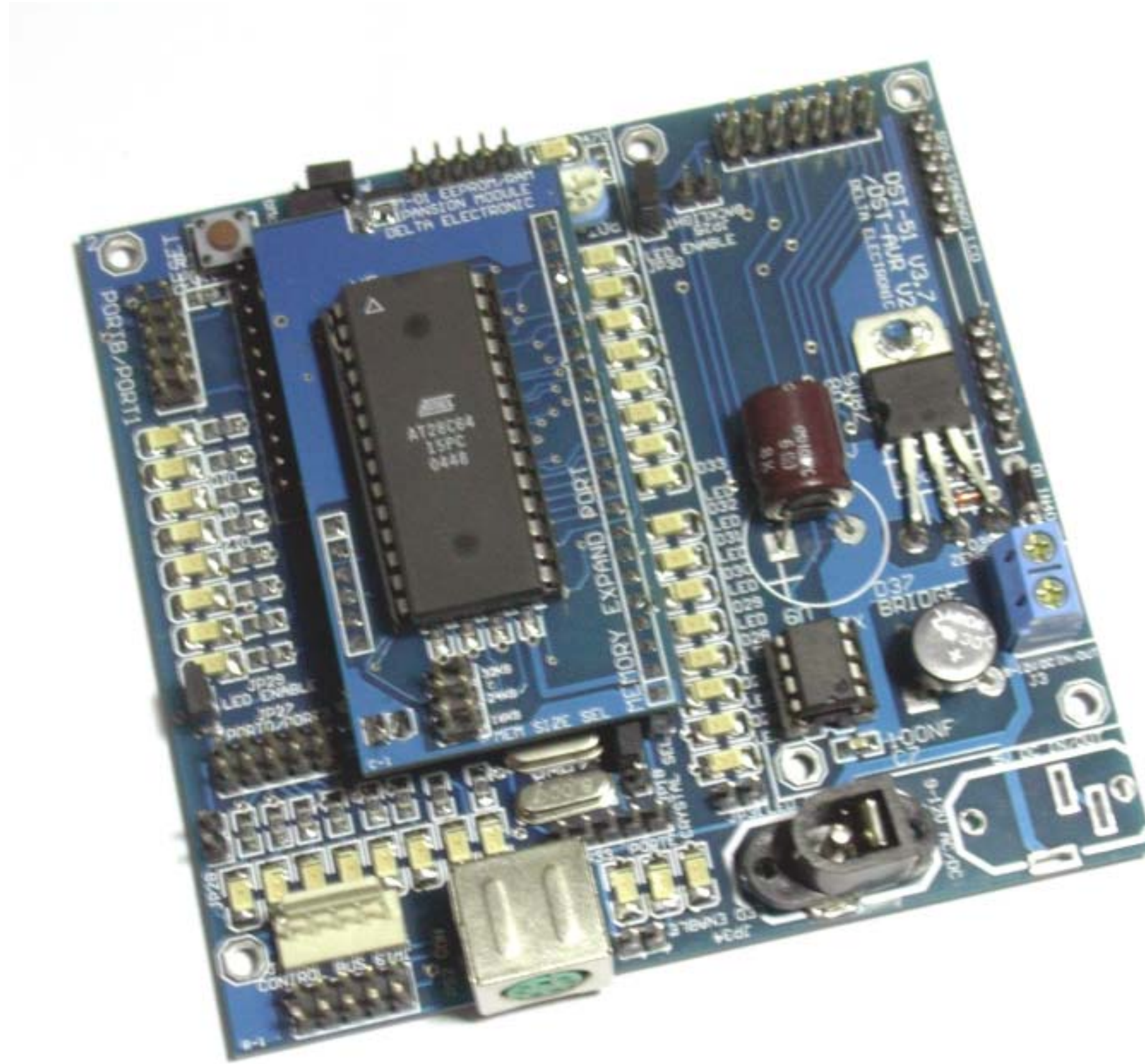


# Reference Manual

## DST-51/2 V3.7



- **AT89C51, AT89C52, AT89C55, AT89S51, AT89S52, AT89S53, AT89S8252 Support**
- **AVR ATMega8515 Support**
- **8 Kb Parallel EEPROM & 8Kb I2C Serial EEPROM**
- **2 LCD Port**
- **35 bit LED Logic Tester include**

## DESKRIPSI

DST-51/2 V3.7 adalah pengembangan lebih lanjut dari DST-51/2 V2.1 yang kini juga dilengkapi dengan berbagai fitur tambahan untuk mempermudah pengguna dalam mengembangkan suatu sistem aplikasi.

DST-51/2 V3.7 menggunakan dua mode yaitu Mode External Memory di mana program aplikasi yang didownload akan diletakkan pada memori eksternal dan Mode Flash PEROM di mana program aplikasi langsung didownload ke dalam Flash PEROM.

### 1. Mode Memori Eksternal

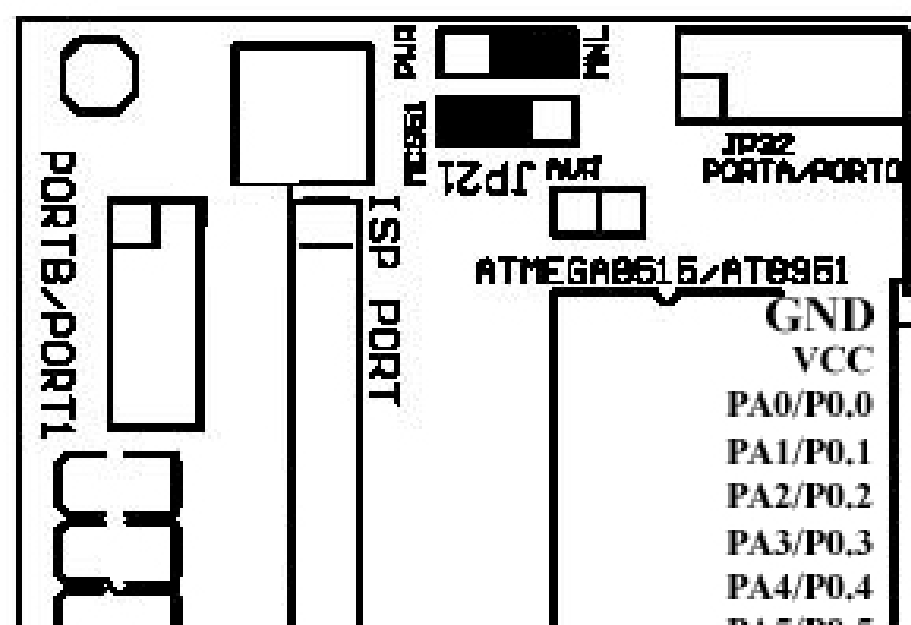
Pada mode ini, program yang didownload akan tersimpan di Memori Eksternal Parallel EEPROM sebesar 8Kb yang dapat diupgrade hingga 32Kb dengan mengganti AT28C64 dengan AT28C256.

Mode ini biasanya digunakan oleh programmer mikrokontroler yang telah terbiasa dengan Mode Memory Eksternal pada DST-51/2 V2.1 atau programmer pemula yang ingin menikmati kemudahan adanya rutin-rutin siap pakai DST-51/2

Mode ini terpisah menjadi dua yaitu Mode Monitor di mana kernel dari DST-51/DST-52 berfungsi sebagai Operating System dari modul dan Mode Stand Alone di mana kernel hanya memanggil dan menjalankan program yang ada pada memori EEPROM Eksternal saja.

#### 1.1. Mode Monitor

Pada mode ini kernel DST-51 atau DST-52 berfungsi sebagai operating system seperti DOS pada PC. Program aplikasi yang didownload pada mode ini akan dikirim dari PC melalui Port Serial/USB dan masuk ke Parallel EEPROM External.



#### Pengaturan Jumper

- Atur jumper MCU Selector pada posisi MCS-51
- Atur jumper Reset pada posisi Manual sehingga reset datang dari port serial PC

Gambar 1 Setting Jumper Monitor

Untuk DST-52 kernel yang digunakan adalah DDT-52 Delta Debugger Tools yang tidak hanya mendownload dan menjalankan program saja, melainkan dapat berfungsi melakukan debug/trace ataupun breakpoint untuk melacak kesalahan program.

