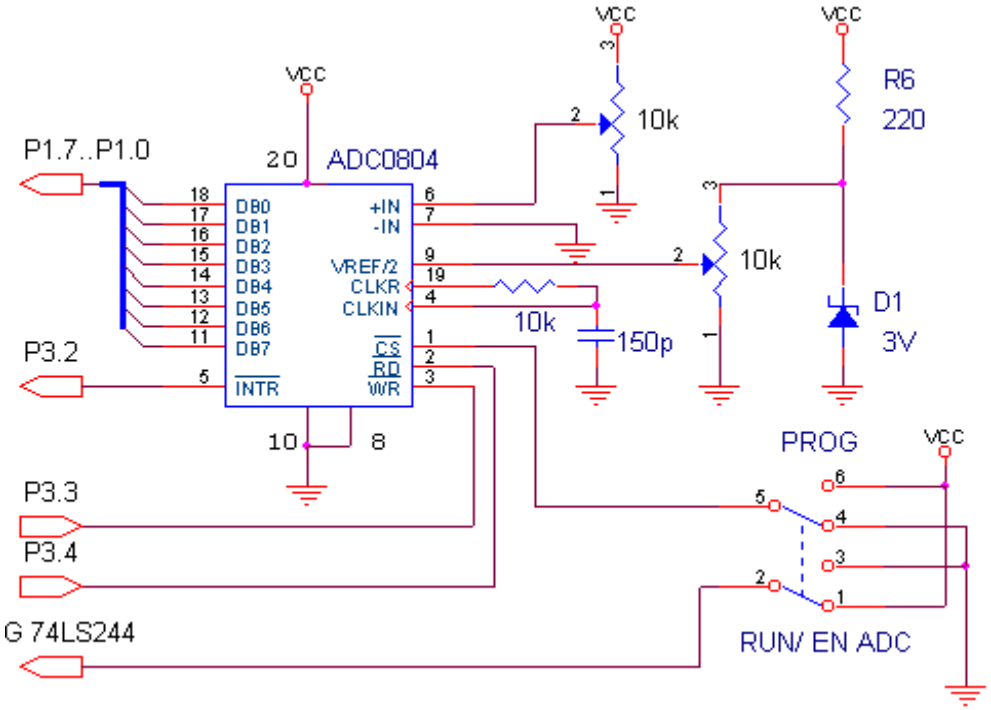
PERCOBAAN 5
ANALOG TO DIGITAL CONVERTER (ADC)

TUJUAN:

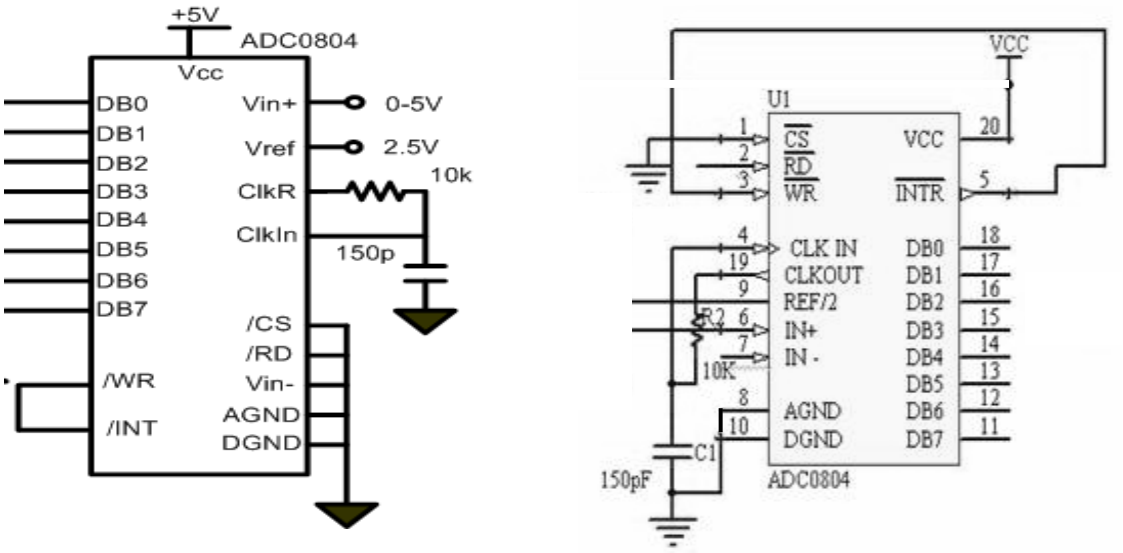
1. Memahami rangkaian interface mikrokontroller dengan ADC 0804
2. Memahami setting tegangan referensi Vref ADC0804
3. Memahami perhitungan tegangan resolusi ADC0804
4. Memahami program assembly untuk menampilkan data ADC ke 7 Segmen
5. Memahami program assembly untuk menampilkan data ADC ke LCD Karakter 2 x 16



Gambar 5.1 Rangkaian ADC0804

DASAR TEORI

Konverter A/D tersedia secara komersial sebagai rangkaian terpadu dengan resolusi 8 bit sampai dengan 16 bit. Pada percobaan ini akan memperkenalkan ADC0804, yaitu sebagai sebuah konverter A/D 8 bit yang mudah diinterfacekandengan sistem mikrokontroller. A/D ini menggunakan metode approksimasi berturut-turut untuk mengkonversikan masukan analog (0-5V) menjadi data digital 8 bit yang ekivalen. ADC0804 mempunyai pembangkit clock internal dan memerlukan catu daya +5V dan mempunyai waktu konversi optimum sekitar 100us.



