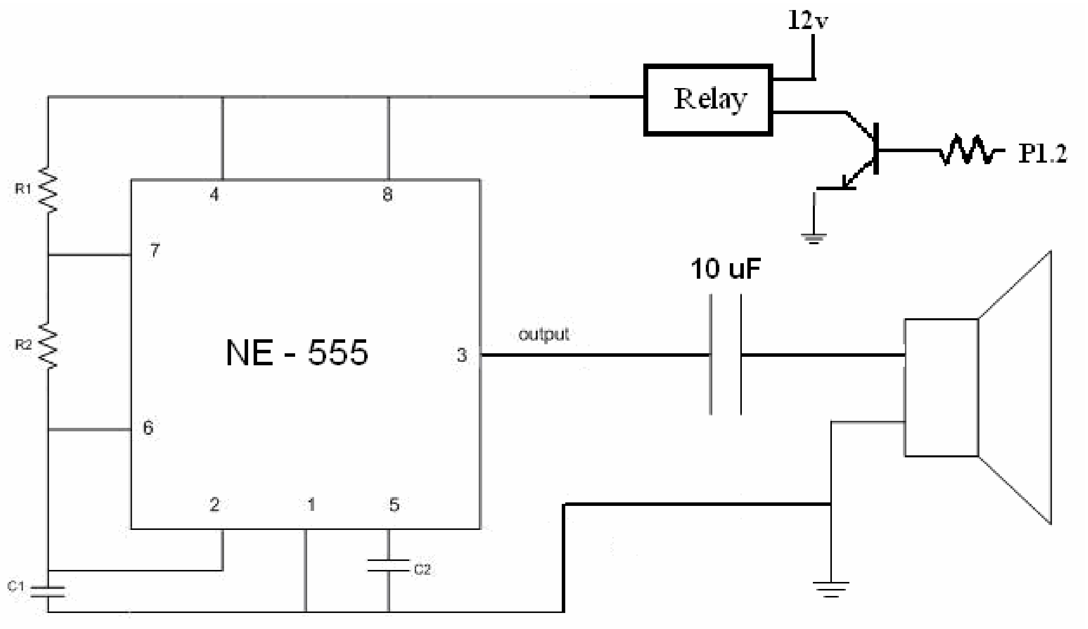
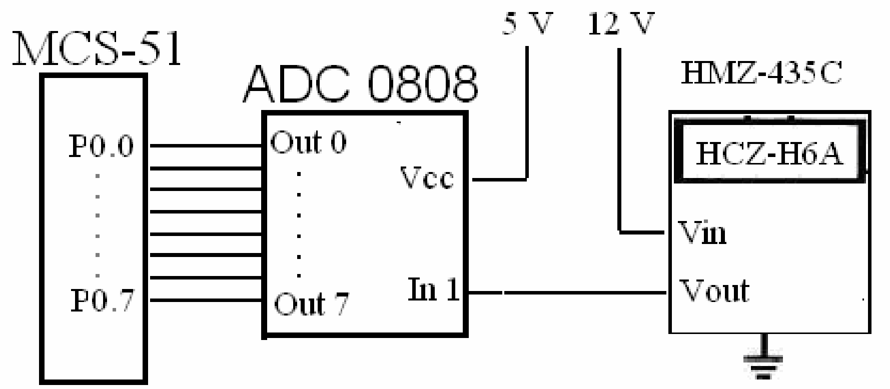
LAMPIRAN A
RANGKAIAN SKEMATIK

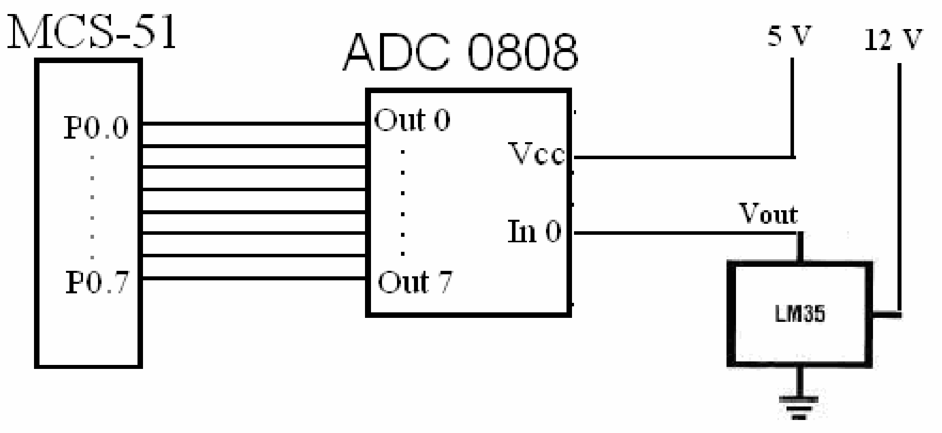
Lampiran A.1 Perancangan Generator Frekuensi Audio



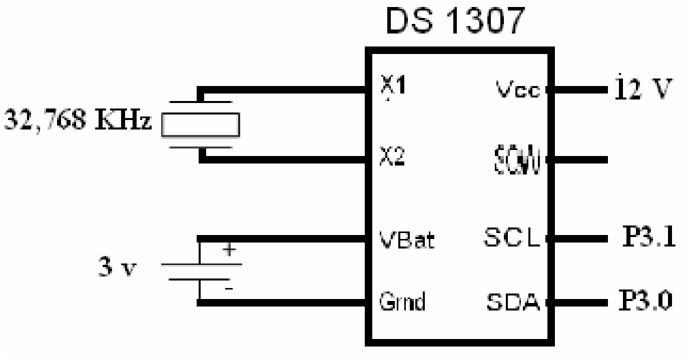
Lampiran A.1 Perancangan Generator Frekuensi Audio



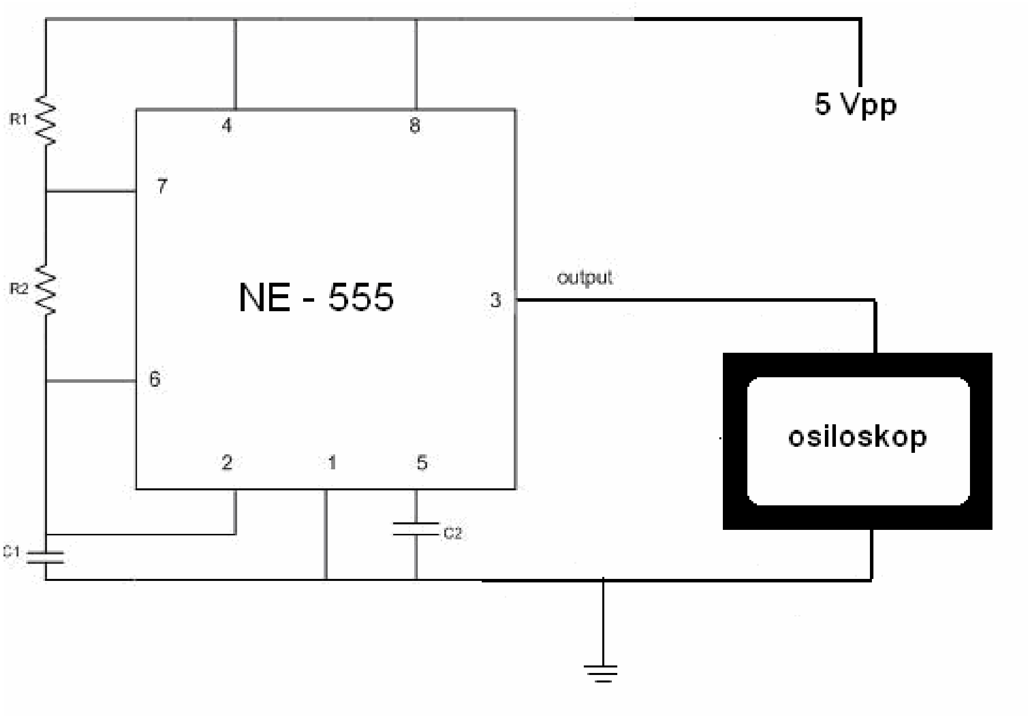
Lampiran A.2 Perancangan Sensor Kelembaban



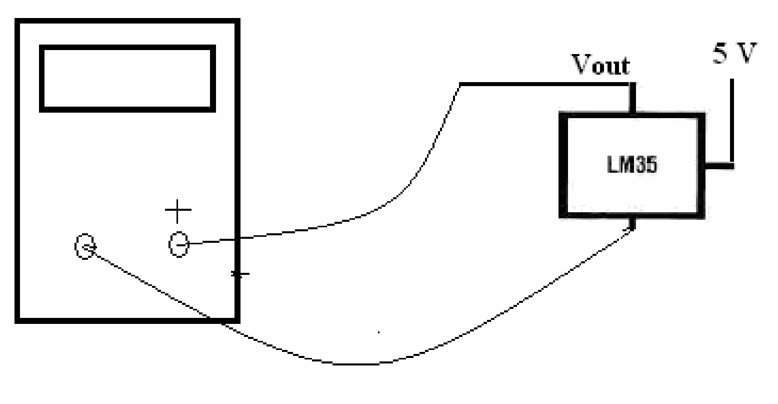
Lampiran A.3 Perancangan Sensor Temperatur



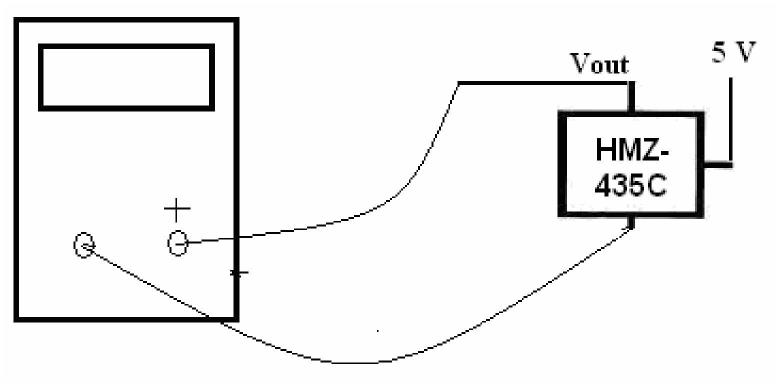
Lampiran A.4 Perancangan RTC (*Real Time Clock*)