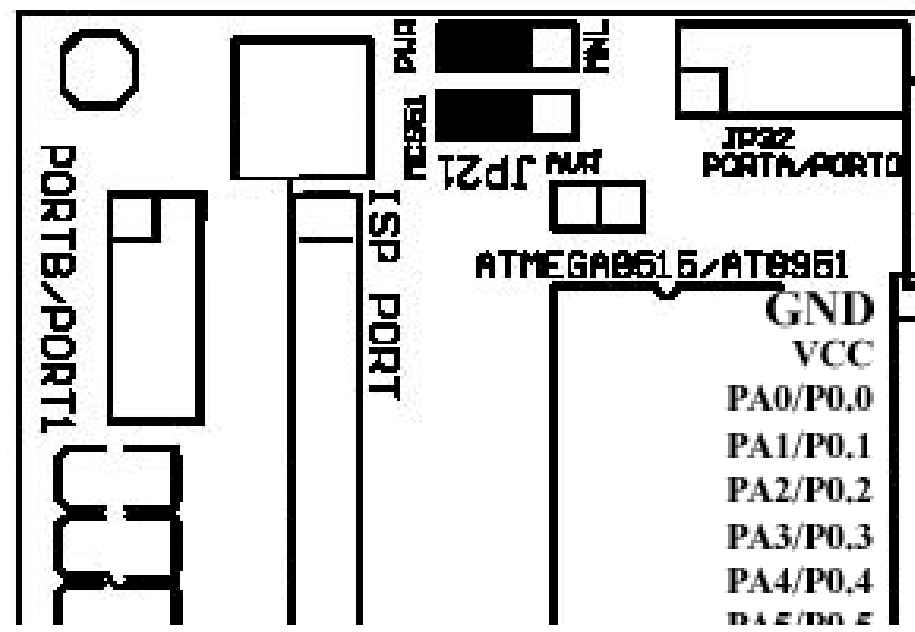
Trace atau debug adalah proses di mana jalannya setiap step program dapat dipantau dengan jelas termasuk perubahan-perubahan yang terjadi pada register-register.

Alamat	Instruksi	
DDT-52! Delta Debugger Tools 52 Ver 2.5		
>T		
2000	CLR P1.0	C2 90 C2 ← Kode Operasi
DPTR=0000 A=00 B=00 SP=08 PC=2002 PSW=00		
R0=00 R1=00 R2=00 R3=00 R4=00 R5=00 R6=00 R7=00 ← Register		
>T		
2002	CLR P1.1	C2 91 54
DPTR=0000 A=00 B=00 SP=08 PC=2004 PSW=00		
R0=00 R1=00 R2=00 R3=00 R4=00 R5=00 R6=00 R7=00		
>		
-		

Proses debug seringkali terhalang oleh rutin-rutin program yang mempunyai nilai iterasi sangat besar sehingga proses harus berputar-putar cukup banyak sebelum keluar dari iterasi tersebut, contohnya pada rutin delay. Untuk melewati bagian tersebut maka pengguna dapat menempatkan breakpoint di lokasi alamat sesudah bagian yang mempunyai nilai iterasi besar tersebut sehingga program dapat berhenti tepat di lokasi tersebut saat program dijalankan dan proses debug dapat dilakukan kembali

## 1.2. Mode Stand Alone

Pada mode ini DST-51/2 tidak lagi terhubung dengan PC karena program yang telah dirancang pada mode monitor telah tersimpan dalam memori EEPROM Eksternal. Pada saat ini, DST-51/2 sudah dapat menjadi aplikasi yang siap pakai karena program yang ada pada memori EEPROM Eksternal akan selalu dieksekusi/dijalankan setiap kali power supply diaktifkan



Gambar 3 Setting Jumper Stand Alone Mode

### Pengaturan Jumper

- Atur jumper MCU Selector pada posisi MCS-51
- Atur jumper Reset pada posisi Power sehingga reset berasal dari kondisi Power Up

PERLU DIPERHATIKAN: Pada Mode Memori Eksternal yang menggunakan Kernel DST-51/2, pengguna dilarang menggunakan Kabel ISP untuk mengisi program karena dapat menghapus isi Kernel. Gantikan Kernel DST-51/2 yang ada dengan IC AT89S51/2 kosong sebelum anda menggunakan Kabel ISP yang seharusnya digunakan pada Mode Flash PEROM.

### Keunggulan Mode Memori Eksternal dibanding Flash PEROM

Mode Memori Eksternal adalah Mode DST-51/2 yang hanya dapat dilakukan pada Professional Version di mana pada bagian ini mempunyai fitur-fitur sebagai berikut:

- Kernel DST-51 Delta Monitor Tools untuk DST-51 atau DDT-52 Delta Debugger Tools untuk DST-52 di mana pada Delta Monitor Tools pengguna dapat mendownload, edit dan melihat isi memori RAM maupun EEPROM tempat program aplikasi didownload sedangkan pada DDT-52 Delta Debugger Tools selain fitur yang ada pada DST-51, pengguna juga dapat melacak kesalahan program dengan trace dan breakpoint seperti dijelaskan di atas.
- Pengguna dapat menggunakan rutin-rutin siap pakai yang ada pada kernel dalam mengembangkan sistem. Contohnya untuk inialisasi LCD, pengguna cukup memberikan perintah **Lcall Init\_LCD** saja
- 8 bit Port 0 dapat digunakan untuk Teknik Interfacing Parallel Bus di mana port ini dapat dihubungkan ke beberapa komponen sekaligus dengan pengalamatan yang berbeda. Mode ini sesuai untuk komponen-komponen seperti ADC0809, MT8888, DS12887 di mana pengguna tidak perlu mengatur sinyal-sinyal di kaki WR, RD dll yang harus dikirimkan ke komponen tersebut.
- Memori EEPROM Eksternal dapat difungsikan juga sebagai media penyimpanan data maupun variabel dan dapat diupgrade hingga 32Kb.

Hal ini biasanya digunakan untuk program-program yang membutuhkan variabel yang cukup banyak di mana 128 byte RAM Internal AT8951 atau 256 byte RAM Internal AT8952 sudah tidak mencukupi.

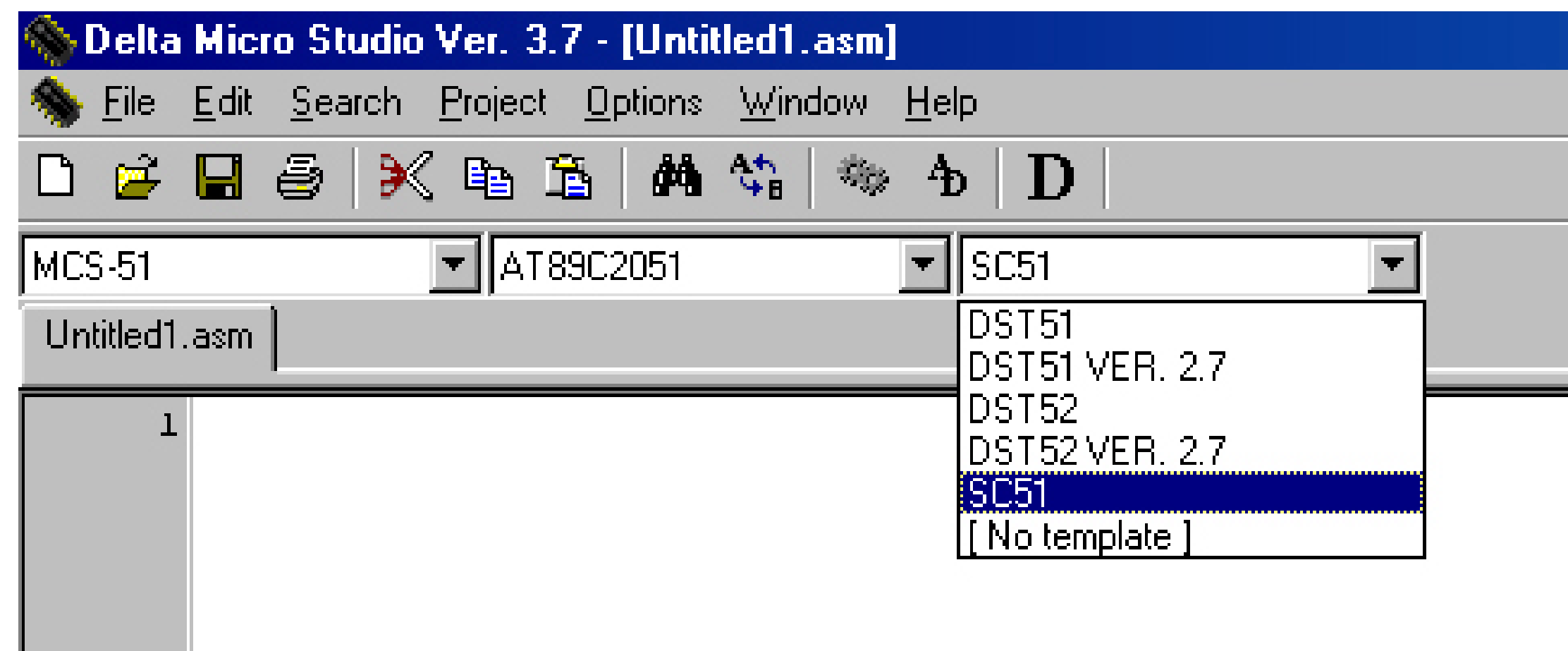
## 2. Mode Flash PEROM

Pada mode ini program aplikasi akan didownload ke dalam Flash PEROM dari AT89S51 atau AT89S52 dan DST-51/2 akan berjalan tanpa bantuan kernel melainkan murni dari program aplikasi yang anda download/programkan.

