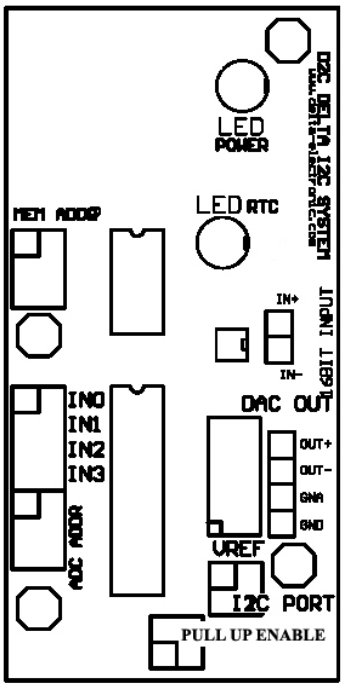
D2C Delta I2C System



DESKRIPSI

Modul ini merupakan antarmuka I2C yang memiliki Serial EEPROM, RTC, ADC dan DAC dalam 1 modul dan dapat dihubungkan dengan berbagai macam system mikrokontroler baik Atmel, Renesas, Motorola, PIC dll

Mem Addr:	Berfungsi untuk mengatur alamat Serial EEPROM 24C08.
16 bit Input:	Berfungsi untuk input analog ADC 16 bit (hanya pada versi ADC 16 bit include)
LED RTC:	Indikasi RTC aktif
LED Power:	Indikasi Power Supply 5V masuk dengan benar
DAC Out:	Output Analog D2C
IN0 … IN3:	4 kanal input analog ADC 8 bit
ADC Addr:	Berfungsi untuk mengatur alamat ADC
I2C Port:	Port I/O yang dihubungkan ke Sistem mikrokontroler
VREF:	Potensio pengatur tegangan referensi ADC 8 bit
Pull Up Enable:	Berfungsi untuk mengaktifkan resistor pull up, khusus untuk antarmuka dengan system mikrokontroler yang tidak

DELTA ELECTRONIC
www.delta-electronic.com

support internal pull up

Pengalamatan ADC 8bit

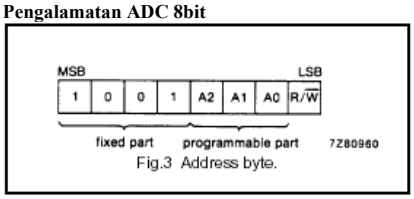
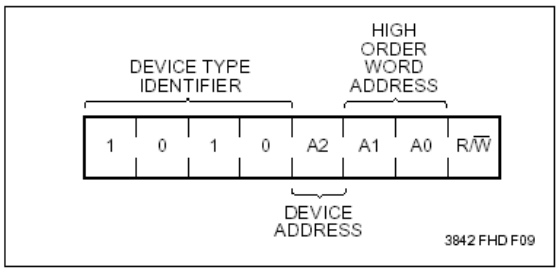


Fig.3 Address byte.

Gambar diambil dari datasheet PCF8591 Phillips Semiconductor

PCF8591 diaktifkan dengan mengirimkan Address Byte yang terdiri dari fix part dan programmable part di mana programmable part ini akan menyesuaikan dengan kondisi logika pada A0,A1 dan A2 di konektor ADC Addr.
Bit terakhir adalah merupakan mode akses di mana logika 1 adalah Read dan logika 0 adalah Write
Untuk lebih detail dapat dipelajari di datasheet PCF8591 www.delta-electronic.com/shop



Gambar diambil dari datasheet Atmel AT24C08

Pengalamatan Serial EEPROM AT24C08

Sama halnya dengan PCF8591, untuk mengaktifkan AT24c08 dilakukan dengan mengirimkan Address Byte yang terdiri dari fix part yaitu Device Type Identifier dalam hal ini 1010, Device Address untuk memilih Serial EEPROM yang digunakan apabila modul ini diparallel dengan Serial EEPROM lain, High Order Word Address dan mode akses R/W
Untuk lebih detailnya dapat dipelajari di datasheet AT24C08 www.delta-electronic.com/shop