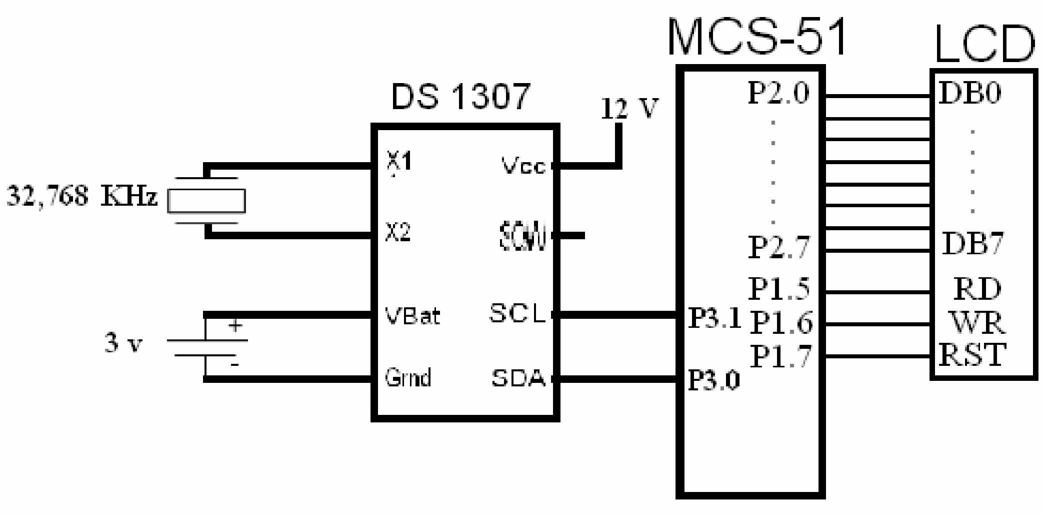
Lampiran A.5 Pengujian Generator Frekuensi Audio



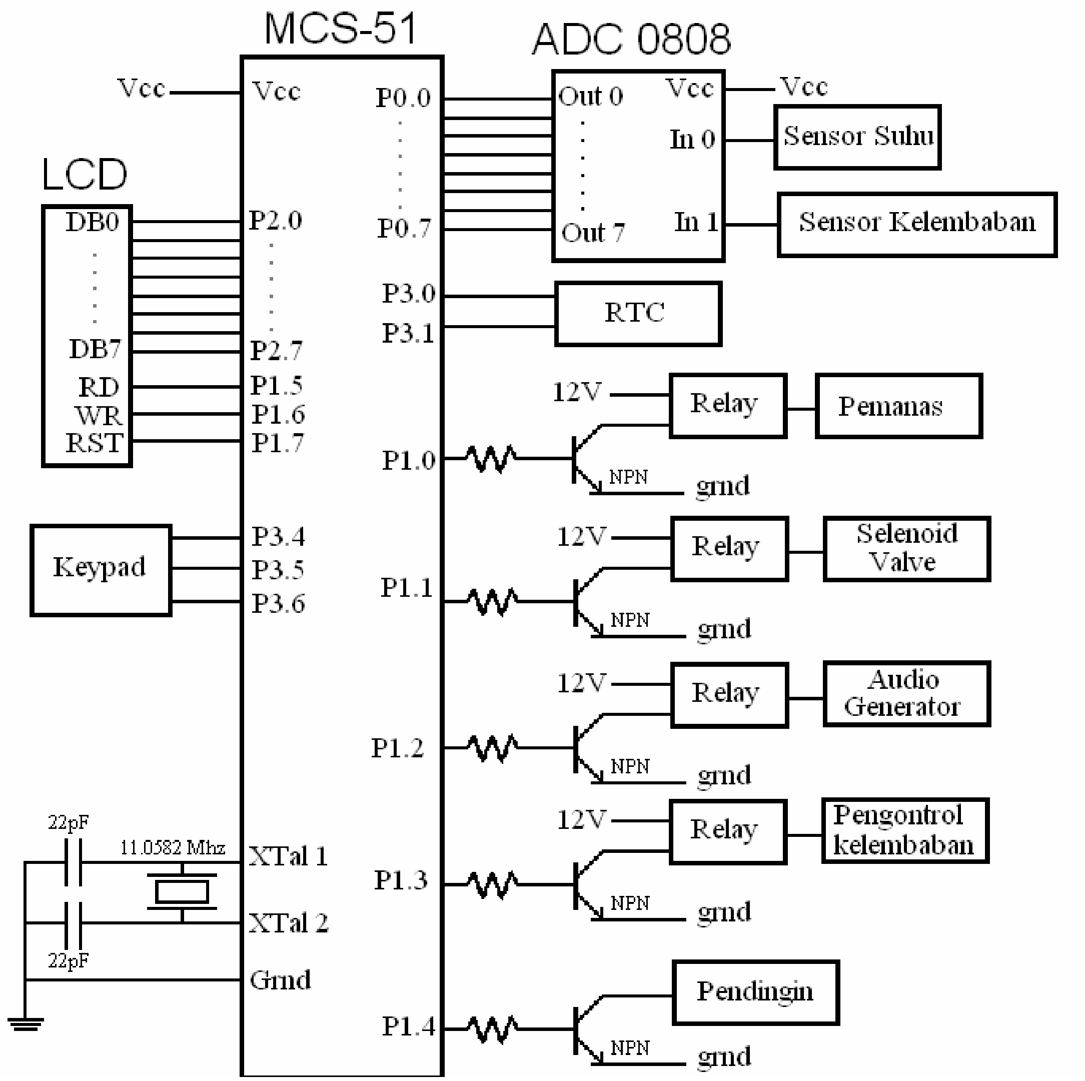
Lampiran A.6 Pengujian Sensor Temperatur



Lampiran A.7 Pengujian Sensor Kelembaban



Lampiran A.4 Pengujian RTC (*Real Time Clock*)