Rutin-rutin siap pakai yang ada di dalam kernel tidak dapat berfungsi lagi pada mode ini. (Lihat daftar rutin siap pakai)

Pengguna dapat tetap menggunakan rutin-rutin siap pakai tersebut dengan mengambil dan meng-copy rutin-rutin tersebut dari CD dan menggabungkan pada program aplikasi.

Apabila anda menggunakan Delta Studio, maka gunakan Template SC51 sebagai pilihan.



**Gambar 4 Setting Delta Studio pada Mode Flash PEROM**

Pengisian program pada mode ini adalah menggunakan teknik ISP (In System Programming) sehingga dibutuhkan Kabel ISP sebagai perangkat programmernya.

### **Keuntungan Mode Flash PEROM dari Mode Memori Eksternal**

Pada mode ini pengguna cukup menggunakan Paket DST-51/2 Economic Version di mana Modul Memori Expanded tidak diperlukan.

- Lebih ekonomis karena tidak menggunakan memori EEPROM eksternal
- RAM Internal AT8951/AT8952 masih bebas karena tidak digunakan oleh Kernel DST-51/2
- Port 0, Port 2, P3.6 dan P3.7 masih bebas karena tidak digunakan sebagai pengalamatan memori EEPROM Eksternal
- Program aplikasi ada pada Flash PEROM sehingga setelah selesai pengembangan/perancangan program, IC yang telah terisi dapat langsung digunakan pada sistem target/aplikasi secara single chip tanpa mengubah alamat program.

**PERLU DIPERHATIKAN:** Bila anda menggunakan Paket DST-51/2 Pro Version atau Multi Micro Version, lepaskan dahulu IC AT8951/2 bersegel Delta yang berfungsi sebagai media penyimpanan program Kernel DST-51/2 dan gantikan dengan IC AT89S51/2 yang masih kosong agar program Kernel bawaan dari Delta tidak hilang.

#### **PETA MEMORI**

Memori pada DST-51/2 terdiri dari 2 bagian besar yaitu:

- Memori Internal, berupa media penyimpanan data maupun program dan terdapat di dalam IC AT8951/2
- Memori Eksternal, berupa media penyimpanan data maupun program dan terdapat di luar IC AT8951/2

Tabel 1 PETA MEMORI PADA DST-51/2

	Memori Internal		Memori Eksternal	
	RAM Internal	Flash PEROM	Eksternal EEPROM	I2C Serial EEPROM
<b>DST-51 Economic Version</b>	128 byte, 00 - 7Fh Untuk menyimpan variabel, stack, data sementara	4Kb, 0000 - 0FFFh Untuk menyimpan Program Aplikasi User	Tidak ada dalam versi ini	1Kb, 000 - 3FFh Untuk menyimpan data permanen
<b>DST-52 Economic Version</b>	256 byte, 00 - FFh Untuk menyimpan variabel, stack, data sementara	8Kb, 0000 - 1FFFh Untuk menyimpan Program Aplikasi User	Tidak ada dalam versi ini	1Kb, 000 - 3FFh Untuk menyimpan data permanen
<b>DST-51 Professional Version</b>	128 byte, 00 - 7Fh Untuk menyimpan variabel, stack, data sementara	4Kb, 0000 - 0FFFh Untuk menyimpan Program Kernel DST-51	8Kb, 0000 - 1FFFh Untuk menyimpan Program Aplikasi User, Variabel, data permanen	8Kb, 000 - 1FFFh Untuk menyimpan data permanen
<b>DST-52 Professional Version</b>	256 byte, 00 - FFh Untuk menyimpan variabel, stack, data sementara	8Kb, 0000 - 1FFFh Untuk menyimpan Program Kernel DST-52	8Kb, 0000 - 1FFFh Untuk menyimpan Program Aplikasi User, Variabel, data permanen	8Kb, 000 - 1FFFh Untuk menyimpan data permanen

Selain memori ada beberapa pengalamatan perlu diperhatikan terutama apabila anda menggunakan Teknik Interfacing Parallel Bus yaitu:

CS2: 4000h – 5FFFh

CS3: 6000h – 7FFFh

CS4: 8000h – 9FFFh

CS5: A000h – BFFFh (telah digunakan untuk LCD M1632, dapat digunakan apabila LCD tidak dipasang)

CS6: C000h – DFFFh

CS7: E000h – FFFFh

### **RAM Internal DST-51/2 pada Mode Memori Eksternal**

Perlu diperhatikan bahwa pada saat DST-51/2 menggunakan kernelnya maka ada sebagian lokasi dari RAM Internal yang digunakan oleh program tersebut dan sebaiknya dihindari pada program aplikasi yang dirancang pada Mode Memori Eksternal.

### **Alamat yang digunakan oleh Program Monitor**

- 20H hingga 27H untuk Variabel Program Monitor
- 30H hingga 4FH untuk Buffer Perintah bagi Program Monitor

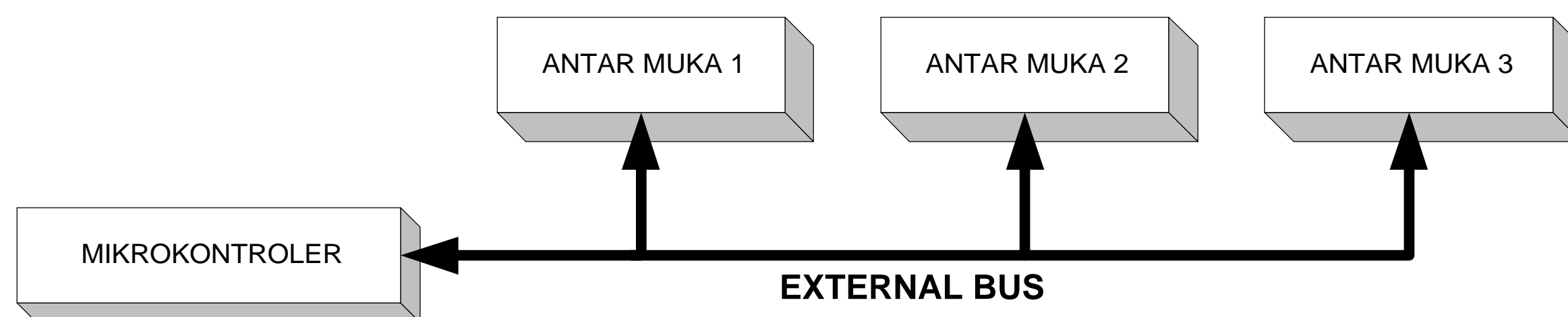
Alamat ini sebetulnya tetap dapat digunakan oleh program aplikasi, namun tidak dapat dilihat dengan perintah Dump Internal.

### **Alamat yang digunakan oleh Rutin Siap Pakai**

- 28H untuk rutin delay
- 29H untuk variabel rutin LCD
- 2AH untuk variabel rutin Phone Interface

Alamat ini tidak dapat digunakan selama program aplikasi anda menggunakan rutin siap pakai ini yang terletak di dalam Kernel. Alamat ini dapat digunakan bila rutin siap pakai sudah digabung bersama program aplikasi atau menggunakan Template SC-51 pada Delta Studio

