

Pemakaian Bahasa C pada DST-51

Dengan adanya Embedded C, Bahasa C dapat digunakan dalam mikrokontroler, khususnya dalam hal ini pada keluarga MCS51 (AT89C51) yang menjadi basis dari DST-51. Untuk memasukkan program yang ditulis dalam Embedded C ke dalam DST-51, maka tahap-tahap yang dilakukan adalah sebagai berikut:

- Tuliskan program dalam Embedded C dengan menggunakan Text Editor
- Compile program yang anda tulis kedalam bentuk hexa
- Download file hexa dengan menggunakan Software Downloader DST-51

Untuk menuliskan program dalam Embedded C dapat digunakan Software-software Text Editor seperti Notepad yang akan menyimpan dalam bentuk file teks, sedangkan untuk compile dapat digunakan SDCC di mana software ini bersifat free.

Instalasi SDCC

Instalasi SDCC ke sistem windows dapat dilakukan dengan langkah-langkah berikut:

- Extract file zip dengan menggunakan Winzip, gunzip atau software ZIP yang lain.
- Copy file-file yang telah di-extract ke direktori yang anda inginkan sebagai direktori dari software tersebut, contoh: C:\Programs\
- Proses compile dilakukan dengan menuliskan lokasi dari direktori lib dan include, contoh untuk meng-compile program yang tersimpan dalam file test.c: `sdcc -I c:\Programs\SDCC\include -L c:\Programs\SDCC\lib\small test.c`

Option-option SDCC untuk DST-51

Agar file hexa hasil compile SDCC dapat digunakan oleh DST-51, maka terdapat beberapa option yang harus diatur sebagai berikut:

```
Sdcc -I c:\Programs\SDCC\include -L c:\Programs\SDCC\lib\small --code-loc 0x2000 -- data-loc 0x50 c:\Programs\SDCC\Project\testc.c
```

--code-loc 0x2000

Option ini berfungsi untuk menempatkan *source code* dalam file hexa di alamat 2000H yang merupakan alamat awal dari memori DST-51

--data-loc 0x50

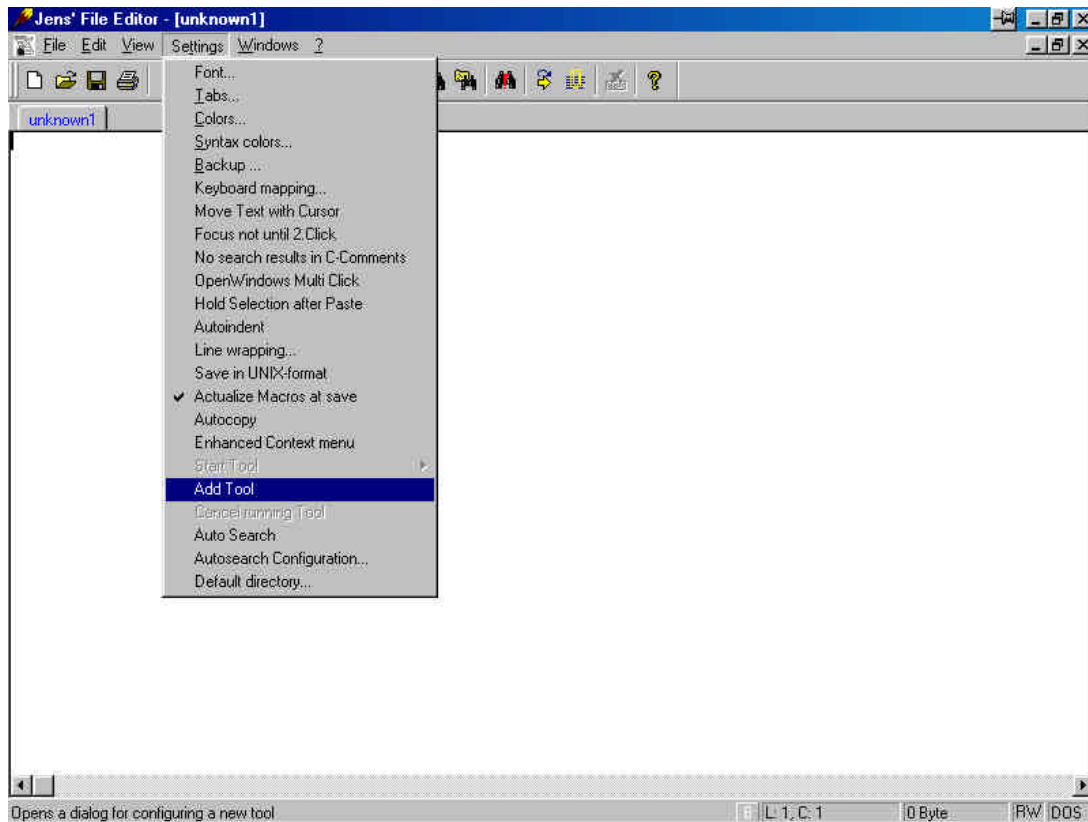
Option ini berfungsi untuk menempatkan variabel-variabel data dalam alamat 50H dari RAM Internal AT89C51 di mana alamat tersebut tidak digunakan oleh Program Monitor DST-51 sehingga dapat dengan bebas digunakan.