Lampiran A.9 Rangkaian Skematik Sistem

LAMPIRAN B
SOFTWARE

```
$mod51

org 00h

cleardisplay    equ 01h
returnhome      equ 02h
entrymode       equ 06h
displaycontrol  equ 0fh
shift           equ 14h
functionset     equ 38h
cgramaddress    equ 40h
ddramaddress    equ 80h
pt              equ 00h
ph              equ 04h
ds1307w         equ 0d0h
ds1307r         equ 0d1h
bitcount        equ 21h
bytecount       equ 22h
byte            equ 23h

lcdrs           bit p1.5
lcdrw           bit p1.6
lcden           bit p1.7
scl             bit p3.0
sda             bit p3.1
;lastread       bit flags.0
;ack            bit flags.5
;bus_fault      bit flags.6
;_2w_busy       bit flags.7

lastread        bit psw.0
ack             bit psw.5
bus_fault       bit psw.6
_2w_busy        bit psw.7

lcall rtc_init
lcall lcdinit
lcall send_start
call tampilan1
call delay_menu
call clr_lcd
call tampilan2
call delay_menu
call clr_lcd
mov r0,#00h
call pisah
call tampilan2a
menu_a:
mov a,p3
anl a,#00111000b
```

```
cjne a,#08h,cek1
jmp tombol1
cek1:
cjne a,#16h,cek2
jmp tombol2
cek2:
cjne a,#32h,menu_a
jmp tombol3
tombol1:
inc r0
call pisah
mov a,r0
mov r5,a
call hit1
call angka
call tampilan2a
cjne r0,#18h,menu_a
jmp loop1
tombol2:
dec r0
call pisah
mov a,r0
mov r5,a
call hit1
call angka
call tampilan2a
cjne r0,#00h,menu_a
jmp loop2
tombol3:
call clr_lcd
mov a,r0
mov 30h,r0
jmp menu_b1
loop1:
mov r0,#00h
sjmp menu_a
loop2:
mov r0,#18h
sjmp menu_a
menu_b1:
mov r1,#00h
mov a,r1
mov r0,a
call pisah
call tampilan2b
jmp menu_b
menu_b:
mov a,p3
anl a,#00111000b
cjne a,#08h,cek11
jmp tombol11
```

```
cek11:
cjne a,#16h,cek22
jmp tombol22
cek22:
cjne a,#32h,menu_b
jmp tombol33
tombol11:
inc r1
call pisah
mov a,r1
mov r5,a
call hit1
call angka
call tampilan2b
cjne r1,#3ch,menu_b
jmp loop11
tombol22:
dec r1
call pisah
mov a,r1
mov r5,a
call hit1
call angka
call tampilan2b
cjne r1,#00h,menu_b
jmp loop22
tombol33:
call clr_lcd
mov a,r1
mov 31h,r1
jmp menu_c1
loop11:
mov r1,#00h
jmp menu_b
loop22:
mov r1,#3ch
jmp menu_b
menu_c1:
call clr_lcd
call tampilan3
call delay_menu
mov r2,#00h
mov a,r2
mov r0,a
call pisah
call tampilan3a
jmp menu_c
menu_c:
mov a,p3
anl a,#00111000b
cjne a,#08h,cek111
```

```
jmp tombol111
cek111:
cjne a,#16h,cek222
jmp tombol222

cek222:
cjne a,#32h,menu_c
jmp tombol333
tombol111:
inc r2
call pisah
mov a,r2
mov r5,a
call hit1
call angka
call tampilan3a
cjne r2,#18h,menu_c
jmp loop111
tombol222:
dec r2
call pisah
mov a,r2
mov r5,a
call hit1
call angka
call tampilan3a
cjne r2,#00h,menu_c
jmp loop222
tombol333:
call clr_lcd
mov a,r2
mov 32h,r2
jmp menu_d1
loop111:
mov r2,#00h
jmp menu_c
loop222:
mov r2,#18h
jmp menu_c
menu_d1:
mov r3,#00h
mov a,r3
mov r0,a
call pisah
call tampilan3b
jmp menu_d
menu_d:
mov r3,#00h
call delay_menu
call tampilan3b
mov a,p3
```

```
anl a,#00111000b
cjne a,#08h,cek1111
sjmp tombol1111
cek1111:
cjne a,#16h,cek2222
jmp tombol2222
cek2222:
cjne a,#32h,menu_d
jmp tombol3333
tombol1111:
inc r3
call pisah
mov a,r3
mov r5,a
lcall hit1
call angka
call tampilan3b
cjne r3,#3ch,menu_d
jmp loop1111
tombol2222:
dec r3
call pisah
mov a,r3
mov r5,a
lcall hit1
call angka
call tampilan3b
cjne r3,#00h,menu_d
jmp loop2222
tombol3333:
mov a,r3
mov 33h,r3
jmp menu_e1
loop1111:
mov r3,#00h
sjmp menu_d
loop2222:
mov r3,#3ch
jmp menu_d
menu_e1:
call clr_lcd
call tampilan4
call delay_menu
mov r4,#00h
mov a,r4
mov r0,a
call pisah
call tampilan4a
jmp menu_e
menu_e:
mov a,p3
```

```
anl a,#00111000b
cjne a,#08h,cek11111
jmp tombol11111
cek11111:
cjne a,#16h,cek22222
jmp tombol22222

cek22222:
cjne a,#32h,menu_e
jmp tombol33333
tombol11111:
inc r4
call pisah
mov a,r4
mov r5,a
lcall hit1
call angka
call tampilan4a
cjne r4,#64h,menu_e
jmp loop11111
tombol22222:
dec r4
call pisah
mov a,r4
mov r5,a
lcall hit1
call angka
call tampilan4a
cjne r4,#00h,menu_e
jmp loop22222
tombol33333:
mov 34h,r4
jmp menu_f1
loop11111:
mov r4,#99h
jmp menu_e
loop22222:
mov r4,#00h
jmp menu_e
menu_f1:
mov r5,#00h
mov a,r5
mov r0,a
call pisah
call tampilan4b
jmp menu_f
menu_f:
mov a,p3
anl a,#00111000b
cjne a,#08h,cek111111
jmp tombol111111
```

```
cek111111:
cjne a,#16h,cek222222
jmp tombol222222
cek222222:
cjne a,#32h,menu_f
jmp tombol333333

tombol111111:
inc r5
call pisah
lcall hit1
call angka
call tampilan4b
cjne r5,#64h,menu_f
jmp loop111111
tombol222222:
dec r5
call pisah
mov a,r5
mov r5,a
lcall hit1
call angka
call tampilan4b
cjne r5,#00h,menu_f
jmp loop222222
tombol333333:
mov 35h,r5
jmp menu_g1
loop111111:
mov r5,#99
jmp menu_f
loop222222:
mov r5,#00
jmp menu_f
menu_g1:
call clr_lcd
call tampilan5
call delay_menu
mov r6,#00h
mov a,r6
mov r0,a
call pisah
call tampilan5a
jmp menu_g
menu_g:
mov a,p3
anl a,#00111000b
cjne a,#08h,cek1111111
jmp tombol1111111
cek1111111:
cjne a,#16h,cek2222222
```

```
jmp tombol2222222
cek2222222:
cjne a,#32h,menu_g
jmp tombol3333333
tombol1111111:
inc r6
call pisah
mov a,r6
mov r5,a
lcall hit1
call angka
call tampilan5a
cjne r6,#64h,menu_g
jmp loop1111111
tombol2222222:
dec r6
call pisah
mov a,r6
mov r5,a
lcall hit1
call angka
call tampilan5a
cjne r6,#00h,menu_g
jmp loop2222222
tombol3333333:
mov 36h,r6
jmp menu_g
loop1111111:
mov r6,#64h
sjmp menu_g
loop2222222:
mov r6,#00h
jmp menu_g
menu_h1:
mov r7,#00h
mov a,r7
mov r0,a
call pisah
call tampilan5b
jmp menu_h
menu_h:
mov a,p3
anl a,#00111000b
cjne a,#08h,cek11111111
jmp tombol11111111
cek11111111:
cjne a,#16h,cek22222222
jmp tombol222222
cek22222222:
cjne a,#32h,menu_h
jmp tombol33333333
```

```
tombol11111111:
inc r7
call pisah
mov a,r7
mov r5,a
lcall hit1
call angka
call tampilan5b
cjne r6,#5fh,menu_h
jmp loop11111111
tombol22222222:
dec r7
call pisah
mov a,r7
mov r5,a
lcall hit1
call angka
call tampilan5b
cjne r7,#00h,menu_h
jmp loop22222222
tombol33333333:
mov 30h,r7
jmp tampilan6
loop11111111:
mov r7,#5fh
jmp menu_h
loop22222222:
mov r7,#00h
jmp menu_h
tampilan6:
mov a,#'t'
call display
call cursor
call cursor
call fsuhu1
call angka
mov a,r6
call display
call cursor
mov a,r7
call display
call cursor
mov r1,#12
call geser
call cursor
mov a,#'h'
call display
call cursor
call cursor
mov a,45h
call dbase2
```

```
mov r1,#17
call geser
call cursor
call baca
mov r0,39h
call pisah
call angka
mov a,r6
call display
call cursor
mov a,r7
call display
call cursor
mov a,3ah
call display
call cursor
call baca
mov r0,40h
call pisah
call angka
mov a,r6
call display
call cursor
mov a,r7
call display
call cursor
mov a,3ah
call display
call cursor
call baca
mov r0,41h
call pisah
call angka
fwaktu:
call baca
jam1:
mov a,39h
cjne a,30h,jam2
jmp menit1
menit1:
mov a,40h
cjne a,31h,jam2
jmp pagi
pagi:
mov a,p1
mov b,#02h
add a,b
mov a,p1
mov b,#08h
add a,b
call
```

```
call mati_pagi1
call mati_pagi2
jmp fsuhu
mati_pagi1:
mov a,p1
anl a,#11111101b
ret
mati_pagi2:
mov a,p1
anl a,#11111011b
ret

jam2:
mov a,39h
cjne a,32h,fsuhu
jmp menit2
menit2:
mov a,40h
cjne a,33h,fsuhu
call siang
jmp fsuhu
siang:
mov a,p1
mov b,#08h
add a,b
call mati_pagi2
jmp fsuhu
fsuhu:
mov a,#pt
mov 46h,a
call adc
mov r5,a
lcall hit
jmp cek_suhu
cek_suhu:
lcall hit2
cjne a,34h,gantung1t
jmp stabilt
gantung1t:
jc negatipt
jmp positipt
negatipt:
mov a,p1
mov b,#01h
add a,b
jmp mati1t
positipt:
cjne a,35h,gantung2t
jmp stabilt
gantung2t:
jc stabilt
```

```
mov a,p1
mov b,#10h
add a,b
jmp mati2t
mati1t:
cjne a,34h,mati11t
mov a,p1
anl a,#00000001b
mov b,00h
add a,b
jmp stabilt

mati11t:
jc negatipt
jmp stabilt
mati2t:
cjne a,36h,mati22t
mov a,p1
anl a,#00010000b
mov b,00h
add a,b
jmp stabilt
mati22t:
jc stabilt
jmp gantung2t
stabilt:
jmp fkelembaban
fkelembaban:
mov a,#ph
mov 46h,a
call adc
lcall hit2
mov 45h,a
call dbase2
mov a,45h
cjne a,36h,gantung1h
ljmp hit
jmp fwaktu
gantung1h:
jc negatiph
jmp positiph
negatiph:
mov a,p1
mov b,#02h
add a,b
mov b,a
mov a,p1
mov 60h,#10h
add a,60h
jmp mati1h
positiph:
```

```
cjne a,37,gantung2h
jmp stabilh
gantung2h:
mov a,p1
mov b,a
mov a,#08h
add a,b
jmp mati2h
mati1h:
cjne a,36h,mati11h
mov a,p1
anl a,#00010000b
mov b,#00h
add a,b
mov 65h,a
mov a,p1
anl a,#00000010b
mov 60h,#00h
add a,60h
jmp fwaktu
mati11h:
jc negatiph
jmp fwaktu
mati2h:
cjne a,37h,mati22h
mov a,p1
anl a,#00001000b
mov b,00h
add a,b
jmp stabilh
mati22h:
jc stabilh
jmp gantung2h
stabilh:
jmp fwaktu
lcdinit:
lcall delay
anl p1,#11100111b
mov p2,#functionset
lcall wrtcmd
mov r5,#5
lcall delay
mov p2,#functionset
lcall wrtcmd
mov r6,#50
djnz r6,$
mov p2,#functionset
lcall wrtcmd
ret
input:
mov a,#'i'
```

```
call display
call cursor
mov a,#'n'
call display
call cursor
mov a,#'p'
call display
call cursor
mov a,#'u'
call display
call cursor
mov a,#'t'
call display
call cursor
ret
tampilan1:
mov a,#'t'
call display
call cursor
mov a,#'u'
call display
call cursor
mov a,#'g'
call display
call cursor
mov a,#'a'
call display
call cursor
mov a,#'s'
call display
call cursor
call cursor
mov a,#'a'
call display
call cursor
mov a,#'k'
call display
call cursor
mov a,#'h'
call display
call cursor
mov a,#'i'
call display
call cursor
mov a,#'r'
call display
call cursor
mov r1,#17
call geser
call cursor
mov a,#'a'
```

```
call display
call cursor
mov a,#'n'
call display
call cursor
mov a,#'o'
call display
call cursor
mov a,#'e'
call display
call cursor
mov a,#'s'
call display
call cursor
ret

tampilan2:
call input
call cursor
mov a,#'w'
call display
call cursor
mov a,#'a'
call display
call cursor
mov a,#'k'
call display
call cursor
mov a,#'t'
call display
call cursor
mov a,#'u'
call display
call cursor
call cursor
mov a,#'i'
call display
call cursor
ret
tampilan2a:
mov a,#'c'
call display
call cursor
call cursor
mov a,#'i'
call display
call cursor
mov r1,#26
call geser
call cursor
mov a,#'j'
```

```
call display
call cursor
mov a,#'a'
call display
call cursor
mov r0,#00h
call display
mov a,#'m'
call display
call cursor
call cursor
mov a,r6
call display
call cursor
mov a,r7
call display
call cursor
ret
tampilan2b:
mov r1,#24
call geser
call cursor
mov a,#'m'
call display
call cursor
mov a,#'e'
call display
call cursor
mov a,#'n'
call display
call cursor
mov a,#'i'
call display
call cursor
mov a,#'t'
call display
call cursor
call cursor
mov a,r6
call display
call cursor
mov a,r7
call display
call cursor
ret
tampilan3:
call tampilan2
mov a,#'i'
call display
call cursor
ret
```

```
tampilan3a:
mov a,#'c'
call display
call cursor
call cursor
mov a,#'i'
call display
call cursor
call display
mov r1,#26
call geser
call cursor
mov a,#'j'
call display
call cursor
mov a,#'a'
call display
call cursor
mov a,#'m'
call display
call cursor
call cursor
mov a,r6
call display
call cursor
mov a,r7
call display
call cursor
ret
tampilan3b:
mov r1,#24
call geser
call cursor
mov a,#'m'
call display
call cursor
mov a,#'e'
call display
call cursor
mov a,#'n'
call display
call cursor
mov a,#'i'
call display
call cursor
mov a,#'t'
call display
call cursor
call cursor
mov a,r6
call display
call cursor
mov a,r7
call display
call cursor
ret
tampilan4:
call input
call cursor
mov a,#'t'
call display
call cursor
mov a,#'e'
call display
call cursor
mov a,#'m'
call display
call cursor
mov a,#'p'
call display
call cursor
mov a,#'e'
call display
call cursor
mov a,#'r'
call display
call cursor
mov a,#'a'
call display
call cursor
mov a,#'t'
call display
call cursor
mov a,#'u'
call display
call cursor
mov a,#'r'
call display
call cursor
ret
tampilan4a:
mov r1,#24
call geser
call cursor
mov a,#'t'
call display
call cursor
mov a,#'i'
call display
call cursor
call cursor
mov a,r6
```

```
call display
call cursor
mov a,r7
call display
call cursor
ret
tampilan4b:
mov r1,#23
call geser
call cursor
mov a,#'t'
call display
call cursor
mov a,#'i'
call display
call cursor
call cursor
mov a,r6
call display
call cursor
mov a,r7
call display
call cursor
ret
tampilan5:
call input
call cursor
mov a,#'h'
call display
call cursor
call display
mov a,#'u'
call display
call cursor
call display
mov a,#'m'
call display
call cursor
call display
mov a,#'i'
call display
call cursor
call display
mov a,#'d'
call display
call cursor
call display
mov a,#'i'
call display
call cursor
call display
```

```
mov a,#'t'
call display
call cursor
call display
mov a,#'y'
call display
call cursor
call display
ret
tampilan5a:
mov r1,#24
call geser
call cursor
mov a,#'h'
call display
call cursor
mov a,#'i'
call display
call cursor
call cursor
mov a,r6
call display
call cursor
mov a,r7
call display
call cursor
ret
tampilan5b:
mov r1,#23
call geser
call cursor
mov a,#'h'
call display
call cursor
mov a,#'i'
call display
call cursor
call cursor
mov a,r6
call display
call cursor
mov a,r7
call display
call cursor
ret
cursor:
lcall     busy
cjne      a, #10h, text
mov       p2, #0c0h
lcall     wrtcmd
sjmp      text
```

```
geser:
djnz r1,geser
call cursor
ret
clr_lcd:
mov a,p2
anl a,#00h
ret
display:
mov p2, a
lcall writedata
ret
writecommand:
mov	r5, p2
lcall	busy
mov	p2, r5
wrtcmd:
clr lcdrs
clr lcdrw
clr lcden
setb lcden
clr lcden
call readram
ret
writedata:
mov r6, p2
lcall busy
mov p2, r6
wrtdata:
setb lcdrs
clr lcdrw
clr lcden
setb lcden
clr lcden
ret
readram:
lcall busy
clr lcden
clr lcdrs
clr lcdrw
mov p2, #0ffh
setb lcdrs
setb lcdrw
setb lcden
nop
mov a, p2
clr lcden
clr lcdrs
clr lcdrw
ret
read:
```

```
clr lcden
clr lcdrs
clr lcdrw
mov p2,#0ffh
setb lcdrw
setb lcden
nop
mov a,p2
clr lcden
clr lcdrw
ret
busy:
lcall read
jb acc.7, busy
lcall read
jb acc.7, busy
ret
wait5s:
mov r4,#50
loadr5:
mov r5,#200
loadr6:
mov r6,#250
djnz r6,$
djnz r5,loadr6
djnz r4,loadr5
ret
delay:
lcall dly1ms
djnz r5, delay
ret
dly1ms:
mov r6,#250
loop:
nop
nop
djnz r6, loop
ret
delay_5m:
mov r1,#79h
dly3:
mov r2,#79h
dly4:
mov r3,#79h
dly5:
djnz r3,dly2
djnz r2,dly1
djnz r1,dly0
ret
delay_adc:
mov r1,#39
```

```
dly3:
mov r2,#39
dly4:
mov r3,#39
dly5:
djnz r3,dly2
djnz r2,dly1
djnz r1,dly0
ret
delay_menu:
mov r1,#70h
dly0:
mov r2,#70h
dly1:
mov r3,#70h
dly2:
djnz r3,dly2
djnz r2,dly1
djnz r1,dly0
ret
hit:
mov a,r5
anl a,#11110000b
rl a
rl a
rl a
rl a
mov b,#16
mul ab
mov r6,a
mov a,r5
anl a,#00001111b
mov b,a
mov a,r6
add a,b
mov b,#5
div ab
mov r6,a
mov r7,b
ret
hit1:
mov a,r5
anl a,#11110000b
rl a
rl a
rl a
rl a
mov b,#16
mul ab
mov r6,a
mov a,r5
```

```
anl a,#00001111b
mov b,a
mov a,r6
add a,b
ret
hit2:
mov a,r6
mov b,#10
mul ab
mov b,r7
add a,b
ret
adc:
mov a,p3
mov b,46
add a,b
mov p3,a
call delay_adc
nop
mov a,p0
ret
pisah:
mov a,r0
anl a,#11110000b
rl a
rl a
rl a
rl a
mov r6,a
mov a,r0
anl a,#00001111b
mov r7,a
ret
angka:
mov a,#30h
mov b,r6
add a,b
mov r6,a
mov a,#30h
mov b,r7
add a,b
mov r7,a
ret
fsuhu1:
mov a,#pt
mov 46,a
call adc
mov r5,a
lcall hit
ret
baca:
```

```
mov r1,#39
mov bytecount,#00h
clr lastread
lcall send_start
mov a,#ds1307w
lcall send_byte
mov a,#08h
lcall send_byte
lcall send_stop
lcall send_start
mov a,#ds1307r
lcall send_byte
read_loop2:
mov a,bytecount
cjne a,#37h,not_last2
setb lastread
not_last2:
lcall read_byte
mov @r1,a
inc r1
inc bytecount
mov a,bytecount
cjne a,#38h,read_loop2
lcall send_stop
clr ri
ret
rtc_init:
setb sda
lcall scl_high
clr ack
clr bus_fault
clr _2w_busy
ret
scl_high:
setb scl
jnb scl,$
ret
send_stop:
clr sda
lcall scl_high
setb sda
clr _2w_busy
ret
delay:
nop
ret
send_start:
setb _2w_busy
clr ack
clr bus_fault
jnb scl,fault
jnb sda,fault
setb sda
lcall scl_high
clr sda
lcall delay
clr scl
ret
fault:
setb bus_fault
ret
send_byte:
mov bitcount,#08h
sb_loop:
jnb acc.7,notone
setb sda
jmp one
notone:
clr sda
one:
lcall scl_high
rl a
clr scl
djnz bitcount,sb_loop
setb sda
lcall scl_high
clr ack
jnb sda,sb_ex
setb ack
sb_ex:
lcall delay
clr scl
lcall delay
ret
read_byte:
mov bitcount,#008h
mov a,#00h
setb sda
read_bits:
lcall scl_high
mov c,sda
rlc a
clr scl
djnz bitcount,read_bits
jb lastread,ackn
clr sda
ackn:
lcall scl_high
clr scl
ret
dbase2:
hh1:
```

```
cjne a,#1ch,hh2
mov a,#31h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh2:
cjne a,#1dh,hh3
mov a,#31h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh3:
cjne a,#1eh,hh4
mov a,#31h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh4:
cjne a,#1fh,hh5
mov a,#31h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh5:
cjne a,#1fh,hh6
mov a,#31h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh6:
cjne a,#20h,hh7
mov a,#31h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh7:
cjne a,#21h,hh8
mov a,#31h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh8:
cjne a,#22h,hh9
mov a,#31h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh9:
cjne a,#23h,hh10
mov a,#31h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh10:
cjne a,#24h,hh11
mov a,#31h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh11:
cjne a,#25h,hh12
mov a,#31h
call display
call cursor
mov a,#'15'
call display
call cursor
ret
hh12:
cjne a,#26h,hh13
mov a,#31h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh13:
cjne a,#27h,hh14
mov a,#32h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh14:
cjne a,#28h,hh15
mov a,#32h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh15:
cjne a,#29h,hh16
mov a,#31h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh16:
cjne a,#2ah,hh17
mov a,#31h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh17:
cjne a,#2bh,hh18
mov a,#31h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh18:
```

```
cjne a,#2ch,hh19
mov a,#31h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh19:
cjne a,#2dh,hh20
mov a,#32h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh20:
cjne a,#2eh,hh21
mov a,#32h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh21:
cjne a,#2fh,hh22
mov a,#32h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh22:
cjne a,#30h,hh23
mov a,#32h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh23:
cjne a,#31h,hh24
mov a,#32h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh24:
cjne a,#32h,hh25
mov a,#32h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh25:
cjne a,#33h,hh26
mov a,#32h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh26:
cjne a,#34h,hh27
mov a,#32h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh27:
cjne a,#35h,hh28
mov a,#32h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh28:
cjne a,#36h,hh29
mov a,#32h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh29:
cjne a,#37h,hh30
mov a,#32h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh30:
cjne a,#38h,hh31
mov a,#32h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh31:
cjne a,#39h,hh32
mov a,#32h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh32:
cjne a,#3ah,hh33
mov a,#32h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh33:
cjne a,#3bh,hh34
mov a,#32h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh34:
cjne a,#3ch,hh35
mov a,#32h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh35:
```

```
cjne a,#3dh,hh36
mov a,#32h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh36:
cjne a,#3eh,hh337
mov a,#32h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh37:
cjne a,#3fh,hh38
mov a,#33h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh38:
cjne a,#40h,hh39
mov a,#33h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh39:
cjne a,#41h,hh40
mov a,#33h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh40:
cjne a,#42h,hh41
mov a,#33h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh41:
cjne a,#43h,hh42
mov a,#33h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh42:
cjne a,#44h,hh43
mov a,#33h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh43:
cjne a,#45h,hh44
mov a,#33h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh44:
cjne a,#46h,hh45
mov a,#33h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh45:
cjne a,#47h,hh46
mov a,#33h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh46:
cjne a,#48h,hh47
mov a,#33h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh47:
cjne a,#49h,hh48
mov a,#33h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh48:
cjne a,#4ah,hh49
mov a,#33h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh49:
cjne a,#4bh,hh50
mov a,#33h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh50:
cjne a,#4ch,hh51
mov a,#33h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh51:
cjne a,#4dh,hh52
mov a,#33h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh52:
```

```
cjne a,#4eh,hh53
mov a,#33h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh53:
cjne a,#4fh,hh54
mov a,#33h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh54:
cjne a,#50h,hh55
mov a,#34h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh55:
cjne a,#51h,hh56
mov a,#34h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh56:
cjne a,#52h,hh57
mov a,#34h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh57:
cjne a,#53h,hh58
mov a,#34h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh58:
cjne a,#54h,hh59
mov a,#34h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh59:
cjne a,#55h,hh60
mov a,#34h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh60:
cjne a,#56h,hh61
mov a,#34h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh61:
cjne a,#57h,hh62
mov a,#34h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh62:
cjne a,#58h,hh63
mov a,#34h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh63:
cjne a,#59h,hh64
mov a,#34h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh64:
cjne a,#5ah,hh65
mov a,#34h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh65:
cjne a,#5bh,hh66
mov a,#34h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh66:
cjne a,#5ch,hh67
mov a,#34h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh67:
cjne a,#5dh,hh68
mov a,#34h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh68:
cjne a,#5eh,hh69
mov a,#34h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh69:
```

```
cjne a,#5fh,h70
mov a,#34h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh70:
cjne a,#60h,h71
mov a,#34h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh71:
cjne a,#61h,hh72
mov a,#34h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh72:
cjne a,#62h,hh73
mov a,#35h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh73:
cjne a,#63h,hh74
mov a,#35h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh74:
cjne a,#64h,hh75
mov a,#35h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh75:
cjne a,#65h,hh76
mov a,#35h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh76:
cjne a,#66h,hh77
mov a,#35h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh77:
cjne a,#67h,hh78
mov a,#35h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh78:
cjne a,#68h,hh79
mov a,#35h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh79:
cjne a,#69h,hh80
mov a,#35h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh80:
cjne a,#6ah,hh581
mov a,#35h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh81:
cjne a,#6bh,hh82
mov a,#35h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh82:
cjne a,#6ch,hh83
mov a,#35h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh83:
cjne a,#6dh,hh84
mov a,#35h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh84:
cjne a,#6eh,hh85
mov a,#35h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh85:
cjne a,#6fh,hh86
mov a,#35h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh86:
```