### **Teknik Interfacing Sistem Bus**



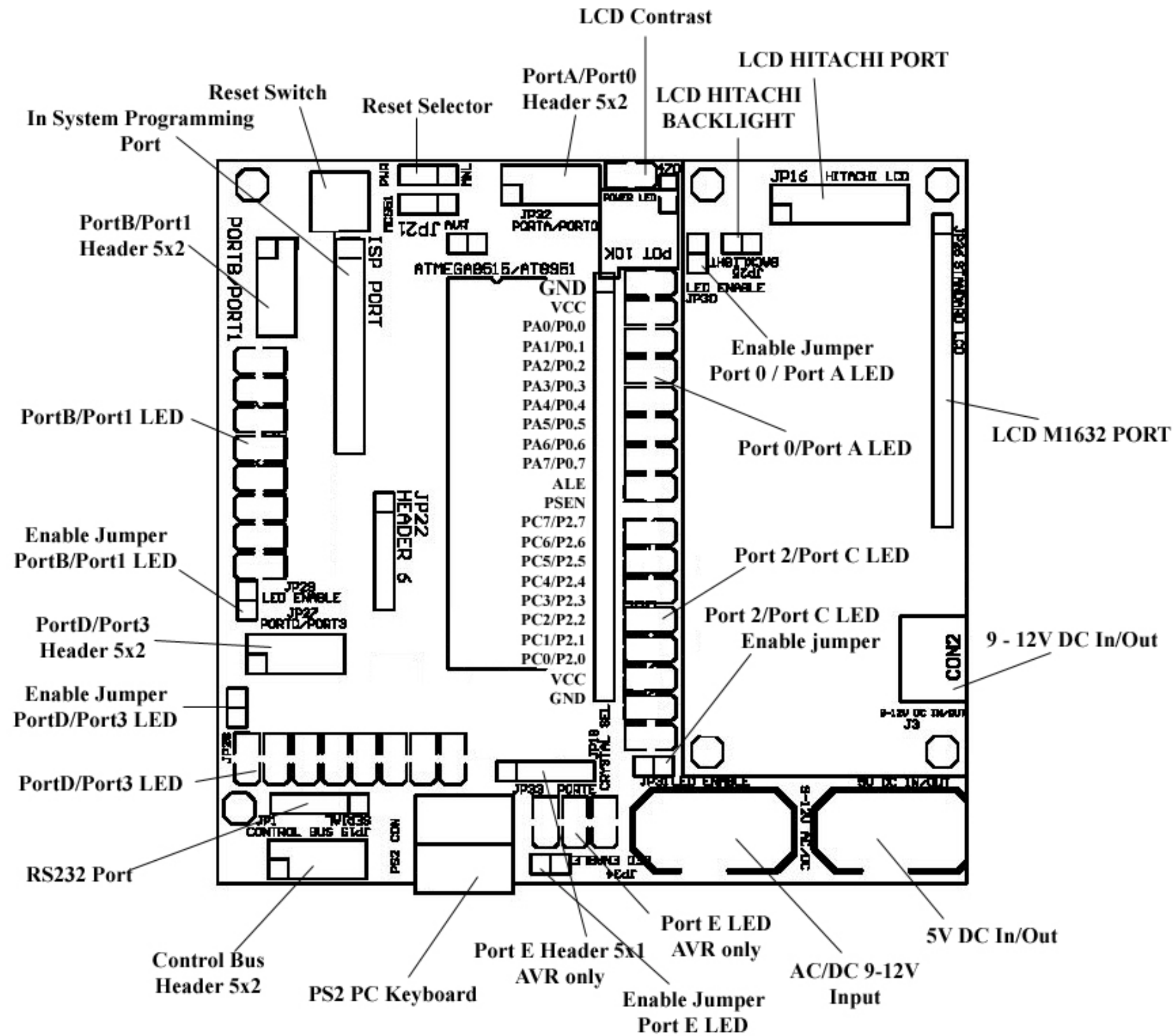
Teknik *Interfacing* (Antar Muka) Sistem Bus adalah teknik penggunaan jalur bus sebagai I/O antara DST-51/2 dan modul-modul ataupun komponen-komponen pendukungnya. Dengan menggunakan teknik ini semua modul-modul ataupun komponen-komponen yang mempunyai kemampuan akses melalui jalur bus terhubung menjadi satu dalam jalur tersebut sehingga jumlah I/O yang digunakan dapat diringkas dan Port 0 yang digunakan sebagai Data Bus untuk mengakses memori eksternal pada Mode Monitor dan Mode Memori Eksternal dapat digunakan sebagai I/O untuk komponen-komponen lain.

Untuk membedakan komponen-komponen yang diakses, maka setiap



komponen ataupun modul mempunyai alamat-alamat tersendiri yang diatur oleh decoder. Pada Modul DST-51/2 disediakan alamat CS3, CS4, CS5, CS6 dan CS7 untuk keperluan tersebut.

## DESKRIPSI HARDWARE DST-51/2



Gambar 6 Tata Letak Komponen Evaluation Board DST-V3.7

**In System Programming Port**

Port yang dihubungkan dengan Kabel ISP untuk pemrograman pada Mode Flash PEROM.

**Reset Switch**

Tombol ini berfungsi untuk me-reset system DST-51/2 secara manual

**Reset Selector**

Jumper pemilih di mana sistem reset dapat diatur aktif secara otomatis atau manual melalui port serial

**Port0 Header 5x2**

Port 0 dengan format header 5 x 2 merupakan Port Data Bus pada Teknik Interfacing Parallel Bus.

**Port1 Header 5x2**

Port 1 dengan format header 5 x 2

**Port2 Header 22**

Port 2 dengan format header 10 x 1 dan digunakan untuk slot EEPROM/RAM Expanded Module pada Mode Memori Eksternal dan sebagai 8 bit General I/O port pada Mode Flash PEROM

**Port 0 Header 22**

Port 0 dengan format header 10 x 1 dan digunakan untuk slot EEPROM/RAM Expanded Module pada Mode Memori Eksternal dan sebagai 8 bit General I/O port pada Mode Flash PEROM

**Port3 Header 5x2**

Port 3 dengan format header 5 x 2

**Control Bus Header 5x2**

Port Control Bus dengan format header 5x2 sesuai standard DST-51/2 V2.1 dan digunakan untuk Teknik Interfacing Parallel Bus

**RS232 Port**

Port RS232 yang dihubungkan pada kabel RS232 untuk berkomunikasi dengan PC

### **PS2 PC Keyboard Connector**

Keyboard Connector yang digunakan untuk aplikasi input dari DST-51

### **Enable Jumper**

Jumper yang mengatur aktif tidaknya LED pada port-port tersebut. Untuk aplikasi yang membutuhkan konsumsi arus yang besar LED logic dapat dinonaktifkan dengan melepas enable jumpernya

### **AC/DC 9-12V Input**

Input sumber daya dari DST-51 di mana input dapat diambil dari tegangan AC maupun DC pada rentang 9 – 12 Volt

### **5V DC In/Out**

Input/output sumber daya 5V DC, pada kondisi standard, connector ini tidak dipasang untuk menghindari pengguna salah memasang dengan connector AC/DC 9-12V input. Pada saat AC/DC 9-12V diberi input, maka bagian ini berfungsi sebagai output. Pada sumber daya 5 Volt, port ini dapat digunakan sebagai input.

### **9-12V DC In/Out**

Input/output sumber daya 9-12V DC, pada saat connector AC/DC 9-12V input diberi sumber daya, maka bagian ini akan berfungsi sebagai output.

Untuk sumber daya 9-12V DC, connector ini dapat juga berfungsi sebagai input.

### **LCD M1632 Port**

Port LCD untuk jenis M1632 standard

### **LCD HITACHI Port**

Port LCD untuk jenis Hitachi

### **LCD HITACHI Backlight**

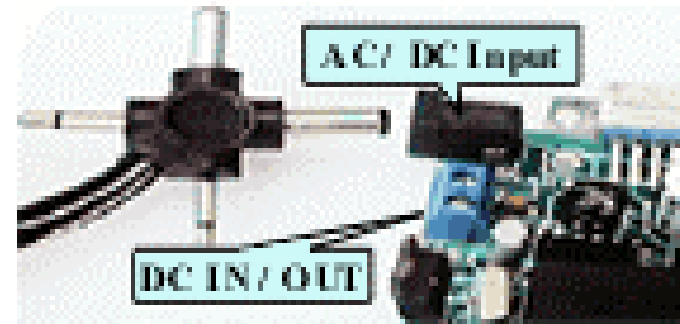
Sumber backlight untuk LCD Hitachi

### **LCD Contrast**

Potensio pengatur kontras dari karakter-karakter pada layar LCD

## POWER SUPPLY

DST-51/2 dilengkapi dengan rangkaian penyearah dan regulator sehingga untuk keperluan power supply hanya diperlukan sebuah trafo penurun tegangan dari 220V ke 9 atau 12 Volt saja. Walau demikian, input power supply 9-12 Volt



AC tersebut juga dapat digunakan sebagai input power supply DC sebesar 9-12 Volt pula di mana power supply (adaptor) tersebut juga banyak tersedia di pasaran dengan harga yang relatif murah.

Bahkan dengan adanya rangkaian penyearah, anda tidak perlu memikirkan polaritas tegangan dari adaptor ataupun power supply tersebut. Input ini dihubungkan ke 9 – 12V AC/DC di mana connector tersebut memang disediakan sebagai connector yang sesuai dengan power supply siap pakai yang ada di pasaran.