Menghubungkan SDCC ke JFE

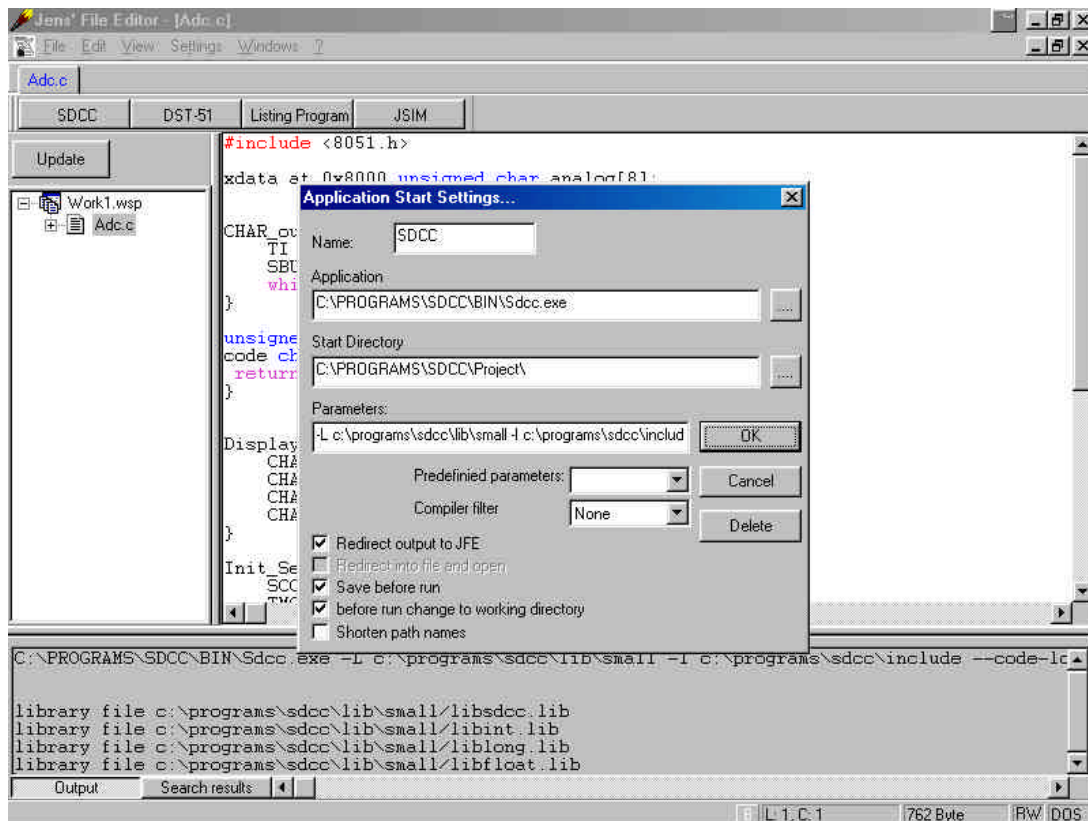
JFE adalah sebuah software yang dapat menghubungkan beberapa software ke dalamnya, sehingga software SDCC dapat terhubung dengan software Downloader DST-51. JFE juga mempunyai editor yang cukup baik sehingga listing program C dapat ditulis dan diedit dengan baik di dalamnya.