Gambar 5.2 Konfigurasi pin ADC0804

Diagram konfigurasi pin ADC0804 ditunjukkan pada gambar 5.2. Pin 11 sampai 18 (keluaran digital) adalah keluaran tiga keadaan, yang dapat dihubungkan langsung dengan bus data bilamana diperlukan. Apabila CS (pin 1) atau RD (pin2) dalam keadaan high ("1"), pin 11 sampai 18 akan mengambang (high impedanze), apabila CS dan RD rendah keduanya, keluaran digital akan muncul pada saluran keluaran.
Sinyal mulai konversi pada WR (pin 3). Untuk memulai suatu konversi, CS harus rendah. Bilamana WR menjadi rendah, konverter akam mengalami reset, dan ketika WR kembali kepada keadaan high, konversi segera dimulai.
Konversi detak konverter harus terletak dalam daerah frekuensi 100 sampai 800kHz. CLK IN (pin 4) dapat diturunkan dari detak mikrokontroller, sebagai kemungkinan lain, kita dapat mempergunakan pembangkit clock internal dengan memasang rangkaian RC antara CLN IN (pin 4) dan CLK R (pin 19).
Pin 5 adalah saluran yang digunakan untuk INTR, sinyal selesai konversi. INTR akan menjadi tinggi pada saat memulai konversi, dan akan aktiv rendah bila konversi telah selesai. Tepi turun sinyal INTR dapat dipergunakan untuk menginterupsi sistem mikrokontroller, supaya mikrokontroller melakukan pencabangan ke subrutine pelayanan yang memproses keluaran konverter.
Pin 6 dan 7 adalah masukan diferensial bagi sinyal analog. A/D ini mempunyai dua ground, AGND (pin 8) dan DGND (pin10). Kedua pin ini harus dihubungkan dengan ground. Pin 20 harus dihubungkan dengan catu daya +5V
Pada A/D 0804 merupakan tegangan referensi yang digunakan untuk offset suatu keluaran digital maksimum. Dengan persamaan sebagai berikut:

$$V_{REF} = \frac{1}{2} V_{IN} maks$$

Misalnya anda menginginkan masuk analog maksimum sebesar 4 V, maka:

Vref=0.5 x 4 = 2 volt

$$V_{RESOLUSI} = \frac{V_{IN} MAKS}{255}$$

Resolusi ini mempunyai arti sebagai berikut:

Vin (volt)	Data Digital (biner)	Data Digital (desimal)
0,000	0000 0000	
0,0156	0000 0001	
0,0313	0000 0010	
4	1111 1111	255

A/D ini dapat dirangkai untuk menghasilkan konversi secara kontinu. Untuk melaksanakannya, kita harus menghubungkan CS, dan RD ke ground dan menyambungkan WR dengan INTR seperti pada gambar dibawah ini. Maka dengan ini keluaran digital yang kontinu akan muncul, karena sinyal INTR menggerakkan masukan WR. Pada akhir konversi INTR berubah menjadi low, sehingga keadaan ini akan mereset konverter dan mulai konversi.

Tabel 5.1 Koneksi Interface ADC ke Mikrokontroller

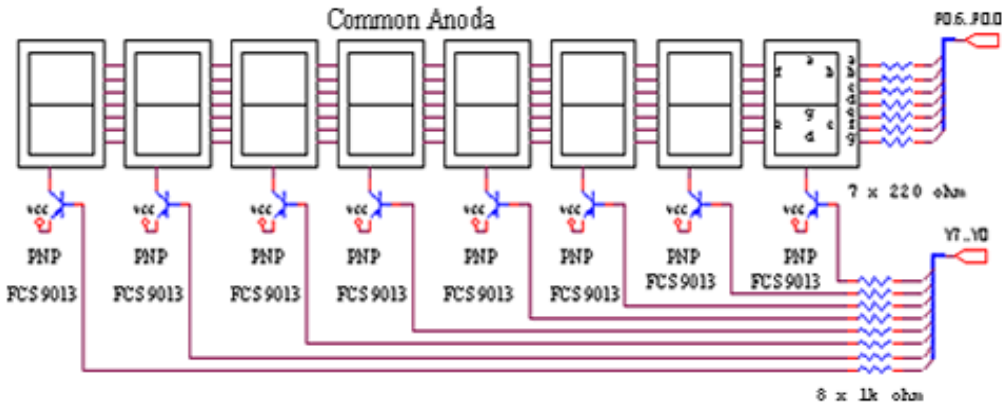
ADC	Port Mikrokontroller
/INTR	P3.2
/WR	P3.3
/RD	P3.4
D0 s/d D7	P1.0 s/d P1.7