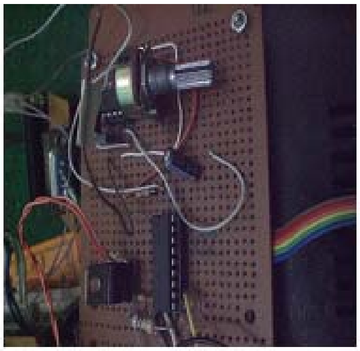
```
cjne a,#70h,hh87
mov a,#35h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh87:
cjne a,#71h,hh88
mov a,#35h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh88:
cjne a,#72h,hh89
mov a,#35h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh89:
cjne a,#73h,hh90
mov a,#36h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh90:
cjne a,#74h,hh91
mov a,#36h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh91:
cjne a,#75h,hh92
mov a,#36h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh92:
cjne a,#76h,hh93
mov a,#36h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh93:
cjne a,#77h,hh94
mov a,#36h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh94:
cjne a,#78h,hh95
mov a,#36h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh95:
cjne a,#79h,hh96
mov a,#36h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh96:
cjne a,#7ah,hh97
mov a,#36h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh97:
cjne a,#7bh,hh98
mov a,#36h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh98:
cjne a,#7ch,hh99
mov a,#36h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh99:
cjne a,#7dh,hh100
mov a,#36h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh100:
cjne a,#7eh,hh101
mov a,#36h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh101:
cjne a,#7fh,hh102
mov a,#36h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh102:
cjne a,#80h,hh103
mov a,#36h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh103:
```

```
cjne a,#81h,hh104
mov a,#36h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh104:
cjne a,#82h,hh105
mov a,#36h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh105:
cjne a,#83h,hh106
mov a,#36h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh106:
cjne a,#84h,hh107
mov a,#37h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh107:
cjne a,#85h,hh108
mov a,#37h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh108:
cjne a,#86h,hh109
mov a,#37h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh109:
cjne a,#87h,hh110
mov a,#37h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh110:
cjne a,#88h,hh111
mov a,#37h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh111:
cjne a,#89h,hh112
mov a,#37h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh112:
cjne a,#8ah,hh113
mov a,#37h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh113:
cjne a,#8bh,hh114
mov a,#37h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh114:
cjne a,#8ch,hh115
mov a,#37h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh115:
cjne a,#8dh,hh116
mov a,#37h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh116:
cjne a,#8eh,hh117
mov a,#37h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh117:
cjne a,#8fh,hh118
mov a,#37h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh118:
cjne a,#90h,hh119
mov a,#37h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh119:
cjne a,#91h,hh120
mov a,#37h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh120:
```

```
cjne a,#92h,hh121
mov a,#37h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh121:
cjne a,#93h,hh122
mov a,#37h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh122:
cjne a,#94h,hh123
mov a,#37h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh123:
cjne a,#95h,hh124
mov a,#37h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh124:
cjne a,#96h,hh125
mov a,#38h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh125:
cjne a,#97h,hh126
mov a,#38h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh126:
cjne a,#98h,hh127
mov a,#38h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh127:
cjne a,#99h,hh128
mov a,#38h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh128:
cjne a,#9ah,hh129
mov a,#38h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh129:
cjne a,#9bh,hh130
mov a,#38h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh130:
cjne a,#9ch,hh131
mov a,#38h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh131:
cjne a,#9dh,hh132
mov a,#38h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh132:
cjne a,#9eh,hh133
mov a,#38h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh133:
cjne a,#9fh,hh134
mov a,#38h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh134:
cjne a,#a0h,hh135
mov a,#38h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh135:
cjne a,#a1h,hh136
mov a,#38h
call display
call cursor
mov a,#36h
call display
call cursor
ret
hh136:
cjne a,#a2h,hh137
mov a,#38h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh137:
```

```
cjne a,#a3h,hh138
mov a,#38h
call display
call cursor
mov a,#37h
call display
call cursor
ret
hh138:
cjne a,#a4h,hh139
mov a,#38h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh139:
cjne a,#a5h,hh140
mov a,#38h
call display
call cursor
mov a,#38h
call display
call cursor
ret
hh140:
cjne a,#a6h,hh141
mov a,#38h
call display
call cursor
mov a,#39h
call display
call cursor
ret
hh141:
cjne a,#a7h,hh142
mov a,#39h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh142:
cjne a,#a8h,hh143
mov a,#39h
call display
call cursor
mov a,#30h
call display
call cursor
ret
hh143
cjne a,#a9h,hh144
mov a,#39h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh144:
cjne a,#aah,hh145
mov a,#39h
call display
call cursor
mov a,#31h
call display
call cursor
ret
hh145:
cjne a,#abh,hh146
mov a,#39h
call display
call cursor
mov a,#32h
call display
call cursor
ret
hh146:
cjne a,#ach,hh147
mov a,#39h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh147:
cjne a,#adh,hh148
mov a,#39h
call display
call cursor
mov a,#33h
call display
call cursor
ret
hh148:
cjne a,#aeh,hh149
mov a,#39h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh149:
cjne a,#afh,hh150
mov a,#39h
call display
call cursor
mov a,#34h
call display
call cursor
ret
hh1150:
cjne a,#b0h,hh151
mov a,#39h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh1151:
cjne a,#b1h,hh152
mov a,#39h
call display
call cursor
mov a,#35h
call display
call cursor
ret
hh152:
jmpdbase2
ret
end
```