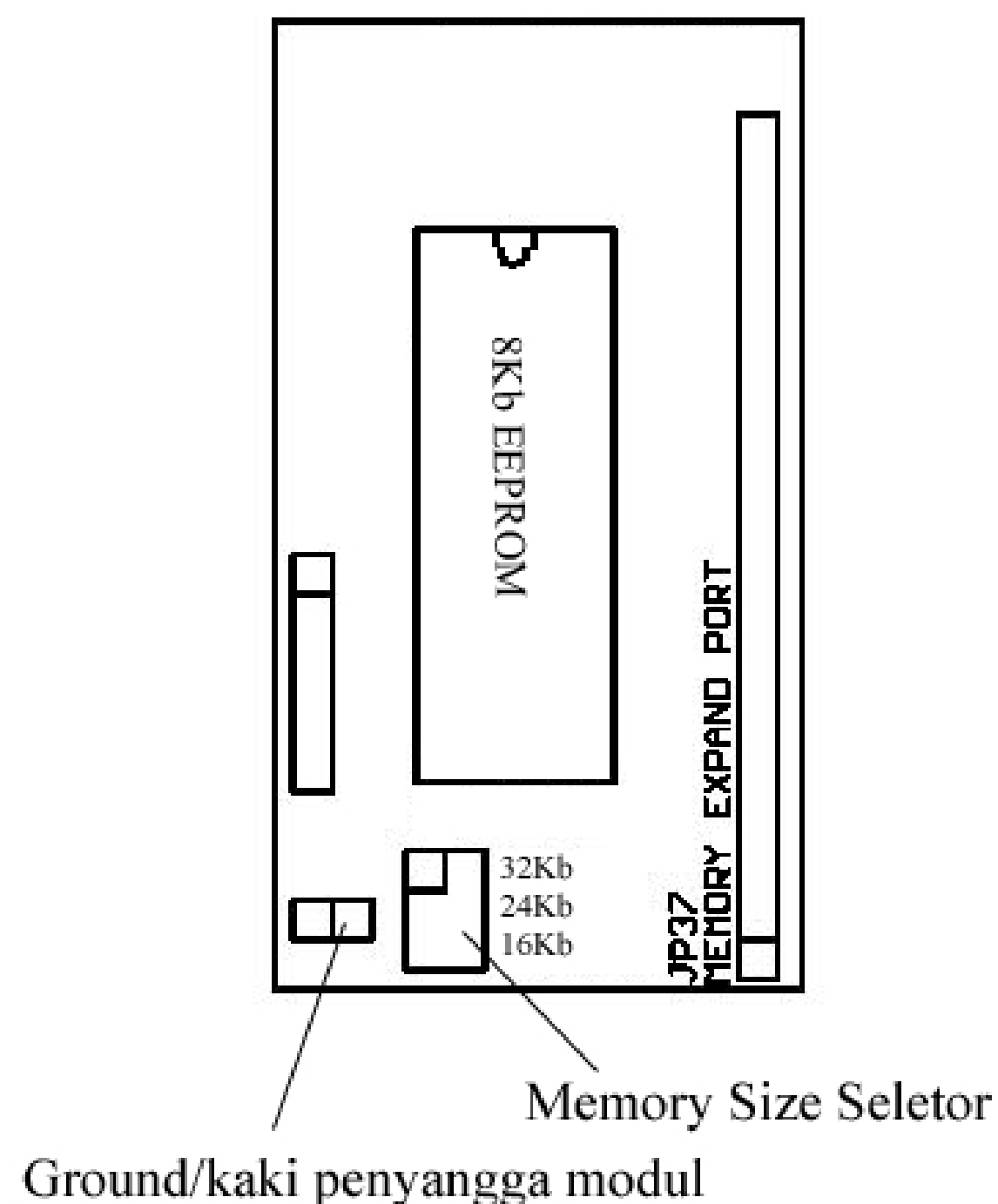
Apabila connector tersebut tidak sesuai dengan power supply anda, maka connector alternatif J3 yang bersifat lebih universal di mana dua buah kabel power supply tanpa connector saja dapat dihubungkan ke dalamnya dapat anda gunakan sebagai input power supply. Namun untuk connector ini perlu diperhatikan polaritas tegangan dari power supply anda. Sisi positif power supply harus dipastikan terhubung dengan sisi positif pada connector ini di mana sisi tersebut ditandai dengan tanda “+”. Hal ini disebabkan karena input melalui connector ini memang didisain tidak mengalirkan tegangan input ke rangkaian regulator melalui penyearah melainkan justru terhubung dengan keluaran dari penyearah dan langsung masuk ke rangkaian regulator. Lebih detailnya dapat dilihat pada skema yang terlampir.

Fungsi J3 tersebut sebetulnya bukan hanya sebagai connector alternatif saja melainkan juga sebagai output **DC sebesar 9-12 Volt** (tergantung input ACnya) apabila connector 9-12V AC/DC digunakan sebagai input power supply. Hal ini disebabkan karena tegangan yang AC yang masuk melalui connector 9-12V AC/DC telah disearahkan oleh rangkaian penyearah dan keluar di connector JP3.

Sedangkan 5 Volt DC In/Out berfungsi sebagai input bila sumber daya menggunakan power supply switching 5 Volt. Pada kondisi standard, connector

ini tidak diberikan agar menghindari kesalahan sumber pemasangan sumber tegangan yang dapat merusak perangkat keras Evaluation Board DST V3.7

### DESKRIPSI HARDWARE EEPROM/RAM EXPANSION MODULE



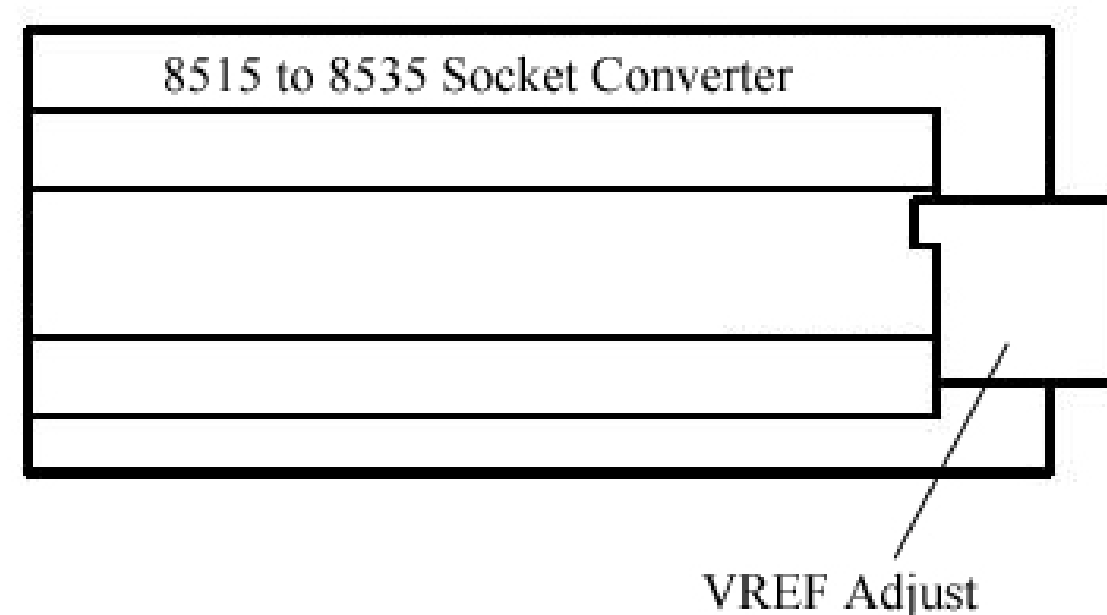
Modul ini merupakan Memori Expansion Module yang berfungsi sebagai media penyimpanan data/program aplikasi pada Mode Eksternal Memori.

Khusus untuk DST Multi Microcontroller kapasitas modul ini dapat mencapai 32Kb dengan mengubah posisi jumper pada memori size selector.

Pada saat modul ini digunakan, LED Enable Jumper pada Port 0 dan Port 2 sebaiknya dilepaskan.