Membuka Ruang Kerja Baru (*New Workspace*)

- Klik File, kemudian New
- Klik Setting, kemudian pilih Add Tool

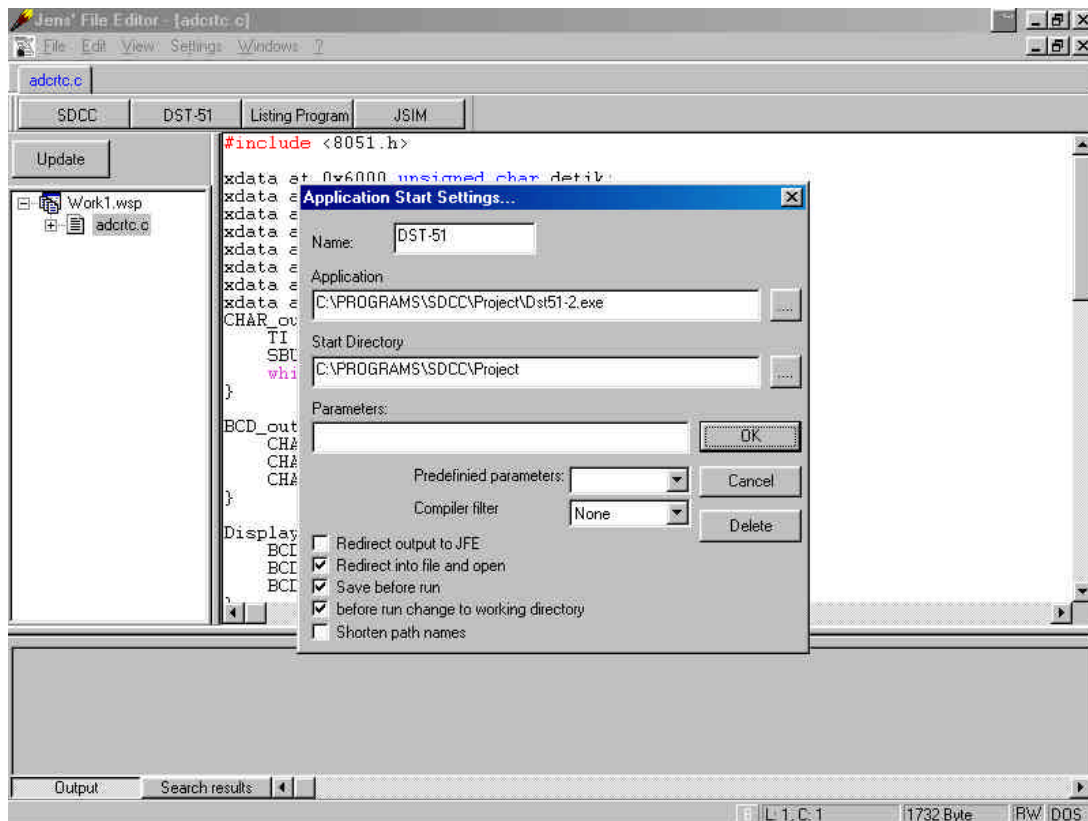


- Isi Bagian Name dengan SDCC
- Isi Bagian Application dengan software SDCC.exe lengkap dengan direktorinya
- Isi Bagian Directory dengan direktori kerja SDCC (direktori tempat file *.c yang akan dirancang diletakkan)
- Isi Bagian Parameter dengan `-I c:\Programs\SDCC\include -L c:\Programs\SDCC\lib\small --code-loc 0x2000 -- data-loc 0x50 --debug` yang merupakan option SDCC untuk DST-51
- Pilihlah \$FILE pada Predefined parameters