LAMPIRAN C
FOTO ALAT

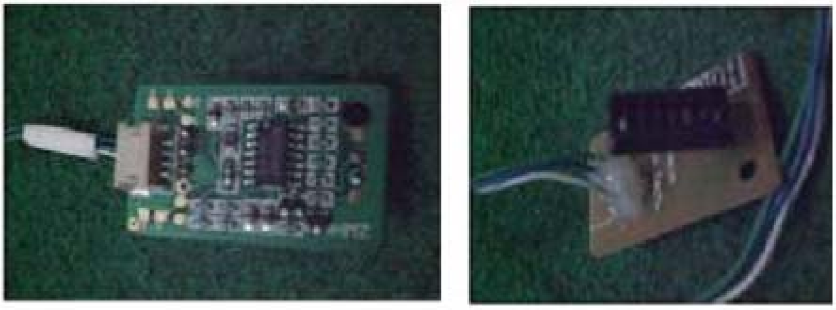


Lampiran C.1 Generator Frekuensi Audio

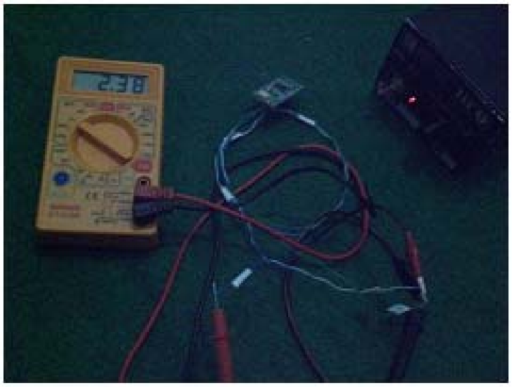
Lampiran C.2 Real Time Clock

Lampiran C.3 Relai

Lampiran C.4 Katup Selenoid



Lampiran C.5 Sensor Kelembaban



Lampiran C.6 Pengujian Sensor Kelembaban

Lampiran C.7 Sensor Temperatur

Lampiran C.8 Pengujian Sensor Temperatur

LAMPIRAN D
DATASHEET

National Semiconductor

November 1995

ADC0808/ADC0809 8-Bit µP Compatible A/D Converters with 8-Channel

Multiplexer General Description

The ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8-single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE® outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