Gambar 8 Tata letak EEPROM/RAM Expanded Module

### DESKRIPSI HARDWARE 8515 TO 8535 SOCKET CONVERTER

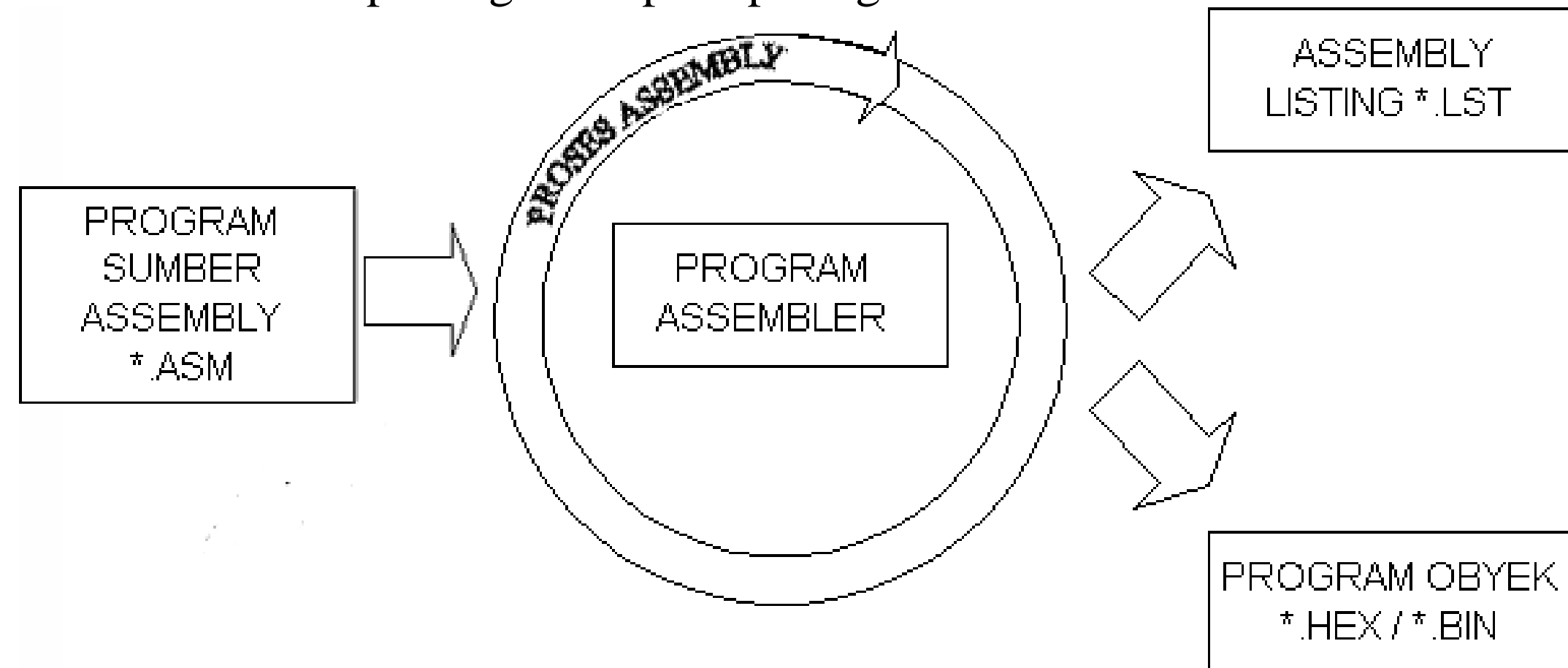


Modul ini digunakan pada paket DST-AVR atau DST-Multi Microcontroller dan berfungsi untuk keluarga AVR 40 pin yang memiliki ADC seperti ATmega8535

Terdapat VREF adjust untuk mengatur tegangan referensi ADC dari ATmega8535

## PENGISIAN PROGRAM

Untuk mengisi program ke dalam mikrokontroler terlebih dahulu harus dilakukan beberapa langkah seperti pada gambar berikut



### Proses Assembly

Program sumber assembly yang telah diketik dengan menggunakan teks editor terlebih dahulu diubah menjadi program obyek melalui proses assembly. Untuk melakukan proses assembly dibutuhkan sebuah program yang disebut program assembler yang berfungsi melakukan konversi ke dalam program obyek di mana program obyek tersebut dapat berbentuk HEX atau BIN.

Selain program obyek, program assembler juga dapat menghasilkan file listing assembly. Untuk program assembler, pengguna dapat menggunakan ASM51 yang sudah termasuk dalam CD DST-5x.

### Proses Download

Program Obyek yang merupakan hasil dari proses assembly dapat didownload ke dalam sistem mikrokontroler dengan menggunakan dua macam cara. Pada Mode Monitor dapat dilakukan dengan menggunakan DLD51.exe sedangkan pada versi ISP dilakukan dengan menggunakan software ISP (lebih detail baca manual Kabel ISP yang digunakan).

## IDE Software

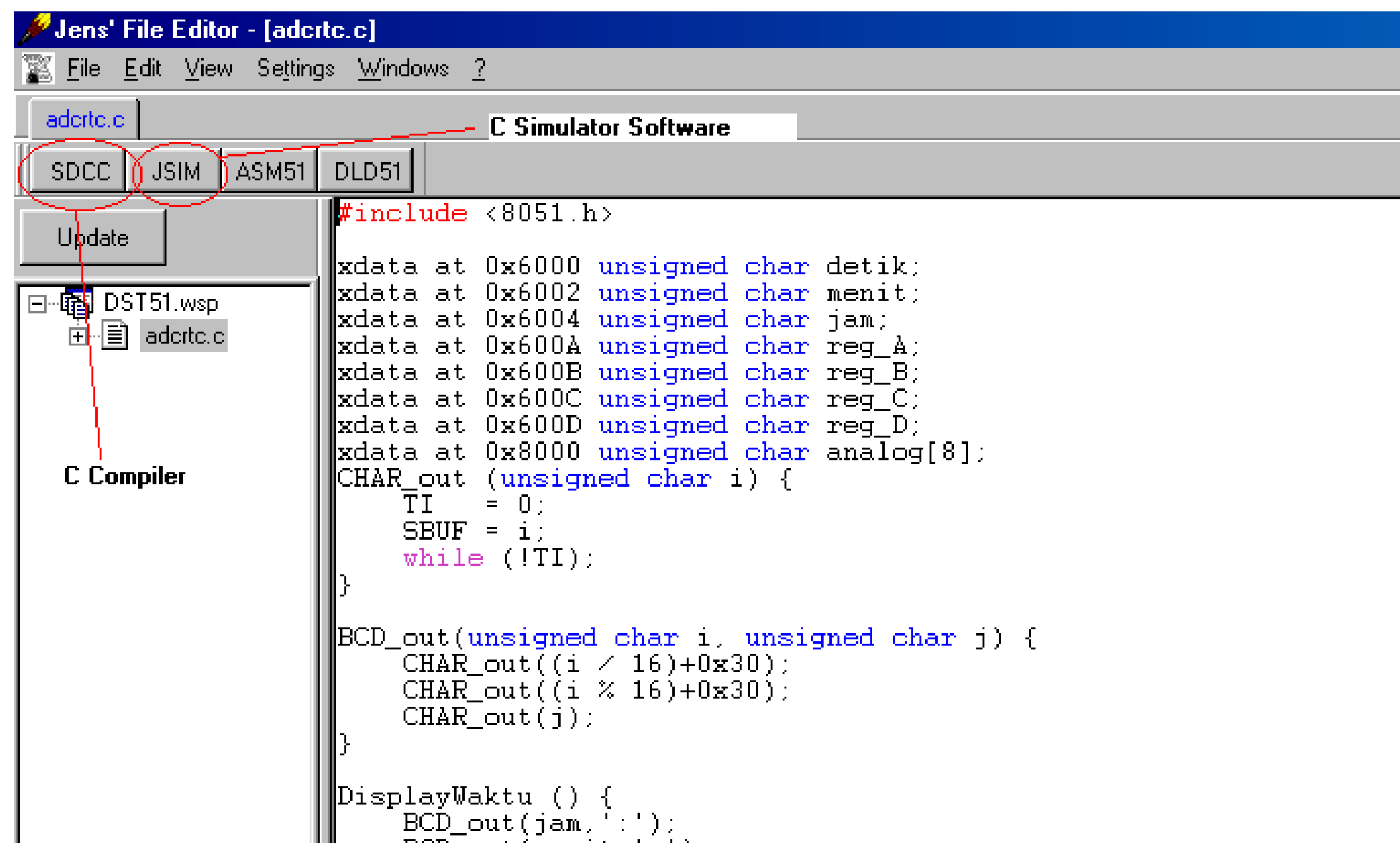
IDE (Integrated Development Environment) adalah software yang berfungsi menggabungkan Editor, Assembly dan Downloader di dalam satu software yang terintegrasi sehingga pengguna dalam merancang program hanya cukup mengendalikan satu software saja.

Pada paket ini disediakan dua buah software IDE yaitu Jeans File Editor yang merupakan software freeware dan Delta Micro Studio yang merupakan Software IDE yang dikembangkan oleh Delta Electronic.

### 1. Jeans File Editor

Software ini adalah merupakan software teks editor freeware di mana pengguna dapat menambahkan tombol-tombol tertentu yang dapat memanggil dan mengirimkan parameter ke program yang dipanggil sehingga membentuk suatu IDE.