Source Code Atmel AT8951 untuk PCF8591

```
;PROGRAM PENGAMBILAN DATA ADC 4 KANAL MULTI CHANNEL DAN
OUTPUT ANALOG
      .DATA
      Org   30H
```

```
CE          BIT   P1.0
CS          BIT   P1.1
DR1         BIT   P1.2
CLK1        BIT   P1.3
DATA        BIT   P1.4

      .CODE
      Org   00
      Ajmp  Start
      Org   03
      Reti
      Org   0BH
      Reti
      Org   13H
      Reti
      Org   1BH
      Reti
      Org   23H
      Reti

;Error di P0.2 Clr

Start:
      Lcall Init_Serial
      Mov   A,#00
Loop:
      Push  A
      Lcall SetConfigPCF8591
      Lcall BacaPCF8591
      Lcall Serial_Out
      Lcall Delay_1detik
      Pop   A
      Inc   A
      Cjne  A,#04H,Loop

      Mov   A,#01000000b
      Lcall SetAddressPCF8591
Loop2:
      Mov   A,#0              ;Set DAC Zero
      Lcall KirimDataI2C            ;
      Lcall Delay_1detik            ;
      Mov   A,#0FFH                 ;Set DAC Full
      Lcall KirimDataI2C            ;
      Lcall Delay_1detik            ;
      Ajmp  Loop2


BacaPCF8591:
      Lcall Buat_StartBit
      Mov   A,#91H
      Lcall KirimDataI2C
      Mov   B,#0
      Djnz  B,$
```

```
      Lcall BacaDataI2C
      Ret
;*****************************************************
***************
; Config PCF8591 : | 0  | AOE | AIP0 | AIP1 | 0  | AIF |
CH1 | CH0 |
; AOE: Analog Output Enable
; AIP0, AIP1: 00 = Multi channel
;           01 = Three Differential Inputs
;           10 = Single Ended and differential
;           11 = Two Differential Inputs
; AIF : Auto Increment Flag
; CH1, CH0 : 00 = CH0
;            01 = CH1
;            10 = CH2
;            11 = CH3

SetConfigPCF8591:
      Lcall SetAddressPCF8591
      Lcall Buat_StopBit            ;
      Push  B                 ;Delay Conversi ADC
      Mov   B,#90             ;
      Djnz  B,$               ;
      Pop   B                 ;
      Ret                     ;

SetAddressPCF8591
      Push  A
      Lcall Buat_StartBit           ;Kirim Device Address
      Mov   A,#90H                  ;
      Lcall KirimDataI2C            ;
      Pop   A                 ;
      Lcall KirimDataI2C            ;Kirim Config
      Ret

            Org   *

ASCII_Out:
      Acall Hex_ASCII2
      Acall Serial_Out
      XCH   A,B
      Acall Serial_Out
      Ret

Out_DPTR:
      Acall Enter_Code
      Mov   A,DPH
      Acall ASCII_Out
      Mov   A,DPL
      Acall ASCII_Out
      Ret

Enter_Code:
      Push  A
```

```
      Mov   A,#0DH
      Acall Serial_Out
      Mov   A,#0AH
      Acall Serial_Out
      Pop   A
      Ret

Init_Serial:
      MOV     SCON,#52H         ; Mode 1 Ren
;     MOV   TMOD,#20H         ; T0 Mode 2, T1 Mode 2
      Mov   A,TMOD
      Anl   A,#0FH
      Orl   A,#20H
      Mov   TMOD,A
      MOV   TH1,#0FDH         ; 9600 Baudrate
      Setb  TR1
      MOV     PCON,#0H         ;
      RET

Serial_Out:
      Clr   TI
      Mov   SBUF,A
TungguKirim:
      Jnb   TI,TungguKirim
      Ret

Serial_In:
      Clr   RI
TungguTerima:
      Jnb   RI,TungguTerima
      Mov   A,SBUF
      Ret

KirimPesan_Serial:
      Mov   A,#00H
      Movc  A,@A+DPTR
      Cjne  A,#0FH,KirimTerus
      Ret

KirimTerus:
      Acall Serial_Out
      Inc   DPTR
      Ajmp  KirimPesan_Serial

            Org   *
*********** ASCII TO HEXA ***************
ASCII_HEX
      Push  A                 ;Simpan Acc A di Stack