Features

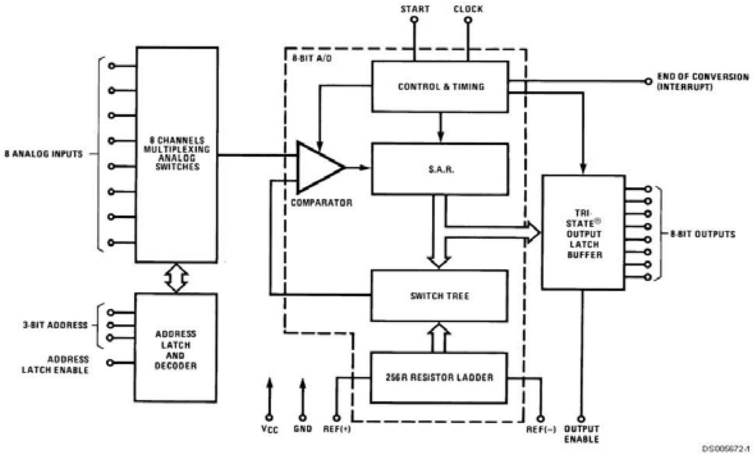
n Easy interface to all microprocessors n Operates ratiometrically or with 5 V_{DC} or analog span adjusted voltage reference n No zero or full-scale adjust required n 8-channel multiplexer with address logic n 0V to 5V input range with single 5V power supply n Outputs meet TTL voltage level specifications n Standard hermetic or molded 28-pin DIP package n 28-pin molded chip carrier package n ADC0808 equivalent to MM74C949 n ADC0809 equivalent to MM74C949-1

Key Specifications

n Resolution: 8 Bits n Total Unadjusted Error: ±½ LSB and ±1 LSB n Single Supply: 5 V_{DC} n Low Power: 15 mW n Conversion Time: 100 µs

ADC0808/ADC0809 8-Bit µP Compatible A/D Converters with 8-Channel Multiplexer

Block Diagram



DS005672-1

See Ordering Information

Absolute Maximu**m** Ratings (Notes 1, 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/

Distributors for availability and specifications.
Dual-In-Line Package (ceramic) 300˚C
Molded Chip Carrier Package
Vapor Phase (60 seconds) 215˚C

Infrared (15 seconds) 220˚C ESD Susceptibility (Note 8)

400V

Supply Voltage (V_{CC}) (Note 3) 6.5V

Voltage at Any Pin −0.3V to (V_{CC}+0.3V) **Operating Conditions** (Notes 1, 2)

Except Control Inputs Temperature Range (Note 1) $T_{MIN} \leq T_A \leq T_{MAX}$ Voltage at Control Inputs −0.3V to +15V ADC0808CJ

(START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)

Storage Temperature Range −65˚C to +150˚C

−55˚C≤T_A≤+125˚C ADC0808CCJ, ADC0808CCN, ADC0809CCN

Package Dissipation at T_A=25˚C 875 mW

Lead Temp. (Soldering, 10 seconds)

Dual-In-Line Package (plastic) 260˚C

−40˚C≤T_A≤+85˚C ADC0808CCV, ADC0809CCV −40˚C ≤ T_A ≤ +85˚C Range of V_{CC} (Note 1) 4.5 V_{DC} to 6.0 V_{DC}

Electrical Characteristics

Converter Specifications: V_{CC}=5V_{DC}=V_{REF+},$V_{REF(-)}$=GND, T_{MIN}≤T_A≤T_{MAX} and f_{CLK}=640 kHz unless otherwise stated.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	ADC0808					
	Total Unadjusted Error	25˚C			±½	LSB
	(Note 5)	T_{MIN} to T_{MAX}			±¾	LSB
	ADC0809					
	Total Unadjusted Error	0˚C to 70˚C			±1	LSB
	(Note 5)	T_{MIN} to T_{MAX}			±1¼	LSB
	Input Resistance	From Ref(+) to Ref(−)	1.0	2.5		kΩ
	Analog Input Voltage Range	(Note 4) V(+) or V(−)	GND−0.10		V_{CC}+0.10	V_{DC}
$V_{REF(+)}$	Voltage, Top of Ladder	Measured at Ref(+)		V_{CC}	V_{CC}+0.1	V
	Voltage, Center of Ladder		V_{CC}/2-0.1	V_{CC}/2	V_{CC}/2+0.1	V
$V_{REF(-)}$	Voltage, Bottom of Ladder	Measured at Ref(−)	−0.1	0		V
I_{IN}	Comparator Input Current	f_c=640 kHz, (Note 6)	−2	±0.5	2	μA

Electrical Characteristics

Digital Levels and DC Specifications: ADC0808CJ 4.5V≤V_{CC}≤5.5V, −55˚C≤T_A≤+125˚C unless otherwise noted ADC0808CCJ, ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, 4.75≤V_{CC}≤5.25V, −40˚C≤T_A≤+85˚C unless otherwise noted

Symbol
Parameter

Conditions

Min

Typ

Max

Units

ANALOG MULTIPLEXER

$I_{OFF(+)}$	OFF Channel Leakage Current	V_{CC}=5V, V_{IN}=5V,				
		T_A=25˚C		10	200	nA
		T_{MIN} to T_{MAX}			1.0	μA
$I_{OFF(-)}$	OFF Channel Leakage Current	V_{CC}=5V, V_{IN}=0,				
		T_A=25˚C	−200	−10		nA

		T_{MIN} to T_{MAX}	−1.0			μA

CONTROL INPUTS

$V_{IN(1)}$	Logical “1” Input Voltage		V_{CC}−1.5			V
$V_{IN(0)}$	Logical “0” Input Voltage				1.5	V
$I_{IN(1)}$	Logical “1” Input Current (The Control Inputs)	V_{IN} =15V			1.0	μA
$I_{IN(0)}$	Logical “0” Input Current (The Control Inputs)	V_{IN} =0	−1.0			μA
I_{CC}	Supply Current	f_{CLK} =640 kHz		0.3	3.0	mA

Electrical Characteristics (Continued)

Digital Levels and DC Specifications: ADC0808CJ 4.5V≤V_{CC}≤5.5V, −55˚C≤T_A≤+125˚C unless otherwise noted ADC0808CCJ, ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, 4.75≤V_{CC}≤5.25V, −40˚C≤T_A≤+85˚C unless otherwise noted

Symbol
Parameter

Conditions

Min

Typ

Max

Units

DATA OUTPUTS AND EOC (INTERRUPT)

$V_{OUT(1)}$	Logical “1” Output Voltage	I_O =−360 μA	V_{CC}−0.4			V
$V_{OUT(0)}$	Logical “0” Output Voltage	I_O =1.6 mA			0.45	V
$V_{OUT(0)}$	Logical “0” Output Voltage EOC	I_O =1.2 mA			0.45	V
I_{OUT}	TRI-STATE Output Current	V_O =5V V_O =0	−3		3	μA μA

Electrical Characteristics

Timing Specifications V_{CC}=$V_{REF(+)}$=5V, $V_{REF(-)}$=GND, t_r=t_f=20 ns and T_A=25˚C unless otherwise noted.

Symbol	Parameter	Conditions	MIn	Typ	Max	Units
t_{WS}	Minimum Start Pulse Width	(Figure 5)		100	200	ns
t_{WALE}	Minimum ALE Pulse Width	(Figure 5)		100	200	ns
t_s	Minimum Address Set-Up Time	(Figure 5)		25	50	ns
t_H	Minimum Address Hold Time	(Figure 5)		25	50	ns
t_D	Analog MUX Delay Time From ALE	R_S =0Ω (Figure5)		1	2.5	μS
t_{H1},t_{H0}	OE Control to Q Logic State	C_L =50 pF, R_L =10k (Figure8)		125	250	ns
t_{1H},t_{0H}	OE Control to Hi-Z	C_L =10 pF, R_L =10k (Figure8)		125	250	ns
t_c	Conversion Time	f_c =640 kHz, (Figure5) (Note 7)	90	100	116	μS
f_c	Clock Frequency		10	640	1280	kHz
t_{EOC}	EOC Delay Time	(Figure 5)	0		8+2 μS	Clock Periods

C_{IN}	Input Capacitance	At Control Inputs		10	15	pF
C_{OUT}	TRI-STATE Output Capacitance	At TRI-STATE Outputs		10	15	pF

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to GND, unless othewise specified.

Note 3: A zener diode exists, internally, from V_{CC} to GND and has a typical breakdown voltage of 7 V_{DC}.

Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CCn} supply. The spec allows 100 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute 0V_{DC} to 5V_{DC} input voltage range will therefore require a minimum supply voltage of 4.900 V_{DC} over temperature variations, initial tolerance and loading.

Note 5: Total unadjusted error includes offset, full-scale, linearity, and multiplexer errors. See Figure2. None of these A/Ds requires a zero or full-scale adjust. How ever, if an all zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example: 0.5V to 4.5V full-scale) the reference voltages can be adjusted to achieve this. See Figure 13.

Note 6: Comparator input current is a bias current into or out of the chopper stabilized comparator. The bias current varies directly with clock frequency and has little temperature dependence (Figure *NO TGT: fig NS0592*). See paragraph 4.0.

Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

Note 8: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Functional Description

Multiplexer. The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. Table1shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

TABLE 1.

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

The bottom resistor and the top resistor of the ladder network in Figure1are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached +½ LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. Figure2shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

The A/D converter’s successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 8 clock pulses after the rising edge of start conversion.