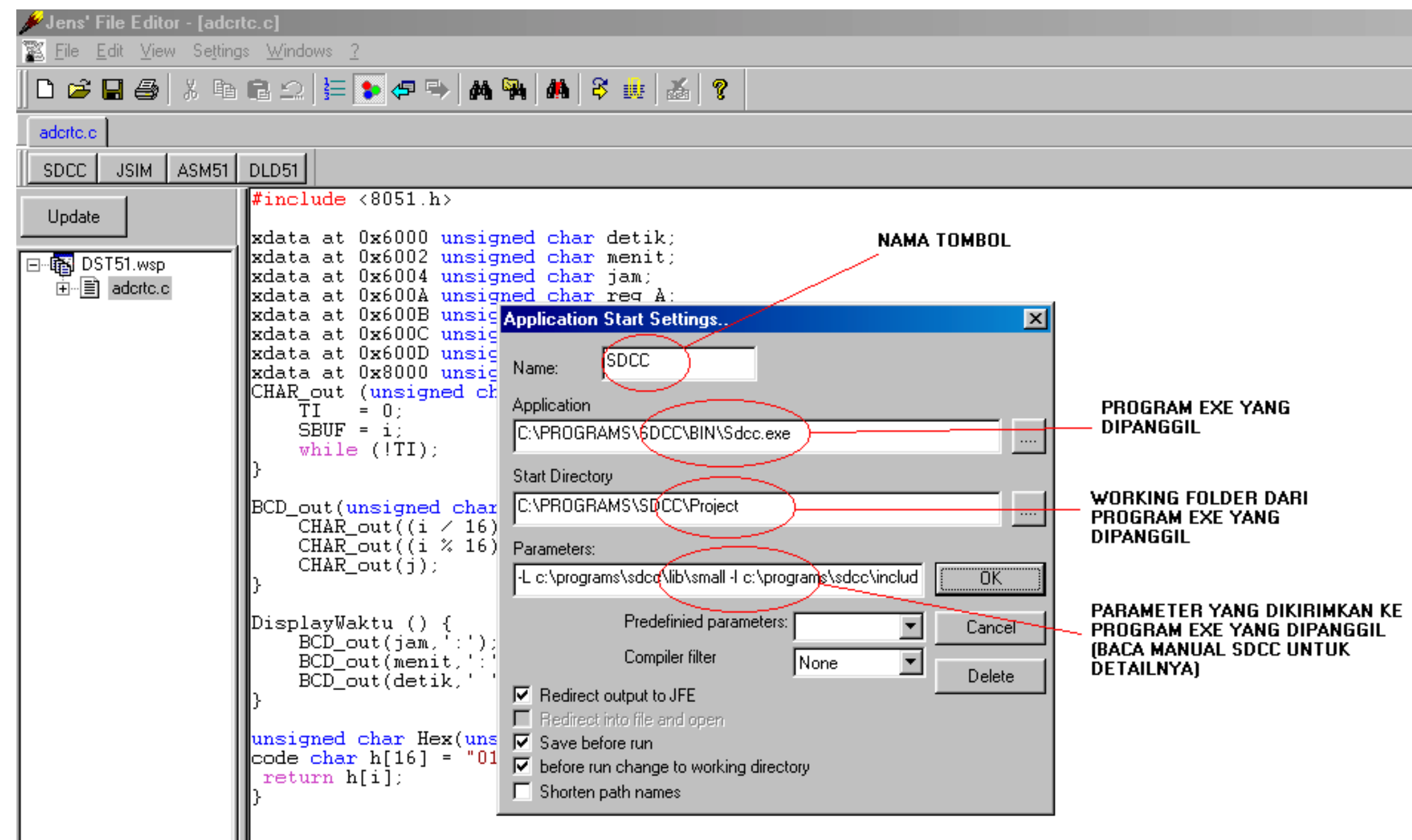
Software ini sebetulnya tidak berjalan sendiri, namun memanggil software assembler dan downloader untuk proses assembly ataupun download sehingga kinerjanya masih tergantung pada software-software tersebut. Contohnya untuk assembler, memanggil ASM51 dan downloadernya memanggil DLD51.



Gambar 11 JFE yang sudah disetup

JFE juga dapat digabungkan dengan C Compiler SDCC maupun C Simulator JSIM seperti pada gambar 11 di atas.

Agar JFE dapat disetup dengan tombol-tombol seperti pada gambar 11 maka pengguna harus menambahkan tombol-tombol tersebut secara manual dengan fasilitas Add Tool di Menu Setting



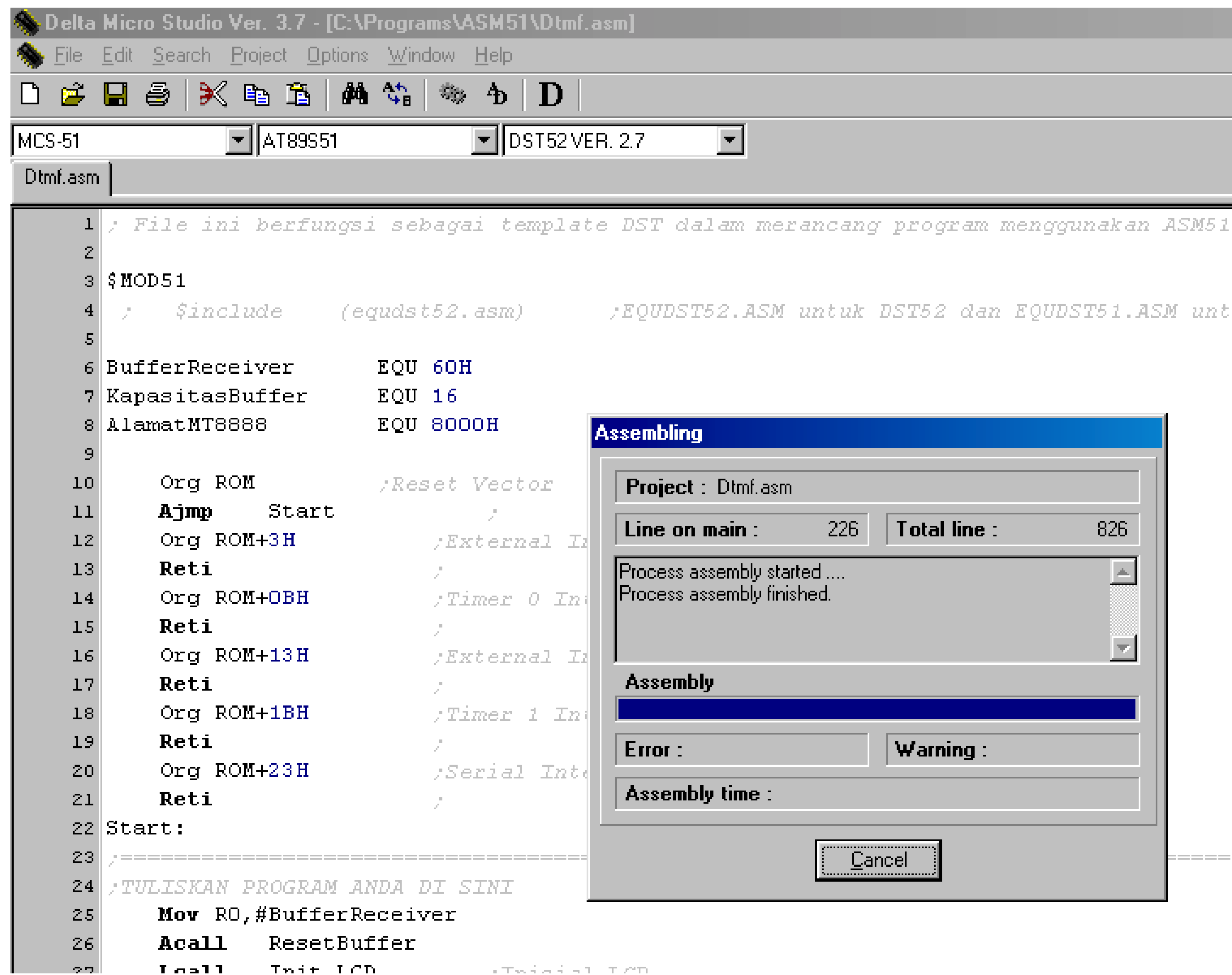
**Gambar 12 Add Tool pada JFE**

Namun untuk pengguna yang tidak ingin direpotkan dengan pengaturan setting pada JFE dapat langsung menggunakan JFE yang telah disetting secara standard oleh Delta Electronic untuk kebutuhan DST-51 dengan cara meng-ekstrak file DSTTools.zip yang ada pada CD DST-51 ke folder C:\Programs. Pengguna dapat memindah ke folder lain, namun setting pada setiap tombol-tombol pada JFE harus disesuaikan dengan folder tersebut. Lebih detail mengenai hal ini dapat anda lihat di file QuickStart yang ada pada CD.

## 2. Delta Micro Studio

Apabila pada JFE kinerjanya masih tergantung dengan software-software lain, maka Delta Micro Studio berjalan secara independent karena Software Assembler, Editor dan Downloaernya sudah terintegrasi di dalam software ini. Pada saat proses download maupun assembly Delta Micro Studio tidak perlu membuka window baru untuk software lain seperti pada JFE.





**Gambar 13 Proses Assembly & Download pada Delta Studio**

Tidak hanya sekedar menggabungkan software assembly dan download saja, Delta Studio juga memberikan kemudahan bagi pengguna dengan adanya proses assembly dan download yang hanya dilakukan dengan satu proses melalui short Key Ctrl F9 saja.