Pointer
      Mov   A,B               ;Konversikan Acc B ke Hexa
      Acall ASCII_Hex1        ;
      Mov   B,A               ;Simpan hasil di B
      Pop   A                 ;Konversikan Acc A ke Hexa
      Acall ASCII_Hex1        ;
```

```
      Swap  A                 ;Pindahkan hasil konversi
ke nibble
      Add   A,B               ;atas dan jumlahkan dengan
B
      Ret

ASCII_Hex1:
      Cjne  A,#03AH,*+3
      Jnc   Karakter
      Clr   C
      Subb  A,#30H
      Ret

Karakter:
      Clr   C
      Subb  A,#37H
      Ret
********************************
* ROUTINE KONVERSI HEXA KE ASCII
********************************
;-----------------------------
;Acc A = Angka -> tambah 30H
;Acc A <> Angka -> tambah 37H

            Org   *
Hex_ASCII2:
      Push  A                 ;Simpan Acc A ke SP
      Acall Hex_ASCII1        ;Konversi 1 nibble
      Mov   B,A               ;Simpan nibble bawah di Reg
B
      Pop   A                 ;Ambil Acc A dari SP
      Swap  A                 ;Tukar
      Acall Hex_ASCII1        ;Konversi 1 nibble
      Ret

Hex_ASCII1:
      Anl   A,#0FH                  ;Hapus Nibble Atas
      Cjne  A,#10,*+3         ;Acc A = 10 dan carry 0 ->
bukan angka
                              ;Acc A <>10 dan carry 0 ->
bukan angka
      Jnc   Bukan_Angka       ;Acc A <>10 dan carry 1 ->
tambah 30H
      Add   A,#30H
      Ret

Bukan_Angka:
      Add   A,#37H
      Ret

;*********************************************************
*******************
; SUBROUTINE MENULIS DAN MEMBACA DATA SERIAL EEPROM V2.0
; Revisi: 26/09/07
```

```
; - Baca dan Tulis Serial EEPROM dengan pengalamatan 8
bit dan 16 bit
; - Penulisan data Serial EEPROM secara Paging 32 byte
dengan pengalamatan 8 bit
;   dan 16 bit

      .DATA
Slave_AddrSEE     Ds    1
DataI2C     Ds    1
SDA   Bit   P2.0
SCL   Bit   P2.1

      .Code

***********
; SUBROUTINE TULIS SERIAL EEPROM 16 BIT
; - R7 = Device Address = A0h
; - B = Word Address 1
; - R6 = Word Address 2
; - DataI2C = Data yang ditulis
; Hasil:
; - Carry Flag Set bila No ACK
; - F0 set bila terjadi arbitrasi

Tulis_SEE16b:
      Lcall Siapkan16bAlamatSEE
      Jc    Wrong_Write
      Jb    F0,NotValid
      Mov   A,DataI2C
      Lcall KirimDataI2C
      Jc    Wrong_Write
      Jb    F0,NotValid
      Lcall Buat_StopBit
      Lcall DelaySEE
Notvalid:
      Ret


Wrong_Write:
      Lcall Buat_StopBit
      Clr   C
      Ret

***********
; SUBROUTINE TULIS SERIAL EEPROM 8 BIT
; - R7 = Device Address = A0h
; - B = Word Address
; - DataI2C = Data yang ditulis
; Hasil:
; - Carry Flag Set bila No ACK
; - F0 set bila terjadi arbitrasi

Tulis_SEE8b:
      Lcall Siapkan8bAlamatSEE
```

```
      Jc    Wrong_Write
      Jb    F0,NotValid
      Mov   A,DataI2C
      Lcall KirimDataI2C
      Jc    Wrong_Write
      Jb    F0,NotValid
      Lcall Buat_StopBit
      Lcall DelaySEE
      Ret

***********
; SUBROUTINE BACA EEPROM 16 BIT
; - R7 = Device Address = A0h
; - B = Word Address 1
; - R6 = Word Address 2
; Hasil:
; - Carry Flag Set bila No ACK
; - F0 set bila terjadi arbitrasi
; - Data yang dibaca di A

Baca_SEE16b:
      Lcall Siapkan16bAlamatSEE
      Jc    Wrong_Read
      Jb    F0,NoRead
      Lcall Buat_StartBit           ;Kirim Device Address
dengan
      Lcall ModeBacaSEE
      Jc    Wrong_read
      Jb    F0,NoRead
      Lcall BacaDataI2C
      Ret