Tabel 5.2. Instruksi logika pada pin kontrol A/D 0804

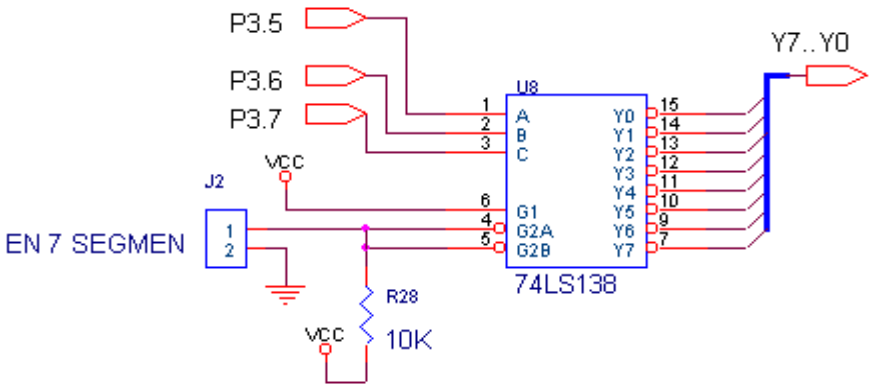
INPUT			**OUTPUT**	**KEGIATAN**
/WR	/RD	/INTR	DO S/D D7	Hi-Z (High Impedansi)
1	1	1	-	-
0	1	1	Hi-Z	Reset
1	1	1	Hi-Z	-
1	1	0	Hi-Z	Konversi Selesai
1	0	1	Data Out	Data Ready

Percobaan 5.1.
ADC0804 dan Display ke 7 Segmen

Pada percobaan ini, Data ADC dalam desimal akan ditampilkan pada 8 x 7 Segmen pada Display 1, Display 2, dan Display 3 yang masing-masing menampilkan data ratusan, puluhan dan satuan.



(a)



(b)

Gambar 5.3. Interface rangkaian display 7 segmen

Tabel 5.1. kebenaran 74LS138

INPUT SELECTOR			ENABLE			OUTPUT							
C	B	A	G1	/G2A	/G2B	Y1	Y2	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	1	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	0	1	0	1	1	1	1	1	1
0	1	0	1	0	0	1	1	0	1	1	1	1	1
0	1	1	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	0	0	1	1	1	1	1	0	1	1
1	1	0	1	0	0	1	1	1	1	1	1	0	1
1	1	1	1	0	0	1	1	1	1	1	1	1	0

Pada tabel kebenaran tersebut tampak bahwa seven segmen yang hidup tergantung pada output dari dekoder 74LS138, yang sedang mengeluarkan logika low ”0”, sehingga dari 8 buah display tersebut, selalu hanya satu display yang akan dihidupkan. Agar display tampak nyala secara bersamaan maka ketiga display tersebut harus dihidupkan secara bergantian dengan waktu tunda tertentu.

Untuk melakukan percobaan ini lakukan beberapa langkah sebagai berikut:

1. Pada saat langkah pemrograman posisikan saklar togle ke posisi PROG
2. Posisikan saklar togle ke RUN untuk mengaktifkan ADC0804 CS=0
3. Hubungkan modul Microcontroller Trainer dengan power supply +5V
4. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
5. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
6. Ketik program berikut ini:

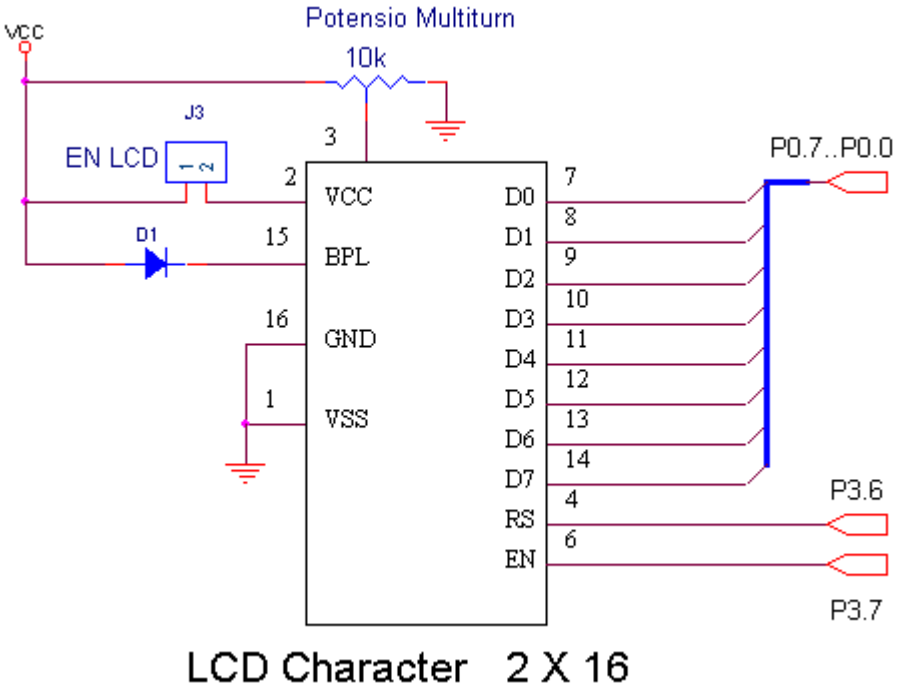
```
        org 0h
        ratusan equ 30h
        puluhan equ 31h
        satuan equ 32h
        ;
        org 0h
start:  call ADC
        call Bin2Dec
        call Display2SevenSegmen
        sjmp start
        ;
;================================================
;Subrutin ini digunakan untuk mengambil data ADC
;================================================
ADC:    clr P3.3
        nop
        nop
        nop
        setb P3.3
eoc:    jb P3.2,eoc
        clr P3.4
        mov A,P1
        setb P3.4
        ret
        ;
;================================================
;Subrutin ini untuk menampilkan data ke 7 Segmen
;dalam bentuk: ratusan, puluhan, and satuan
;data desimal diubah ke segmen dengan menggunakan
;Look up table Data7segmen
;================================================
Display2SevenSegmen:
        mov A, ratusan
        mov DPTR,#Data7segmen
        movc A,@A+DPTR
        mov P0,A
```

```
        Setb P3.5 ;
        clr P3.6
        Setb P3.7
        call delay
        ;
        mov A,puluhan
        mov DPTR,#Data7segmen
        movc A,@A+DPTR
        mov P0,A
        clr P3.5 ;
        Setb P3.6
        Setb P3.7
        call delay
        ;
        mov A,satuan
        mov DPTR,#Data7segmen
        movc A,@A+DPTR
        mov P0,A
        Setb P3.5 ;
        Setb P3.6
        Setb P3.7
        call delay
        ret
        ;
delay:  mov R0,#0
delay1:mov R2,#0fh
        djnz R2,$
        djnz R0,delay1
        ret
        ;
;================================================
;Subrutin ini untuk merubah data biner ke desimal
;menjadi 3 digit = ratusan-puluhan-satuan
;================================================
Bin2Dec:
        mov b,#100d
        div ab
        mov ratusan,a
        mov a,b
        mov b,#10d
        div ab
        mov puluhan,a
        mov satuan,b
        ret
        ;
Data7segmen:
        db 11000000b,11111001b,10100100b,10110000b,10011001b
        db 10010010b,10000010b,11111000b,10000000b,10010000b
        ;
        end
```

7. Simpanlah program yang anda ketik dan beri nama : prog51a.asm
8. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari *.asm ke *.hex.
9. Lakukan modifikasi pada program tersebut dengan manambahkan kata SUHU, pada Display1, 2, 3 dan 4 diikuti dengan data ADC.

Percobaan 5.2.
ADC0804 dan Display ke LCD Karakter 2x16

Pada percobaan ini, Data ADC dalam desimal akan ditampilkan pada LCD Karakter 2x16 pada Baris 1, Colom 1, 2 dan 3, yang masing-masing menampilkan data ratusan, puluhan dan satuan.



Untuk melakukan percobaan ini lakukan beberapa langkah sebagai berikut:

1. Pada saat langkah pemrograman posisikan saklar togle ke posisi PROG
2. Posisikan saklar togle ke RUN untuk mengaktifkan ADC0804 CS=0
3. Hubungkan modul Microcontroller Trainer dengan power supply +5V
4. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
5. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program
6. Ketik program berikut ini:

```
        ratusan equ 30h
        puluhan equ 31h
        satuan equ 32h
        ;
        org 0h
        call init_LCD
        call write_char
start:  call ADC
        call Bin2Dec
        call Write2LCD
        sjmp start
;================================================
;Subrutin ini digunakan untuk mengambil data ADC
;================================================
```

```
ADC:   clr P3.3
       nop
       nop
       nop
       setb P3.3
eoc:   jb P3.2,eoc
       clr P3.4
       mov A,P1
       cpl A
       mov P0,A
       setb P3.4
       ret
;=========================================================
;Subrutin untuk menampilkan data ke LCD character 2 x16
;pada DDRAM 0C9 0CA 0CB untukratusan, puluhan, and satuan
;=========================================================
Write2LCD:
       mov r1,#0c9h
       call write_inst
       mov a,ratusan
       add a,#30h
       mov r1,a
       call write_data
       ;
       mov r1,#0cah
       call write_inst
       mov a,puluhan
       add a,#30h
       mov r1,a
       call write_data
       ;
       mov r1,#0cbh
       call write_inst
       mov a,satuan
       add a,#30h
       mov r1,a
       call write_data
       ret
;================================================
;Subrutin ini untuk merubah data biner ke desimal
;menjadi 3 digit = ratusan-puluhan-satuan
;================================================
Bin2Dec:
       mov b,#100d
       div ab
       mov ratusan,a
       mov a,b
       mov b,#10d
       div ab
       mov puluhan,a
       mov satuan,b
       ret
;================================================
;Subrutin untuk menampilkan tulisan Data ADC0804
; pada baris 1
;================================================
```

```
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
       acall write_inst
write1:clr a ; A = 0
       movc a,@a+dptr       ; A = [A+ DPTR]
       mov r1,A             ; R1 = A
       inc dptr             ; DPTR = DPTR +1
       acall write_data     ;
       djnz r3,write1       ; R3 = R3-1,
       ret
Init_lcd:
       mov r1,#00000001b ;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     ; D7 s/d D0 = P0 = R1
       setb P3.7     ; EN = 1 = P2.1
       call delay    ; call delay time
       clr P3.7      ; EN = 0 = P2.1
       ret
       ;
Write_data:
       setb P3.6     ; RS = P2.0 = 1, write mode data
       mov P0,R1     ; D7 s/d D0 = P0 = R1
       setb P3.7     ; EN = 1 = P2.1
       call delay    ; call delay time
       clr p3.7      ; EN = 0 = P2.1
       ret
       ;
delay: mov R0,#0
delay1:mov R2,#0fh
       djnz R2,$
       djnz R0,delay1
       ret
       ;
word1: DB ' Data ADC0804 '
           end
```