CONVERTER CHARACTERISTICS

The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

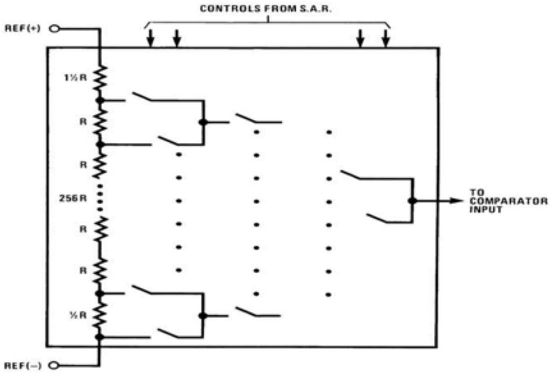
The 256R ladder network approach (Figure 1) was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed throught a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

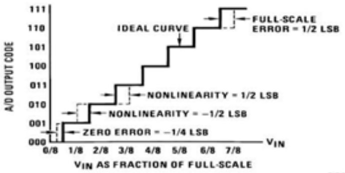
Figure 4 shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

Functional Description (Continued)



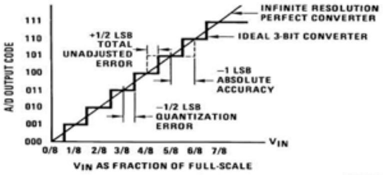
DS005672-2

FIGURE 1. Resistor Ladder and Switch Tree



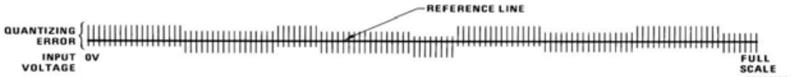
DS005672-13

FIGURE 2. 3-Bit A/D Transfer Curve



DS005672-14

FIGURE 3. 3-Bit A/D Absolute Accuracy Curve

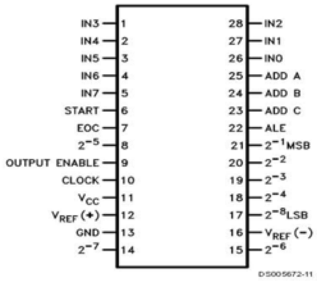


DS005672-15

FIGURE 4. Typical Error Curve

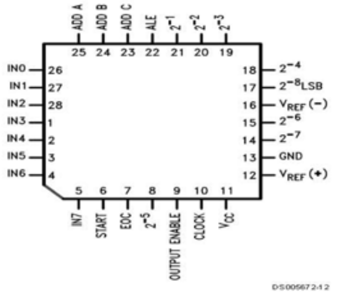
Connection Diagrams

Dual-In-Line Package



Order Number ADC0808CCN, ADC0809CCN, ADC0808CCJ or ADC0808CJ
See NS Package J28A or N28A

Molded Chip Carrier Package



Order Number ADC0808CCV or ADC0809CCV
See NS Package V28A

Timing Diagram

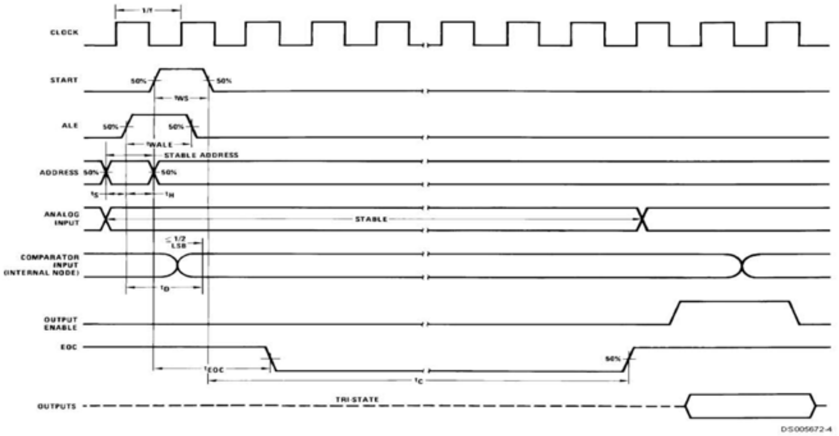


FIGURE 5.

Typical Performance Characteristics

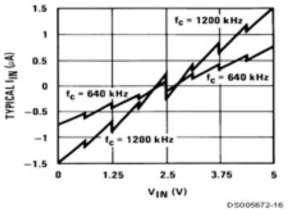


FIGURE 6. Comparator I_{IN} vs V_{IN} ($V_{CC}=V_{REF}=5V$)

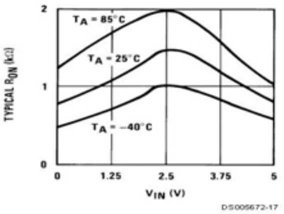


FIGURE 7. Multiplexer R_{ON} vs V_{IN} ($V_{CC}=V_{REF}=5V$)

TRI-STATE Test Circuits and Timing Diagrams

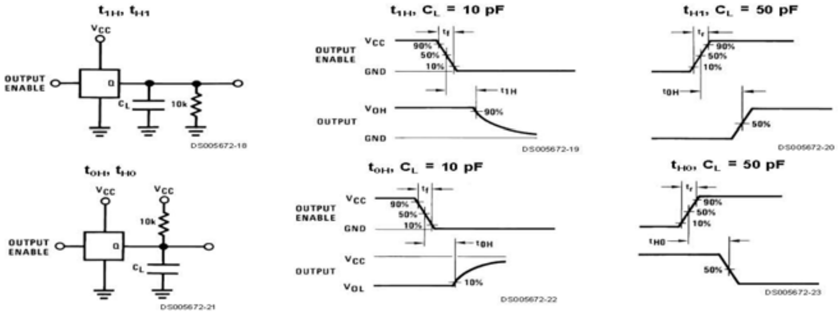


FIGURE 8.

Applications Information

OPERATION

1.0 RATIOMETRIC CONVERSION

The ADC0808, ADC0809 is designed as a complete Data Acquisition System (DAS) for ratiometric conversion systems. In ratiometric systems, the physical variable being measured is expressed as a percentage of full-scale which is not necessarily related to an absolute standard. The voltage input to the ADC0808 is expressed by the equation

$$\frac{V_{IN}}{V_{fs}-V_Z}=\frac{D_X}{D_{MAX}-D_{MIN}} \qquad (1)$$

V_{IN}=Input voltage into the ADC0808

V_{fs}=Full-scale voltage

V_Z=Zero voltage

D_X=Data point being measured

D_{MAX}=Maximum data limit

D_{MIN}=Minimum data limit

A good example of a ratiometric transducer is a potentiometer used as a position sensor. The position of the wiper is directly proportional to the output voltage which is a ratio of the full-scale voltage across it. Since the data is represented as a proportion of full-scale, reference requirements are greatly reduced, eliminating a large source of error and cost for many applications. A major advantage of the ADC0808, ADC0809 is that the input voltage range is equal to the supply range so the transducers can be connected directly across the supply and their outputs connected directly into the multiplexer inputs, (*Figure 9*).

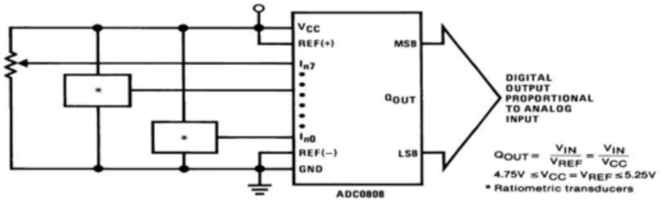
Ratiometric transducers such as potentiometers, strain gauges, thermistor bridges, pressure transducers, etc., are suitable for measuring proportional relationships; however, many types of measurements must be referred to an absolute standard such as voltage or current. This means a system reference must be used which relates the full-scale voltage to the standard volt. For example, if $V_{CC}=V_{REF}=5.12V$, then the full-scale range is divided into 256 standard steps. The smallest standard step is 1 LSB which is then 20 mV.

2.0 RESISTOR LADDER LIMITATIONS

The voltages from the resistor ladder are compared to the selected into 8 times in a conversion. These voltages are coupled to the comparator via an analog switch tree which is referenced to the supply. The voltages at the top, center and bottom of the ladder must be controlled to maintain proper operation.

The top of the ladder, Ref(+), should not be more positive than the supply, and the bottom of the ladder, Ref(−), should not be more negative than ground. The center of the ladder voltage must also be near the center of the supply because the analog switch tree changes from N-channel switches to P-channel switches. These limitations are automatically satisfied in ratiometric systems and can be easily met in ground referenced systems.

Figure 10 shows a ground referenced system with a separate supply and reference. In this system, the supply must be trimmed to match the reference voltage. For instance, if a 5.12V is used, the supply should be adjusted to the same voltage within 0.1V.



$$Q_{OUT} = \frac{V_{IN}}{V_{REF}} = \frac{V_{IN}}{V_{CC}}$$

$4.75V \leq V_{CC} = V_{REF} \leq 5.25V$

* Ratiometric transducers

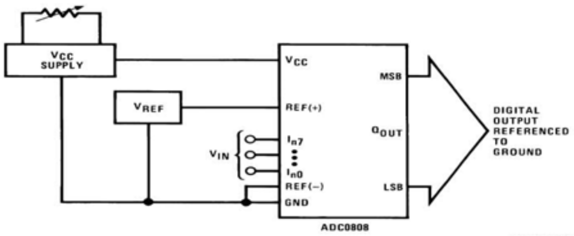
DS005672-7

FIGURE 9. Ratiometric Conversion System

The ADC0808 needs less than a milliamp of supply current so developing the supply from the reference is readily accomplished. In *Figure 11* a ground referenced system is shown which generates the supply from the reference. The buffer shown can be an op amp of sufficient drive to supply the milliamp of supply current and the desired bus drive, or if a capacitive bus is driven by the outputs a large capacitor will supply the transient supply current as seen in *Figure 12*. The LM301 is overcompensated to insure stability when loaded by the 10 µF output capacitor.

The top and bottom ladder voltages cannot exceed V_{CC} and ground, respectively, but they can be symmetrically less than V_{CC} and greater than ground. The center of the ladder voltage should always be near the center of the supply. The sensitivity of the converter can be increased, (i.e., size of the LSB steps decreased) by using a symmetrical reference system. In *Figure 13*, a 2.5V reference is symmetrically centered about $V_{CC}/2$ since the same current flows in identical resistors. This system with a 2.5V reference allows the LSB bit to be half the size of a 5V reference system.