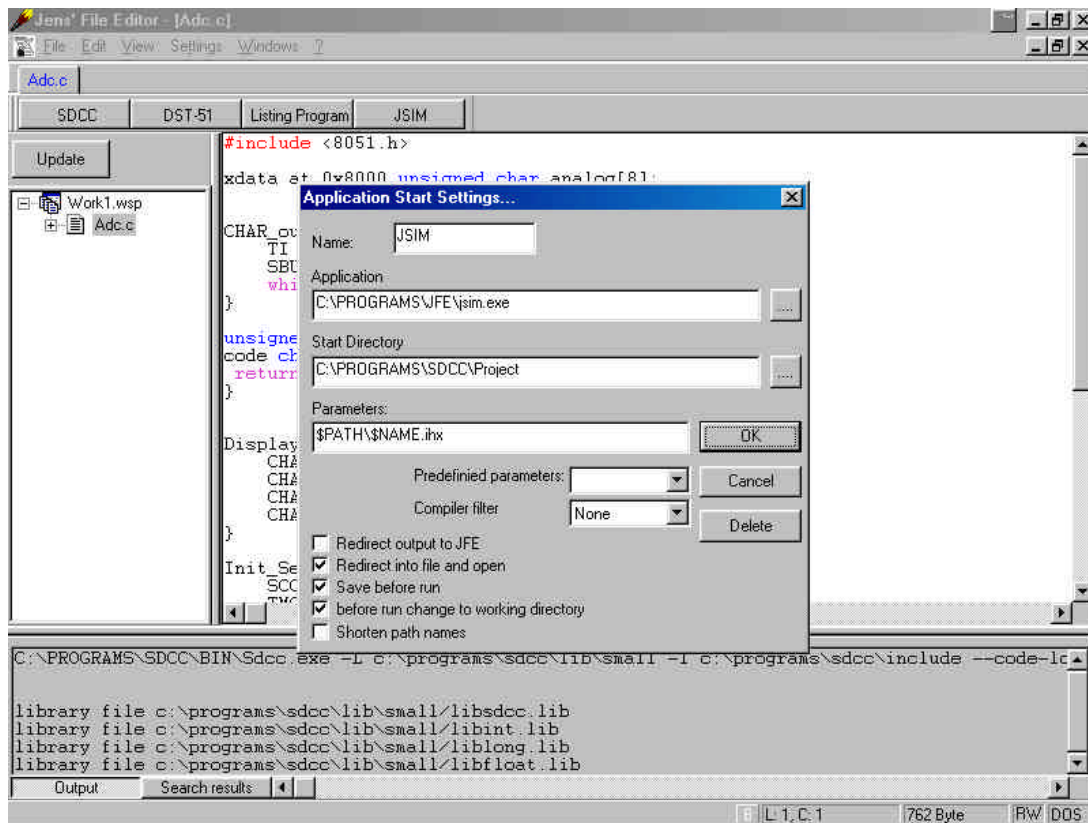
Menghubungkan Software Downloader DST-51 ke JFE

- Klik Setting, kemudian Add Tool
- Isi Bagian Name dengan DST-51
- Isi Bagian Application dengan software DST51.exe lengkap dengan direktorinya
- Isi Bagian Directory dengan direktori kerja SDCC (direktori tempat file *.ihx yang merupakan file hexa hasil compile SDCC terletak)



Menghubungkan JSIM ke JFE

- Klik Setting, kemudian Add Tool
- Isi Bagian Name dengan JSIM
- Isi Bagian Application dengan software JSIM.exe lengkap dengan direktorinya
- Isi Bagian Directory dengan direktori kerja SDCC (direktori tempat file *.c yang akan dirancang diletakkan)
- Isi Parameter dengan \$PATH\\$.ihx agar tool ini akan langsung membuka file dengan ekstensi ihx, nama dan direktori yang sama dengan file *.c yang ditulis.
- Proses Debug dari JSIM hanya dapat digunakan bila option -- debug pada SDCC diaktifkan.



Contoh Program

1. Program Penampil Jam Digital melalui Port Serial dengan Kit RTC-1287

Bahasa C:

```
#include <8051.h>
```

```
xdata at 0x6000 unsigned char detik;
xdata at 0x6002 unsigned char menit;
xdata at 0x6004 unsigned char jam;
xdata at 0x600A unsigned char reg_A;
xdata at 0x600B unsigned char reg_B;
xdata at 0x600C unsigned char reg_C;
xdata at 0x600D unsigned char reg_D;
```

```
CHAR_out (unsigned char i) {
    TI = 0;
    SBUF = i;
    while (!TI);
}
```

```
BCD_out(unsigned char i, unsigned char j) {
    CHAR_out((i / 16)+0x30);
    CHAR_out((i % 16)+0x30);
    CHAR_out(j);
}
```

```
DisplayWaktu () {
    BCD_out(jam,':');
```