7. Simpanlah program yang anda ketik dan beri nama : prog52a.asm
8. Pada program MIDE tersebut pilih Build /F9 atau untuk melakukan kompilasi program dari *.asm ke *.hex.
9. Lakukan pemrograman mikrokontroller dengan menggunakan Program ISP Software (Lihat Petunjuk Penggunaan)
10. Lakukan modifikasi pada program tersebut dengan manambahkan kata SUHU: , pada Baris 2 diikuti dengan data ADC.

Percobaan 5.3.
Aplikasi program komparator dengan memanfaatkan instruksi aritmatika dan instruksi lompatan untuk pengaturan suhu dengan display LCD Karakter

Dalam dunia elektronika, rangkaian komparator, umumnya diwujudkan dengan memanfaatkan rangkaian op-amp yang dibangun sebagai komparator. Sesuai dengan prinsip kerja komparator, membandingkan dua buah tegangan yang masuk pada input INV dan NON INV, untuk menghasilkan suatu output tegangan saturasi. Dengan memanfaatkan instruksi aritmatika SUBB dan instruksi lompatan JZ dan JC, maka rangkaian analog ini dapat digantikan dengan menggunakan pemrograman assembly.

Gambar 5.10 Rangkaian komparator analog dengan IC OP-AMP

Apabila tegangan yang masuk pada VREF lebih besar daripada tegangan yang masuk pada VIN maka VOUT akan mengeluarkan tegangan ~0 volt. Dan sebaliknya bila tegangan yang masuk pada VREF lebih kecil dari pada VIN maka VOUT akam mengeluarkan tegangan VSAT.

```
ControlSuhu:
       mov a,dataSetting    ; contoh dataSetting=50
       mov b,dataADC        ; contoh dataADC=30
       clr c
       subb a,b
       jnz OnHeater
       ret
OnHeater:
       jc OffHeater
       call HeaterOn        ;Instruksi hidupkan heater
       ret
OffHeater:
       Call HeaterOff       ;Instruksi matikan heater
       ret
       ;
```

Pada instruksi tersebut diambil selisih antara dataSetting dan dataADC dengan menggunakan instruksi SUBB, pengurangan ini akan menghasilkan tiga keadaan yaitu: NOL, NEGATIF atau POSITIF. Hasil-hasil inilah yang harus dideteksi, keadaan

NEGATIF dapat dideteksi dengan memantau bit C (carry), keadaan NOL dapat dideteksi dengan memantau register A (accumulator).

Apabila diberikan keadaan input sesuai dengan contoh tersebut maka:

A=dataSetting=50

B=dataADC=30

SUBB A,B

A=50-30 =20 (keadaan POSITIF)

Sesuai dengan instruksi diatas maka program akan menuju ke Ret OnHeater, pada baris ini dilakukan proses pengujian keadaan, dengan instruksi JC, karena keadaan POSITIF maka C=0 (clear) sehingga program akan memanggil HeaterOn

Apabila diberikan keadaan input sesuai dengan contoh tersebut maka:

A=dataSetting=50

B=dataADC=50

SUBB A,B

A=50-50 =00 (keadaan NOL)

Sesuai dengan instruksi diatas maka program akan menuju ke Ret.

Apabila diberikan keadaan input sesuai dengan contoh tersebut maka:

A=dataSetting=50

B=dataADC=51

SUBB A,B

A=50-51 =-1 (keadaan NEGATIF)

Sesuai dengan instruksi diatas maka program akan menuju ke OnHeater, pada baris ini dilakukan proses pengujian keadaan, dengan instruksi JC, karena keadaan NEGATIF maka C=1 (clear) sehingga program akan memanggil label OffHeater

Pada percobaan 5.3. ini indikator heater On dan Off, ditunjukkan pada layar LCD Karakter pada baris 1. seperti yang ditunjukkan pada pemrograman berikut ini:

1. Pada saat langkah pemrograman posisikan saklar togle ke posisi PROG
2. Posisikan saklar togle ke RUN untuk mengaktifkan ADC0804 CS=0
3. Hubungkan modul Microcontroller Trainer dengan power supply +5V
4. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
5. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program

```
dataSetting    equ 30h
dataADC        equ 31h
ratusan        equ 32h
puluhan        equ 33h
satuan         equ 34h
       org 0h
       mov dataSetting,#50d   ; contoh datasetting=50
       call init_lcd
start: call ADC
       call ControlSuhu
       call bin2dec
       call Display2LCD
       sjmp start
        ;
ControlSuhu:
       mov a,dataSetting      ; contoh dataSetting=50
       mov b,dataADC ; contoh dataADC=30
       clr c
       subb a,b
       jnz OnHeater
       ret
OnHeater:
       jc OffHeater
       call HeaterOn ;Instruksi hidupkan heater
       ret
OffHeater:
       Call HeaterOff ;Instruksi matikan heater
       ret
        ;
HeaterOn:
       mov R1,#80h
       call write_inst
       mov R1,#'O'
       call write_data
        ;
       mov R1,#81h
       call write_inst
       mov R1,#'n'
       call write_data
        ;
       mov R1,#82h
       call write_inst
       mov R1,#' '
       call write_data
       ret
HeaterOff:
       mov R1,#80h
       call write_inst
       mov R1,#'O'
       call write_data
        ;
       mov R1,#81h
       call write_inst
       mov R1,#'f'
       call write_data
        ;
       mov R1,#82h
       call write_inst
       mov R1,#'f'
       call write_data
       ret
```

```
;=======================================
;Subrutin ini untuk merubah data biner ke desimal
;menjadi 3 digit = ratusan-puluhan-satuan
;=======================================
Bin2Dec:
        mov A,dataADC
        mov b,#100d
        div ab
        mov ratusan,a
        mov a,b
        mov b,#10d
        div ab
        mov puluhan,a
        mov satuan,b
        ret
;================================================
;Subrutin untuk menampilkan data ke LCD character 2 x16
;pada DDRAM 0C9 0CA 0CB untukratusan, puluhan, and satuan
;================================================
Display2LCD:
        mov r1,#0c0h
        call write_inst
        mov a,ratusan
        add a,#30h
        mov r1,a
        call write_data
        ;
        mov r1,#0c1h
        call write_inst
        mov a,puluhan
        add a,#30h
        mov r1,a
        call write_data
        ;
        mov r1,#0c2h
        call write_inst
        mov a,satuan
        add a,#30h
        mov r1,a
        call write_data
        ret
;=======================================
;Subrutin ini digunakan untuk mengambil data ADC
;=======================================
ADC:    clr P3.3
        nop
        nop
        nop
        setb P3.3
eoc:    jb P3.2,eoc
        clr P3.4
        mov A,P1
        mov dataADC,A
        setb P3.4
        ret
         ;
Init_lcd:
        mov r1,#00000001b ;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       ; D7 s/d D0 = P0 = R1
        setb P3.7       ; EN = 1 = P2.1
        call delay      ; call delay time
        clr P3.7        ; EN = 0 = P2.1
        ret
        ;
Write_data:
        setb P3.6       ; RS = P2.0 = 1, write mode data
        mov P0,R1       ; D7 s/d D0 = P0 = R1
        setb P3.7       ; EN = 1 = P2.1
        call delay      ; call delay time
        clr p3.7        ; EN = 0 = P2.1
        ret
        ;
delay:  mov R0,#0
delay1: mov R2,#0fh
        djnz R2,$
        djnz R0,delay1
         ret
         end
```

6. Simpanlah program yang anda ketik dan beri nama : prog53a.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 modifikasi pada program tersebut dengan manambahkan kata SUHU: , pada Baris 2 diikuti dengan data ADC.

Percobaan 5.4.

Kalibrasi dataADC ke suhu dengan menggunakan metode Look Up Table.

Kenapa kita membutuhkan look up table?: Look up table adalah suatu cara yang digunakan untuk menghindari proses perkalian dan pembagian yang bertele-tele dan memusingkan bila dilakukan dengan menggunakan bahasa assembly, yang tentunya harus dilakukan bila kita akan kalibrasi suatu alat ukur. Contoh kalibrasi Termometer dengan menggunakan persamaan persamaan berikut ini: Suhu = DataADC * 100/ 255 ^{o}C.

Contoh table untuk konversi data ke besaran suhu (dengan menggunakan program Microsoft Excell). Karena data decimal maksimal adalah 255 dan suhu maksimal 100 maka Data look up tablenya adalah 255/100.

Pada percobaan 5.4, kalibrasi dilakukan untuk perubahan range desimal (0 s/d 255) menjadi range suhu(000.0 s/d 100.0 oC)