Applications Information (Continued)

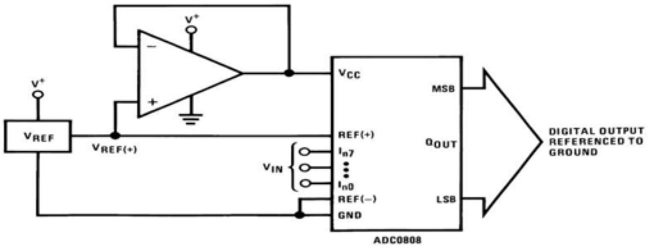


$$Q_{OUT} = \frac{V_{IN}}{V_{REF}}$$

$4.75V \leq V_{CC} = V_{REF} \leq 5.25V$

DS005672-24

FIGURE 10. Ground Referenced Conversion System Using Trimmed Supply



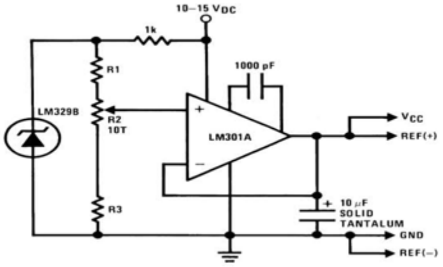
DS005672-25

$$Q_{OUT} = \frac{V_{IN}}{V_{REF}}$$

$4.75V \leq V_{CC} = V_{REF} \leq 5.25V$

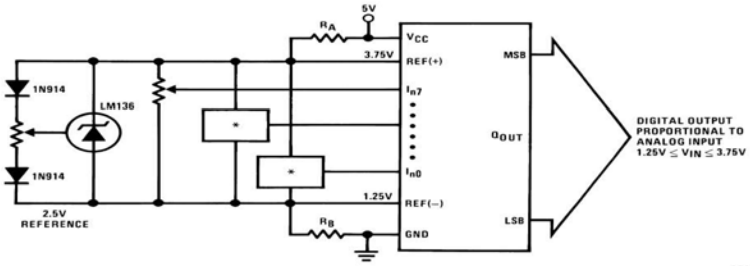
FIGURE 11. Ground Referenced Conversion System with Reference Generating V_{CC} Supply

Applications Information (Continued)



DS005672-26

FIGURE 12. Typical Reference and Supply Circuit



DS005672-27

$R_A = R_B$
*Ratiometric transducers

FIGURE 13. Symmetrically Centered Reference

3.0 CONVERTER EQUATIONS

The transition between adjacent codes N and N+1 is given by:

$$V_{IN} = \left\{ (V_{REF(+)} - V_{REF(-)}) \left[\frac{N}{256} + \frac{1}{512} \right] \pm V_{TUE} \right\} + V_{REF(-)} \qquad (2)$$

The center of an output code N is given by:

$$V_{IN} \left\{ (V_{REF(+)} - V_{REF(-)}) \left[\frac{N}{256} \right] \pm V_{TUE} \right\} + V_{REF(-)} \qquad (3)$$

The output code N for an arbitrary input are the integers within the range:

$$N = \frac{V_{IN} - V_{REF(-)}}{V_{REF(+)} - V_{REF(-)}} \times 256 \pm \text{Absolute Accuracy} \qquad (4)$$

where: V_{IN}=Voltage at comparator input
$V_{REF(+)}$=Voltage at Ref(+)
$V_{REF(-)}$=Voltage at Ref(−)
V_{TUE}=Total unadjusted error voltage (typically $V_{REF(+)}$+512)

Applications Information (Continued)

4.0 ANALOG COMPARATOR INPUTS

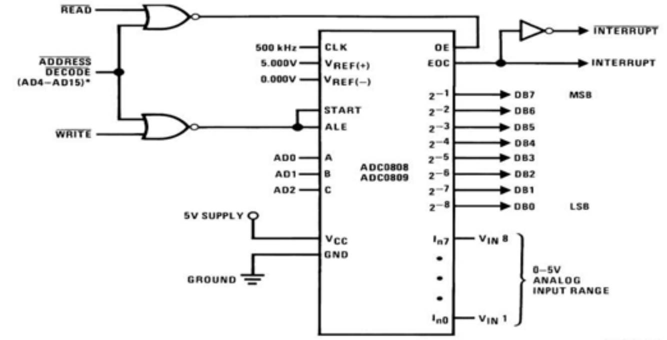
The dynamic comparator input current is caused by the periodic switching of on-chip stray capacitances. These are connected alternately to the output of the resistor ladder/switch tree network and to the comparator input as part of the operation of the chopper stabilized comparator.

The average value of the comparator input current varies directly with clock frequency and with V_{IN} as shown in *Figure 6*.

If no filter capacitors are used at the analog inputs and the signal source impedances are low, the comparator input current should not introduce converter errors, as the transient created by the capacitance discharge will die out before the comparator output is strobed.

If input filter capacitors are desired for noise reduction and signal conditioning they will tend to average out the dynamic comparator input current. It will then take on the characteristics of a DC bias current whose effect can be predicted conventionally.

Typical Application



DS005672-10

*Address latches needed for 8085 and SC/MP interfacing the ADC0808 to a microprocessor

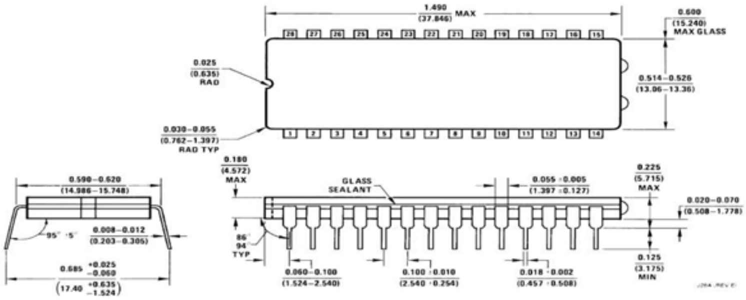
MICROPROCESSOR INTERFACE TABLE

PROCESSOR	$\overline{\text{READ}}$	$\overline{\text{WRITE}}$	INTERRUPT (COMMENT)
8080	$\overline{\text{MEMR}}$	$\overline{\text{MEMW}}$	INTR (Thru RST Circuit)
8085	$\overline{\text{RD}}$	$\overline{\text{WR}}$	INTR (Thru RST Circuit)
Z-80	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{INT}}$ (Thru RST Circuit, Mode 0)
SC/MP	NRDS	NWDS	SA (Thru Sense A)
6800	VMA•ϕ2•R/W	VMA•ϕ•$\overline{\text{R/W}}$	$\overline{\text{IRQA}}$ or $\overline{\text{IRQB}}$ (Thru PIA)

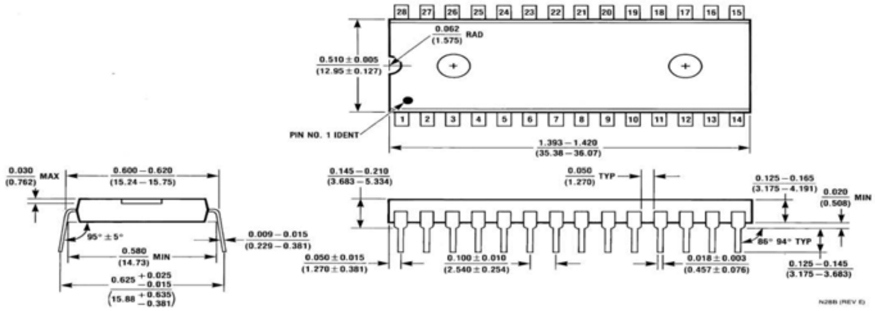
Ordering Information

TEMPERATURE RANGE		−40°C to +85°C			−55°C to +125°C
Error	±½ LSB Unadjusted	ADC0808CCN	ADC0808CCV	ADC0808CCJ	ADC0808CJ
	±1 LSB Unadjusted	ADC0809CCN	ADC0809CCV		
Package Outline		N28A Molded DIP	V28A Molded Chip Carrier	J28A Ceramic DIP	J28A Ceramic DIP

Physical Dimensions inches (millimeters) unless otherwise noted

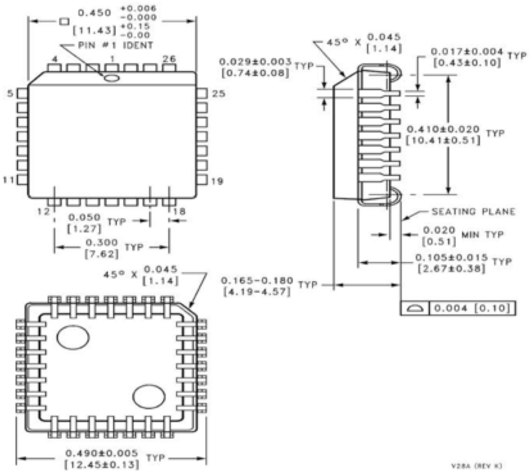


Ceramic Dual-In-Line Package (J)
Order Number ADC0808CCJ or ADC0808CJ
NS Package Number J28A



Molded Dual-In-Line Package (N)
Order Number ADC0808CCN or ADC0809CCN
NS Package Number N28B

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



Molded Chip Carrier (V)
Order Number ADC0808CCV or ADC0809CCV
NS Package Number V28A

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

National Semiconductor Corporation
Americas
Tel: 1-800-272-9959
Fax: 1-800-737-7018
Email: support@nsc.com

www.national.com

National Semiconductor Europe
Fax: +49 (0) 1 80-530 85 86
Email: europe.support@nsc.com
Deutsch Tel: +49 (0) 1 80-530 85 85
English Tel: +49 (0) 1 80-532 78 32
Français Tel: +49 (0) 1 80-532 93 58
Italiano Tel: +49 (0) 1 80-534 16 80

National Semiconductor Asia Pacific Customer Response Group
Tel: 65-2544466
Fax: 65-2504466
Email: sea.support@nsc.com

National Semiconductor Japan Ltd.
Tel: 81-3-5620-6175
Fax: 81-3-5620-6179