DELTA ELECTRONIC

<http://www.delta-electronic.com>

```

BCD_out(menit, ':');
BCD_out(detik, 0x0D);
}

```

```

Init_Serial () {
    SCON = 0x52; // Mode 1 Ren
    TMOD = 0x20; // T0 Mode 2, T1 Mode 2
    TH1 = -3; // 9600 Baudrate
    TR1 = 1;
}

```

```

main () {
    char i;

```

```

    Init_Serial ();

```

```

    reg_A = 0x20; // Aktifkan osilator
    reg_B = 0x12; // Update Interrupt aktif, Mode 24 Jam; Output BCD Data

```

```

    while (1) {
        i = reg_C & 0x80; // Periksa Flag IRQ

        if (i) { // ada Update Interrupt
            reg_C = reg_C & 0x7F; // non-aktifkan ada Update Interrupt
            DisplayWaktu(); // Displaykan waktu
        }
    }
}

```

Bahasa Assembly:

```

Alamat_RTC EQU 6000H
Alamat_ADC EQU 8000H
Detik EQU 00H
Menit EQU 02H
Jam EQU 04H
RegA EQU 0AH
RegB EQU 0BH
RegC EQU 0CH
RegD EQU 0DH

```

```

    Org 2000H ;Reset Vector
    Ajmp Start ;
    Org 2003H ;External Interrupt 0 Vector
    Reti ;
    Org 200BH ;Timer 0 Interrupt Vector
    Reti ;
    Org 2013H ;External Interrupt 1 Vector
    Reti ;
    Org 201BH ;Timer 1 Interrupt Vector
    Reti ;
    Org 2023H ;Serial Interrupt Vector
    Reti ;

```

```

Start:

```

```

Mov    DPTR,#Alamat_RTC+RegA    ;Aktifkan osilator
Mov    A,#20H                    ;
Movx   @DPTR,A                    ;
Inc    DPTR                        ;Update Interrupt aktif, Mode 24 Jam
Mov    A,#12H                    ;Output BCD Data
Movx   @DPTR,A                    ;

Loop:
Mov    DPTR,#Alamat_RTC+RegC    ;Periksa Flag IRQ
Movx   A,@DPTR                    ;
ANL    A,#080H                    ;
Jnb    A.7,Loop                  ;Tidak ada flag, loop
Mov    DPTR,#Alamat_RTC        ;Displaykan waktu
Acall  Display_Waktu            ;
Ajmp   Loop

Display_Waktu:
Mov    DPTR,#Alamat_RTC+Jam
Movx   A,@DPTR
Acall  ASCII_Out
Mov    A,#':'
Acall  Serial_Out
Mov    DPTR,#Alamat_RTC+Menit
Movx   A,@DPTR
Acall  ASCII_Out
Mov    A,#':'
Acall  Serial_Out
Mov    DPTR,#Alamat_RTC+Detik
Movx   A,@DPTR
Acall  ASCII_Out
Ret

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

Init_Serial:
MOV    SCON,#52H                ; Mode 1 Ren
MOV    TMOD,#20H                ; T0 Mode 2, T1 Mode 2
MOV    TH1,#0FDH                ; 9600 Baudrate
MOV    TCON,#040H               ; T1 On, T0 Off
MOV    PCON,#00H                ;
RET

Serial_Out:
CLR    TI
MOV    SBUF,A
JNB    TI,*
RET

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

```

DELTA ELECTRONIC

<http://www.delta-electronic.com>

```

Ret

Hex_ASCII1:
Anl    A,#0FH           ;Hapus Nibble Atas
Add    A,#30H          ;Tambah 30
Ret

KirimPesan_Serial:
Movx   A,@DPTR
Acall  Serial_Out
Jnz    KirimTerus_Serial
Ret

KirimTerus_Serial:
Inc    DPTR
Ajmp   KirimPesan_Serial

```

2. Program Penampil 8 Channel ADC melalui Port Serial dengan Kit AD-0809

Bahasa C:

```

#include <AT89x51.h>

xdata at 0x8000 unsigned char analog[8];

CHAR_out (unsigned char i) {
    TI = 0;
    SBUF = i;
    while (!TI);
}

unsigned char Hex(unsigned char i) {
code char h[16] = "0123456789ABCDEF";
return h[i];
}

DisplayTegangan (unsigned char i) {
    CHAR_out(' ');
    CHAR_out(Hex(i / 16));
    CHAR_out(Hex(i % 16));
}

Init_Serial () {
    SCON = 0x52; // Mode 1 Ren
    TMOD = 0x20; // T0 Mode 2, T1 Mode 2
    TH1 = -3; // 9600 Baudrate
    TR1 = 1;
}

main () {
    char i;