***********
; SUBROUTINE BACA EEPROM 8 BIT
; - R7 = Device Address = A0h
; - B = Word Address 1
; Hasil:
; - Carry Flag Set bila No ACK
; - F0 set bila terjadi arbitrasi
; - Data yang dibaca di A

Baca_SEE8b:
      Clr   C
      Lcall Siapkan8bAlamatSEE
      Jc    Wrong_Read
      Jb    F0,NoRead
      Lcall Buat_StartBit           ;Kirim Device Address
dengan
      Lcall ModeBacaSEE
      Jc    Wrong_read
      Jb    F0,NoRead
      Lcall BacaDataI2C
      Ret
```

```
Wrong_Read:
      Lcall Buat_StopBit
      Clr   C
NoRead:
      Ret

;*********************************************************
************************
; SUBROUTINE PENULISAN DATA I2C SECARA PAGE UNTUK SERIAL
EEPROM PENGALAMATAN 8 BIT
; - Data I2C = Data yang ditulis
; - R7 = Device Address
; - B = Word Address
; Hasil:
; - Carry Flag set bila No ACK
; - F0 set bila terjadi Arbitrasi

PageSEE8bWrite
      Push  05H
      Lcall Siapkan8bAlamatSEE
      Mov   R5,#16
      Ajmp  PageSEEWrite


;*********************************************************
************************
; SUBROUTINE PENULISAN DATA I2C SECARA PAGE UNTUK SERIAL
EEPROM PENGALAMATAN 16 BIT
; - Data I2C = Data yang ditulis
; - R7 = Device Address
; - B = Word Address
; - R6 = Word Address 2
; Hasil:
; - Carry Flag set bila No ACK
; - F0 set bila terjadi Arbitrasi

PageSEE16bWrite:
      Push  05H
Loop2PageSEE16bWrite:
      Lcall Siapkan16bAlamatSEE
      Mov   R5,#31
PageSEEWrite:
      Mov   A,DataI2C
      Lcall KirimDataI2C
      Jc    WrongPWrite
      Jb    F0,NoPWrite
LoopPageSEE16bWrite:
      Mov   A,DataI2C
      Lcall KirimDataI2C
      Jc    WrongPWrite
      Jb    F0,NoPWrite
      Djnz  R5,LoopPageSEE16bWrite
```

```
WrongPWrite:
      Pop   05H
      Lcall Buat_StopBit
      Lcall DelaySEE
NoPWrite:
      Ret

KirimDeviceAddress:
      Lcall Buat_StartBit
      Push  A
      Mov   A,R7              ;Device Address
      Lcall KirimDataI2C            ;
      Pop   A                 ;
      Ret

Kirim1WordAddress:
      Push  A                 ;
      Mov   A,B               ;First Word Address
      Lcall KirimDataI2C            ;
      Pop   A                 ;
      Ret

Kirim2WordAddress:
      Push  A
      Mov   A,R6              ;Second Address
      Lcall KirimDataI2C            ;
      Pop   A
      Ret


Siapkan16bAlamatSEE:
      Lcall KirimDeviceAddress
      Jc    SalahTulisAlamat
      Jb    F0,SalahTulisAlamat
      Lcall Kirim1WordAddress
      Jc    SalahTulisAlamat
      Jb    F0,SalahTulisAlamat
      Lcall Kirim2WordAddress

SalahTulisAlamat:
      Ret

Siapkan8bAlamatSEE:
      Lcall KirimDeviceAddress
      Jc    SalahTulisAlamat
      Jb    F0,SalahTulisAlamat
      Lcall Kirim1WordAddress
      Ajmp  SalahTulisAlamat

ModeBacaSEE:
      Push  A                 ;
      Mov   A,R7
      Setb  A.0
      Mov   Slave_AddrSEE,A
```

```
      Pop   A
      Mov   A,Slave_AddrSEE
      Lcall KirimDataI2C
      Ret

DPHSEE8bit:
      Mov   R7,#0A0H
      MOv   A,DPH
      Anl   A,#00000011b            ;Ambil bit 0 dan 1
DPH
      Rl    A                 ;Geser kiri 1x

      Orl   A,R7              ;Copy bit 1 dan 2 ke R7
      Mov   R7,A              ;
      Mov   B,DPL
      Ret

DPTRSEE8bit:
      Acall DPHSEE8bit
      Acall Baca_SEE8b
      Ret

DPTRSEE16bit:
      Mov   R7,#0A0H
      Mov   B,DPH
      Mov   R6,DPL