1. Pada saat langkah pemrograman posisikan saklar togle ke posisi PROG
2. Posisikan saklar togle ke RUN untuk mengaktifkan ADC0804 CS=0
3. Hubungkan modul Microcontroller Trainer dengan power supply +5V
4. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
5. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program

```
dataADC       equ 30h
       org 0h
start: call ADC
       call Display2SevenSegmen
       sjmp start
       ;
;=============================================
;Subrutin ini digunakan untuk mengambil data ADC
;=============================================
ADC:   clr P3.3
       nop
```

```
        nop
        nop
        setb P3.3
eoc:    jb P3.2,eoc
        clr P3.4
        mov A,P1
        mov dataADC,A
        setb P3.4
        ret
        ;
Display2SevenSegmen:
        mov DPTR,#ratusan    ; DPTR = [ Ratusan ]
        mov A,DataADC        ; A = [DataADC]
        movc A,@A+DPTR       ; A = [A+DPTR]
        mov DPTR,#Data7segmen       ; DPTR = [Data7Segmen]
        movc A,@A+DPTR       ; A = [A+DPTR]
        mov P0,A             ; Copy A ke P0
        Clr P3.5             ; Decoder, A=1,
        Clr P3.6             ; B=0
        Setb P3.7            ; dan C=1
        call delay           ; Panggil waktu tunda
        ;
        mov DPTR,#puluhan    ; DPTR = [ Puluhan ]
        mov A,DataADC        ; A = DataADC
        movc A,@A+DPTR       ; A =[ A+DPTR]
        mov DPTR,#Data7segmen       ; DPTR = [Data7Segmen]
        movc A,@A+DPTR       ; A = [A+DPTR]
        mov P0,A
        Setb P3.5
        Clr P3.6
        Setb P3.7
        call delay
        ;
        mov DPTR,#Satuan
        mov A,DataADC
        movc A,@A+DPTR
        mov DPTR,#Data7segmen
        movc A,@A+DPTR
        mov P0,A
        Clr P3.5 ;
        Setb P3.6
        Setb P3.7
        call delay
        ;
        mov DPTR,#Pecahan
        mov A,DataADC
        movc A,@A+DPTR
        mov DPTR,#Data7segmen
        movc A,@A+DPTR
        mov P0,A
        Setb P3.5 ;
        Setb P3.6
        Setb P3.7
        call delay
        ret
        ;
```

```
delay: mov R0,#0
delay1:mov R2,#0fh
       djnz R2,$
       djnz R0,delay1
       ret
       ;
Pecahan:
db 0,4,8,2,6,0,4,7,1,5,9,3,7,1,5,9,3,7,1,5,8,2,6,0,4,8,2,6,0,4,8,2,5,9,3,7,1,5
db 9,3,7,1,5,9,3,6,0,4,8,2,6,0,4,8,2,6,0,4,7,1,5,9,3,7,1,5,9,3,7,1,5,8,2,6,0,4
db 8,2,6,0,4,8,2,5,9,3,7,1,5,9,3,7,1,5,9,3,6,0,4,8,2,6,0,4,8,2,6,0,4,7,1,5,9,3
db 7,1,5,9,3,7,1,5,8,2,6,0,4,8,2,6,0,4,8,2,5,9,3,7,1,5,9,3,7,1,5,9,3,6,0,4,8,2
db 6,0,4,8,2,6,0,4,7,1,5,9,3,7,1,5,9,3,7,1,5,8,2,6,0,4,8,2,6,0,4,8,2,5,9,3,7,1
db 5,9,3,7,1,5,9,3,6,0,4,8,2,6,0,4,8,2,6,0,4,7,1,5,9,3,7,1,5,9,3,7,1,5,8,2,6,0
db 4,8,2,6,0,4,8,2,5,9,3,7,1,5,9,3,7,1,5,9,3,6,0,4,8,2,6,0
;
Satuan:
db 0,0,0,1,1,2,2,2,3,3,3,4,4,5,5,5,6,6,7,7,7,8,8,9,9,9,0,0,1,1,1,2,2,2,3,3,4,4
db 4,5,5,6,6,6,7,7,8,8,8,9,9,0,0,0,1,1,2,2,2,3,3,3,4,4,5,5,5,6,6,7,7,7,8,8,9,9
db 9,0,0,1,1,1,2,2,2,3,3,4,4,4,5,5,6,6,6,7,7,8,8,8,9,9,0,0,0,1,1,2,2,2,3,3,3,4
db 4,5,5,5,6,6,7,7,7,8,8,9,9,9,0,0,1,1,1,2,2,2,3,3,4,4,4,5,5,6,6,6,7,7,8,8,8,9
db 9,0,0,0,1,1,2,2,2,3,3,3,4,4,5,5,5,6,6,7,7,7,8,8,9,9,9,0,0,1,1,1,2,2,2,3,3,4
db 4,4,5,5,6,6,6,7,7,8,8,8,9,9,0,0,0,1,1,2,2,2,3,3,3,4,4,5,5,5,6,6,7,7,7,8,8,9
db 9,0,0,0,1,1,2,2,2,3,3,3,4,4,5,5,5,6,6,7,7,7,8,8,9,9,9,0
;
puluhan:
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1
db 1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
db 2,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4
db 4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5
db 5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7,7,7,7,7
db 7,7,7,7,7,7,7,7,7,7,7,7,7,7,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8
db 8,8,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,0
;
Ratusan:
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
;
Data7segmen:
           db 11000000b,11111001b,10100100b,10110000b,10011001b
           db 10010010b,10000010b,11111000b,10000000b,10010000b
           ;
           end
```

Percobaan 5.5.

Display data suhu ‘’000,0 ^{o}C S/D 100,0 ^{o}C”

Pada percobaan 5.5, kalibrasi dilakukan untuk perubahan range desimal (0 s/d 255) menjadi range suhu(000.0 s/d 100.0 ^{o}C) dengan menambahkan karakter koma, derajat dan Celcius yang ditampilkan pada 7 segmen.