```

```

Init_Serial ();

for (i=0; i!=8; i++) {
    analog[i] = 0;
    while (INT0);

    DisplayTegangan(analog[i]);    // Displaykan
}

while (1);
}

```

Bahasa Assembly:

```

.CODE

    Org    2000H
    Ajmp   Start
    Org    2003H        ;External INT0
    Reti
    Org    200BH        ;Timer 0 Interrupt
    Reti
    Org    2013H        ;External INT1
    Reti
    Org    201BH        ;Timer 1 Interrupt
    Reti
    Org    2023H        ;Serial I/O Interrupt
    Reti

Start:
    Mov    P1,#02H
    Mov    Dptr,#08000H    ;ADC pada alamat 800XH, CS4 pada DST-51
    Lcall  Init_Serial ;Inisial Serial Port

Loop:
    Acall  ADC            ;Akses ADC
    Acall  Delay          ;
    Mov    A,#' '        ;
    Lcall  Serial_Out    ;Kirim Spasi
    Inc    Dptr           ;Tunjuk ke channel berikut

    Mov    A,DPL         ;
    Cjne  A,#08H,Loop   ;Channel 8? Belum Loop
    Ajmp  *

ADC:
    Mov    A,#00H        ;Start Conversion
    Movx   @Dptr,A       ;
    Jb     INT0,*        ;Tunggu EOC
    Acall  Delay
    Mov    A,#00H        ;Read Convert Data
    Movc   A,@A+Dptr     ;
    Lcall  ASCII_Out     ;Send to Serial Out
    Ret

Delay:
    Push  B
    Mov   B,#0FFH
    Djnz B,*

```

```

    Pop    B
    Ret

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

Init_Serial:
    MOV    SCON,#52H          ; Mode 1 Ren
    MOV    TMOD,#20H         ; T0 Mode 2, T1 Mode 2
    MOV    TH1,#0FDH        ; 9600 Baudrate
    MOV    TCON,#040H       ; T1 On, T0 Off
    MOV    PCON,#00H        ;
    RET

Serial_Out:
    CLR    TI
    MOV    SBUF,A
    JNB    TI,*
    RET

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

```

3. Program Pemutar Nomor Telephone dengan Kit DF-88

Bahasa C:

```

#include <AT89x51.h>
xdata at 0x8000 unsigned char RegisterData;
xdata at 0x8100 unsigned char RegisterKontrol;
xdata at 0x8100 unsigned char RegisterStatus;

```

```

InitMT8888(){
    char i;
    i=RegisterStatus;
    RegisterKontrol = 0x00;
    RegisterKontrol = 0x00;
}

```

DELTA ELECTRONIC
<http://www.delta-electronic.com>

```

    RegisterKontrol = 0x08;
    RegisterKontrol = 0x00;
    RegisterKontrol = 0x0d;
    RegisterKontrol = 0x00;
}

KirimDTMF(unsigned char i){
    RegisterData = i;
}

Delay5mS(unsigned char j){
    unsigned char i;
    i=0;
    TMOD = 0x21;
    for (i = 0; i < j; i++) {
        TH0 = 0xEE;
        TL0 = 0x00;
        TR0 = 1;
        while (!TF0);
        TF0 = 0;
        TR0 = 0;
    }
}

code char nomor[8] = "5020210";
main(){
    P1_1=0;
    Delay5mS(100);
    int i;
    InitMT8888();
    i = 0;
    while (nomor[i] !=0x00) {
        if (nomor[i] != 0x30)
            KirimDTMF(nomor[i]-0x30);
        else KirimDTMF(nomor[i]-0x26);
        i++;
        Delay5mS (40);
    }
    while (1);
}

```

Bahasa Assembly:

```

.CODE

Alamat_DF88      EQU      8000H

    Org    2000H          ;Reset Vector
    Ajmp   Start         ;
    Org    2003H          ;External Interrupt 0 Vector
    Reti   ;
    Org    200BH          ;Timer 0 Interrupt Vector
    Reti   ;