      Acall Baca_SEE16b
      Ret

TulisDPTRSEE8b:
      Acall DPHSEE8bit
      Acall Tulis_SEE8b
      Ret

TulisDPTRSEE16b:
      Mov   R7,#0A0H
      Mov   B,DPH
      Mov   R6,DPL
      Lcall Tulis_SEE16b
      Ret

DelaySEE:
      Push  B
      Mov   B,#2
LoopDelaySEE:
      Push  B
      Mov   B,#00
      Djnz  B,$
      Pop   B
      Djnz  B,LoopDelaySEE
      Pop   B
      Ret
```

```
;*******************************************************
*******************
; I2C SUBROUTINES AS MASTER
;
; Baca Data I2C
; - Subroutine untuk membaca data dari I2C
; - Hasil tersimpan di A
; - Bila Carry Flag set, maka proses pembacaan No ACK

BacaDataI2C:
      Push  B
      Mov   B,#08H
      Clr   A
LoopBacaSEE16b:
      Push  B
      Rl    A
      Setb  SDA
      Setb  SCL
      Clr   C
      Mov   C,SDA
      Mov   A.0,C
      Clr   SCL
      Pop   B
      Djnz  B,LoopBacaSEE16b
      Lcall Ambil_Ack
;     Lcall Buat_StopBit
      Pop   B
      Ret

;*******************
;Kirim Data I2C
; - Subroutine menulis data I2C
; - Data yang ditulis di A
; - F0 set bila terjadi kondisi arbitrasi
; - Carry Flag set bila terjadi No ACK

KirimDataI2C:
      Clr   F0
      Push  B
      Mov   B,#8
Send8_bitloop
      Rlc   A
      Mov   SDA,C
      Jnc   Tidak1
      Jb    SDA,TidakError
      Setb  F0

TidakError:
Tidak1:
      Lcall PulseI2C
      Djnz  B,Send8_bitloop
      Pop   B
      Clr   C
      Lcall Ambil_Ack
```

```
      Ret

;************************
; PULSE I2C
; - Subroutine memberikan sinyal clock di komponen I2C
; - Signal clock akan disinkronkan dengan master lain
yang mengakses bus ini
;   dengan menunggu kondisi SCL logika 1

PulseI2C:
      Push  B
      Setb  SCL
      Jnb   SCL,$
      Clr   SCL
      Pop   B
      Ret


Buat_StartBit:
      Setb  SDA
      Setb  SCL
      Jnb   SDA,$             ;Tunggu jalur free
      Jnb   SCL,$             ;
      Clr   SDA               ;Start data
      Clr   SCL               ;
      Ret

Buat_StopBit:
      Clr   SDA
      Setb  SCL
      Setb  SDA
      Clr   SCL
      Ret

Ambil_Ack:
      Clr   C
      Setb  SDA
      Setb  SCL
      Mov   C,SDA
      Clr   SCL
;     Jnb   SDA,$
      Ret

      Org   *
      .data
Counter_5ms Ds    1           ;Jumlah delay 5mS yg
terjadi

      .CODE
      Org   *
************
* Delay 5 mS sebanyak 200 x

Delay_1detik:
```

```
      Mov   Counter_5mS,#0200

Tunggu_1detik:
      Acall Delay_5mS
      Djnz  Counter_5mS,Tunggu_1detik
      Ret

***********
* Delay 5 mS sebanyak 100 x

Delay_500mS:
      Mov   Counter_5mS,#0100

Tunggu_500mS:
      Acall Delay_5mS
      Djnz  Counter_5mS,Tunggu_500mS
      Ret

************
* Delay 5 mS sebanyak 20 x
Delay_100mS:
      Mov   Counter_5mS,#020
Tunggu_100mS:
      Acall Delay_5mS
      Djnz  Counter_5mS,Tunggu_100mS
      Ret

************
* Delay 5 mS sebanyak 15 x
Delay_75mS:
      Mov   Counter_5mS,#015

Tunggu_75mS:
      Acall Delay_5mS
      Djnz  Counter_5mS,Tunggu_75mS
      Ret

**************
* Delay ini bekerja hanya pada crystal 11.0592 MHz

Delay_5mS
      Push  TMOD
      Mov   TMOD,#21H         ;Timer Mode 16 bit counter
      Mov   TH0,#0EDH
      Mov   TL0,#0FFH
      Setb  TR0
Tunggu_5mS:
      Jbc   TF0,Sudah_5mS
      Ajmp  Tunggu_5mS

Sudah_5mS:
      Clr   TR0
      Pop   TMOD
      Ret
```

DELTA ELECTRONIC
www.delta-electronic.com