1. Pada saat langkah pemrograman posisikan saklar togle ke posisi PROG
2. Posisikan saklar togle ke RUN untuk mengaktifkan ADC0804 CS=0
3. Hubungkan modul Microcontroller Trainer dengan power supply +5V
4. Hubungkan modul Microcontroller Trainer dengan rangkaian programmer
5. Buka Program M-IDE Studio for MCS-51, sebagai editor dan compiler program

```
dataADC       equ 30h
       org 0h
start: Call ADC
       Call display2sevensegmen
       sjmp start
       ;
ADC:   clr P3.3
       nop
       nop
       nop
       setb P3.3
eoc:   jb P3.2,eoc
       clr P3.4
       mov A,P1
       mov dataADC,A
       setb P3.4
       ret
       ;
Display2sevensegmen:
       Mov DPTR,#ratusan
       Mov A,DataADC
       Movc A,@A+DPTR
       mov DPTR,#data7segmen
       Movc A,@A+DPTR
       Mov P0,A
       clr P3.5
       Clr P3.6
       clr P3.7
       call delay
       ;
       mov DPTR,#puluhan
       mov A,DataADC
       movc A,@A+DPTR
       mov DPTR,#data7segmen
       movc A,@A+DPTR
       mov P0,A
       setb P3.5
       clr P3.6
```

```
        clr P3.7
        call delay
        ;
        mov DPTR,#satuan
        mov A,dataADC
        movc A,@A+DPTR
        mov DPTR,#data7segmen
        movc A,@A+DPTR
        mov P0,A
        clr P3.5
        setb P3.6
        clr P3.7
        call delay
        ;
        mov DPTR,#pecahan
        mov A,dataADC
        movc A,@A+DPTR
        mov DPTR,#data7segmen
        movc A,@A+dptr
        mov P0,A
        clr P3.5
        clr P3.6
        setb P3.7
        call delay
        ;
        setb P3.5
        setb P3.6
        clr P3.7
        mov P0,#11101111b
        call delay
        ;
        clr P3.5
        setb P3.6
        setb P3.7
        mov P0,#10011100b
        call delay
        ;
        setb P3.5
        setb P3.6
        setb P3.7
        mov P0,#11000110b
        call delay
        ret
delay:  mov R0,#0
delay1:mov R2,#5h
        djnz R2,$
        djnz R0,delay1
        ret
        ;
pecahan:
db 0,4,8,2,6,0,4,7,1,5,9,3,7,1,5,9,3,7,1,5,8,2,6,0,4,8,2,6,0,4,8,2,5,9,3,7,1,5
db 9,3,7,1,5,9,3,6,0,4,8,2,6,0,4,8,2,6,0,4,7,1,5,9,3,7,1,5,9,3,7,1,5,8,2,6,0,4
db 8,2,6,0,4,8,2,5,9,3,7,1,5,9,3,7,1,5,9,3,6,0,4,8,2,6,0,4,8,2,6,0,4,7,1,5,9,3
db 7,1,5,9,3,7,1,5,8,2,6,0,4,8,2,6,0,4,8,2,5,9,3,7,1,5,9,3,7,1,5,9,3,6,0,4,8,2
db 6,0,4,8,2,6,0,4,7,1,5,9,3,7,1,5,9,3,7,1,5,8,2,6,0,4,8,2,6,0,4,8,2,5,9,3,7,1
db 5,9,3,7,1,5,9,3,6,0,4,8,2,6,0,4,8,2,6,0,4,7,1,5,9,3,7,1,5,9,3,7,1,5,8,2,6,0
db 4,8,2,6,0,4,8,2,5,9,3,7,1,5,9,3,7,1,5,9,3,6,0,4,8,2,6,0
```

```
;
satuan:
db 0,0,0,1,1,2,2,2,3,3,3,4,4,5,5,5,6,6,7,7,7,8,8,9,9,9,0,0,1,1,1,2,2,2,3,3,4,4
db 4,5,5,6,6,6,7,7,8,8,8,9,9,0,0,0,1,1,2,2,2,3,3,3,4,4,5,5,5,6,6,7,7,7,8,8,9,9
db 9,0,0,1,1,1,2,2,2,3,3,4,4,4,5,5,6,6,6,7,7,8,8,8,9,9,0,0,0,1,1,2,2,2,3,3,3,4
db 4,5,5,5,6,6,7,7,7,8,8,9,9,9,0,0,1,1,1,2,2,2,3,3,4,4,4,5,5,6,6,6,7,7,8,8,8,9
db 9,0,0,0,1,1,2,2,2,3,3,3,4,4,5,5,5,6,6,7,7,7,8,8,9,9,9,0,0,1,1,1,2,2,2,3,3,4
db 4,4,5,5,6,6,6,7,7,8,8,8,9,9,0,0,0,1,1,2,2,2,3,3,3,4,4,5,5,5,6,6,7,7,7,8,8,9
db 9,0,0,0,1,1,2,2,2,3,3,3,4,4,5,5,5,6,6,7,7,7,8,8,9,9,9,0
;
puluhan:
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1
db 1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
db 2,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4
db 4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5,5
db 5,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7,7,7,7,7
db 7,7,7,7,7,7,7,7,7,7,7,7,7,7,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8
db 8,8,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,0
;
ratusan:
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
             ;
data7segmen:
             db 11000000b,11111001b,10100100b,10110000b,10011001b
             db 10010010b,10000010b,11111000b,10000000b,10010000b
             ;
             end
```