```

DELTA ELECTRONIC

<http://www.delta-electronic.com>

```

    Org    2013H                ;External Interrupt 1 Vector
    Reti
    Org    201BH                ;Timer 1 Interrupt Vector
    Reti
    Org    2023H                ;Serial Interrupt Vector
    Reti

Start:
    Mov    DPTR,#Alamat_DF88 ;Reset_MT8888
    Lcall  Init_MT8888        ;
    Clr    P1.1                ;Off Hook
    Lcall  Delay_500mS        ;Delay 1/2 detik
    Mov    DPTR,#Nomor_Telephone ;Data Pointer menuju ke alamat
tabel
                                ;nomor telephone

LoopPutar_Nomor:
    Movx   A,@DPTR            ;Ambil nomor telephone yang
ditunjuk
                                ;oleh Data Pointer
    Cjne   A,#0FH,Putar_Nomor ;Data <> 0F, putar nomor
berikutnya
    Ajmp   *                    ;

Putar_Nomor:
    Push   DPH                ;Simpan DPTR ke SP
    Push   DPL                ;
    Mov    DPTR,#Alamat_DF88 ;DPTR = Alamat DF88
    Lcall  Kirim_DTMF         ;Konversi nomor telephone ke DTMF
    Lcall  Delay_100mS        ;Delay 200mS
    Lcall  Delay_100mS
    Pop    DPL                ;Ambil DPTR dari SP
    Pop    DPH                ;
    Inc    DPTR               ;Tunjuk ke tabel nomor telephone
berikut
    Ajmp   LoopPutar_Nomor    ;

Init_MT8888:
    INC    DPH
    MOVX   A,@DPTR
    CLR    A
    MOVX   @DPTR,A
    MOVX   @DPTR,A
    MOV    A,#008h
    MOVX   @DPTR,A
    CLR    A
    MOVX   @DPTR,A
    MOVX   A,@DPTR
    MOV    A,#00Dh
    MOVX   @DPTR,A
    MOV    A,#000h
    MOVX   @DPTR,A
    RET

Kirim_DTMF:
    Anl    A,#0FH              ;Bila terjadi angka 0 maka
    Cjne   A,#00H,BukanNol_DTMF ;ubah menjadi 0AH atau 10 desimal
    Mov    A,#0AH              ;

BukanNol_DTMF:
    MOVX   @DPTR,A            ;Kirim_DTMF

```

DELTA ELECTRONIC

<http://www.delta-electronic.com>

```

        INC     DPH                ;
        MOVX   A,@DPTR            ;Baca Status Register
        RET

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

Nomor_Telephone:
        DB    '5020210',0FH

```

Program Demo

Bahasa C:

DELTA ELECTRONIC

<http://www.delta-electronic.com>

```

#include <8051.h>

xdata at 0x6000 unsigned char detik;
xdata at 0x6002 unsigned char menit;
xdata at 0x6004 unsigned char jam;
xdata at 0x600A unsigned char reg_A;
xdata at 0x600B unsigned char reg_B;
xdata at 0x600C unsigned char reg_C;
xdata at 0x600D unsigned char reg_D;
xdata at 0x8000 unsigned char analog[8];
CHAR_out (unsigned char i) {
    TI = 0;
    SBUF = i;
    while (!TI);
}

BCD_out(unsigned char i, unsigned char j) {
    CHAR_out((i / 16)+0x30);
    CHAR_out((i % 16)+0x30);
    CHAR_out(j);
}

DisplayWaktu () {
    BCD_out(jam, ':');
    BCD_out(menit, ':');
    BCD_out(detik, ' ');
}

unsigned char Hex(unsigned char i) {
code char h[16] = "0123456789ABCDEF";
return h[i];
}

DisplayTegangan (unsigned char i) {
    CHAR_out(':');
    CHAR_out(Hex(i / 16));
    CHAR_out(Hex(i % 16));
    CHAR_out(' ');
}

Init_Serial () {
    SCON = 0x52; // Mode 1 Ren
    TMOD = 0x20; // T0 Mode 2, T1 Mode 2
    TH1 = -3; // 9600 Baudrate
    TR1 = 1;
}

main () {
    char i;
DELTA ELECTRONIC
http://www.delta-electronic.com

```

```

char j;
Init_Serial ();

reg_A = 0x20; // Aktifkan osilator
reg_B = 0x12; // Update Interrupt aktif, Mode 24 Jam; Output BCD Data

while (1) {
    i = reg_C & 0x80;    // Periksa Flag IRQ

    if (i) {            // ada Update Interrupt
        reg_C = reg_C & 0x7F; // non-aktifkan ada Update Interrupt
        DisplayWaktu();    // Displaykan waktu
        for (j=0; j!=8; j++) {
            analog[j] = 0;
// while (INT0);

            CHAR_out(j+0x30);
            DisplayTegangan(analog[j]);    // Displaykan
        }
        CHAR_out(0x0D);
        CHAR_out(0x0A);
    }
}
}

```

Bahasa Assembly:

```

Alamat_RTC EQU 6000H
Alamat_ADC EQU 8000H
Detik EQU 00H
Menit EQU 02H
Jam EQU 04H
RegA EQU 0AH
RegB EQU 0BH
RegC EQU 0CH
RegD EQU 0DH

    Org 2000H ;Reset Vector
    Ajmp Start ;
    Org 2003H ;External Interrupt 0 Vector
    Reti ;
    Org 200BH ;Timer 0 Interrupt Vector
    Reti ;
    Org 2013H ;External Interrupt 1 Vector
    Reti ;
    Org 201BH ;Timer 1 Interrupt Vector
    Reti ;
    Org 2023H ;Serial Interrupt Vector
    Reti ;

Start:
    Mov DPTR, #Alamat_RTC+RegA ;Aktifkan osilator
    Mov A, #20H ;
    Movx @DPTR, A ;

```

```

    Inc    DPTR                ;Update Interrupt aktif, Mode 24 Jam
    Mov    A,#12H              ;Output BCD Data
    Movx   @DPTR,A            ;

Loop:
    Mov    DPTR,#Alamat_RTC+RegC ;Periksa Flag IRQ
    Movx   A,@DPTR            ;
    ANL    A,#080H            ;
    Jnb    A.7,Loop           ;Tidak ada flag, loop
    Mov    DPTR,#Alamat_RTC    ;Displaykan waktu
    Acall  Display_Waktu      ;
    Acall  Ambil_ADC          ;
    Ajmp   Loop

Ambil_ADC:
    Mov    DPTR,#Alamat_ADC

LoopAmbil_ADC:
    Movx   @DPTR,A
;    JB    INT0,*
    Mov    A,#' '
    Acall  Serial_Out
    Mov    A,DPL
    Acall  ASCII_Out
    Mov    A,#'='
    Acall  Serial_Out

    Movx   A,@DPTR
    Acall  ASCII_Out
    Inc    DPTR
    Mov    A,DPL
    Cjne   A,#08H,LoopAmbil_ADC

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

Display_Waktu:
    Mov    DPTR,#Alamat_RTC+Jam
    Movx   A,@DPTR
    Acall  ASCII_Out
    Mov    A,#':'
    Acall  Serial_Out
    Mov    DPTR,#Alamat_RTC+Menit
    Movx   A,@DPTR
    Acall  ASCII_Out
    Mov    A,#':'
    Acall  Serial_Out
    Mov    DPTR,#Alamat_RTC+Detik
    Movx   A,@DPTR
    Acall  ASCII_Out
    Ret

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

```

DELTA ELECTRONIC

<http://www.delta-electronic.com>

```

Init_Serial:
    MOV     SCON,#52H           ; Mode 1 Ren
    MOV     TMOD,#20H          ; T0 Mode 2, T1 Mode 2
    MOV     TH1,#0FDH          ; 9600 Baudrate
    MOV     TCON,#040H         ; T1 On, T0 Off
    MOV     PCON,#00H          ;
    RET

Serial_Out:
    CLR     TI
    MOV     SBUF,A
    JNB     TI,*
    RET

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
    Add    A,#30H              ;Tambah 30
    Ret

KirimPesan_Serial:
    Movx   A,@DPTR
    Acall  Serial_Out
    Jnz    KirimTerus_Serial
    Ret

KirimTerus_Serial:
    Inc    DPTR
    Ajmp   KirimPesan_Serial

Pesan_Channel:
    DB     20H, 'Channel', 00H

```