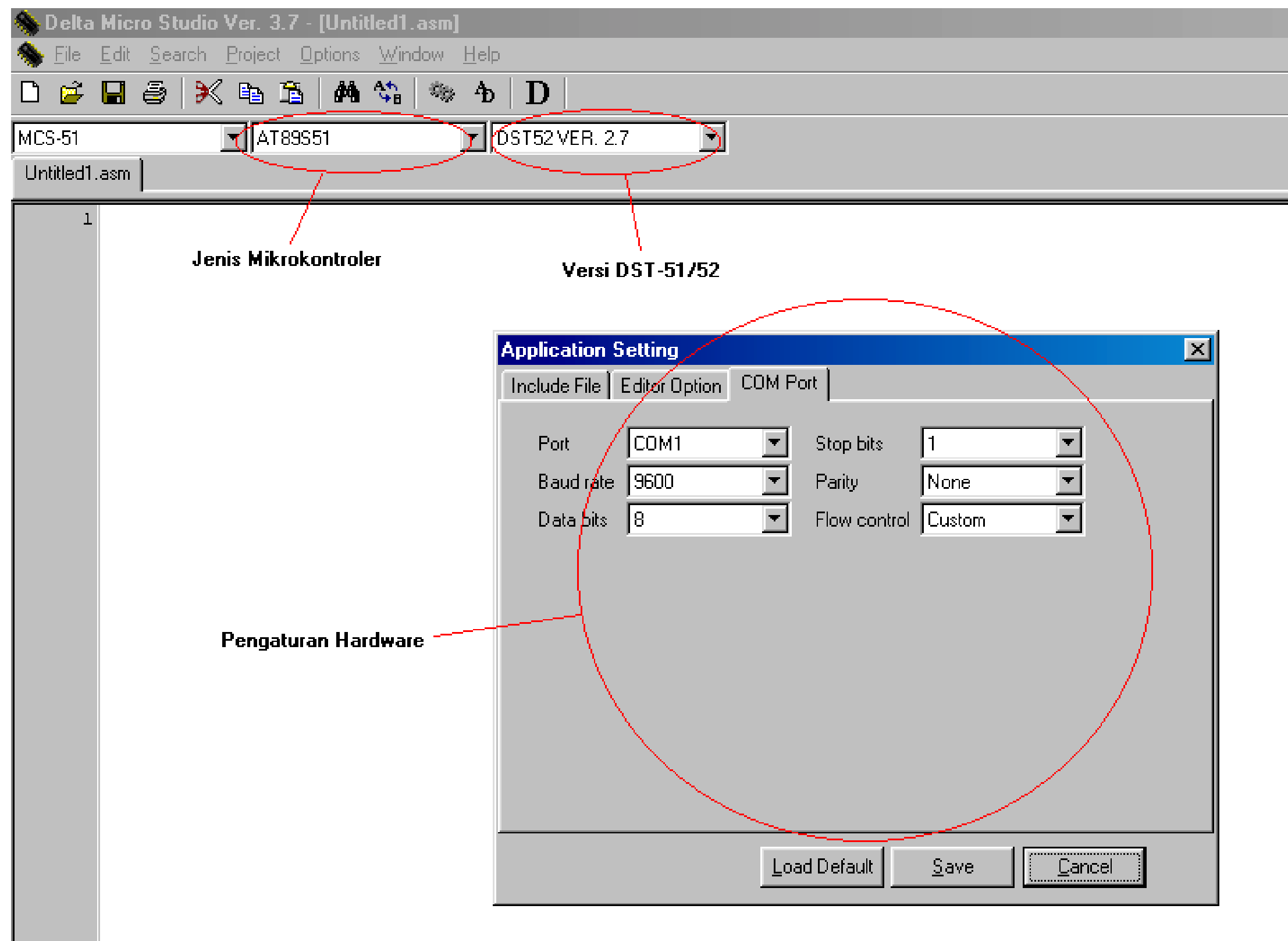
Untuk menjaga kompatibilitas, Delta Studio dapat digunakan untuk melakukan assembly pada source code yang dirancang oleh ASM51 maupun ALDS.

Proses download pada Delta Studio untuk versi V3.7 ini hanya dapat dilakukan pada Mode Monitor saja, **untuk mode Flash PEROM Delta Studio hanya dapat berlaku sebagai Software Editor dan Assembly saja.**

### Setting Delta Studio

Sebelum digunakan ada beberapa hal yang perlu diatur terlebih dahulu pada Delta Studio seperti pada gambar 14. Jenis mikrokontroler menentukan mikrokontroler apa yang sedang kita gunakan dan berikutnya adalah versi dari Modul DST-51/52 yang digunakan.

Setting hardware adalah hal yang harus dilakukan pertama kali agar DST-51/52 dapat terhubung dengan Delta Studio. Setting ini meliputi port serial yang digunakan dan bentuk data komunikasi. Port serial yang digunakan adalah port yang terhubung pada DST-51/52 saat itu, dapat juga berupa port USB to serial yang terdeteksi sebagai COM juga sedangkan bentuk data adalah baudrate 9600 dan 8 bit data, 1 stop bit dan Parity None.

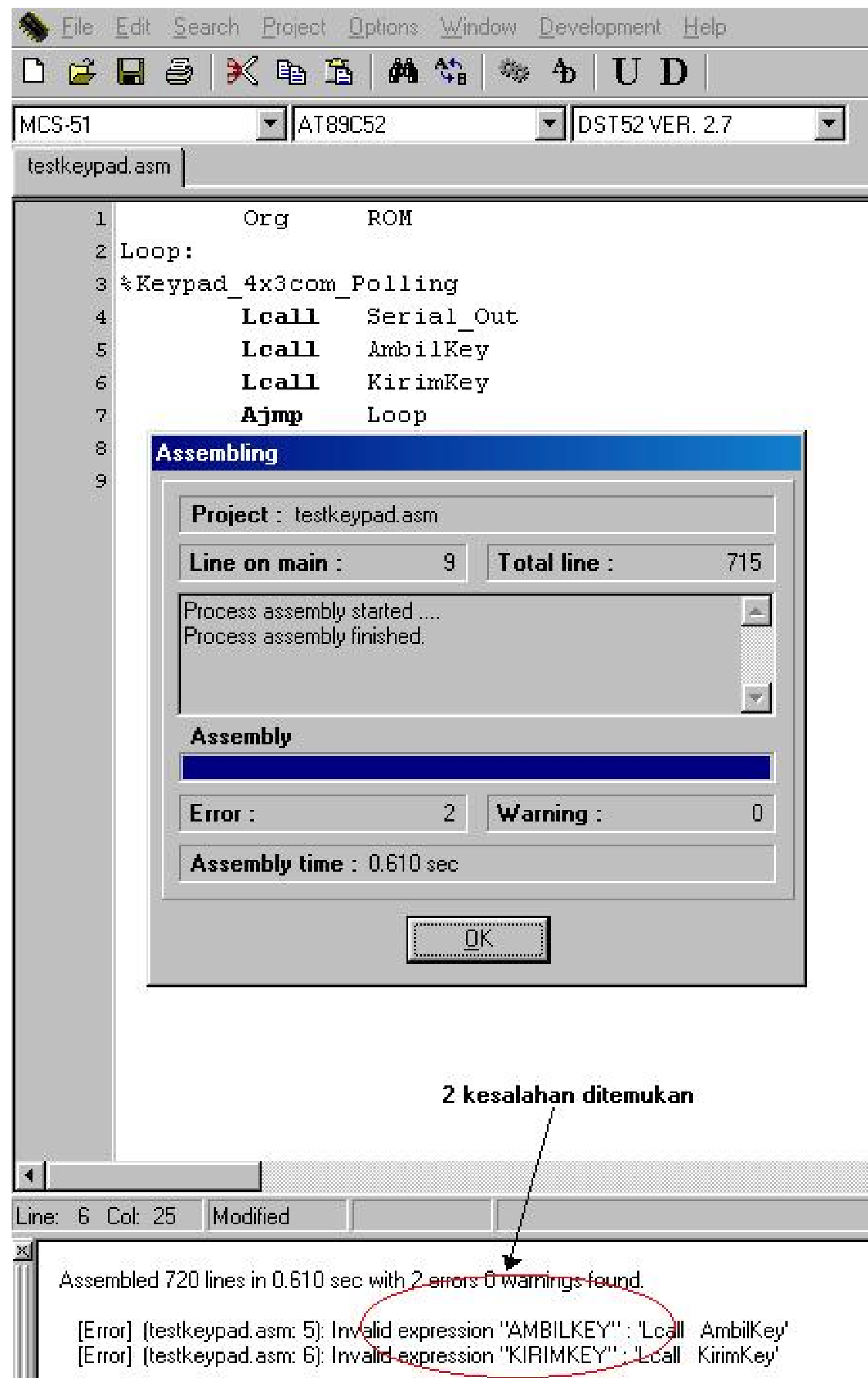


**Gambar 14 Setting Delta Studio**

### **Melacak kesalahan pada proses assembly**

Delta Studio memiliki fitur kemudahan dalam melacak kesalahan yang terjadi pada saat proses assembly di mana bentuk-bentuk kesalahan akan ditampilkan pada layar bagian bawah dari Delta Studio seperti pada gambar 15.

Posisi tepatnya dari kesalahan tersebut dapat ditemukan dengan mudah dengan melakukan double click pada baris tersebut dan Delta Studio akan menunjukkan baris dari daftar program yang mengalami kesalahan. Gambar 16.



**Gambar 15 Daftar Kesalahan Pada Proses Assembly**

The screenshot shows an IDE window titled 'testkeypad.asm'. The menu bar includes File, Edit, Search, Project, Options, Window, Development, and Help. The toolbar contains icons for file operations and development tools. The target configuration is set to MCS-51, AT89C52, and DST52 VER. 2.7. The assembly code is as follows:

```
1      Org      ROM
2 Loop:
3 %Keypad_4x3com_Polling
4      Lcall   Serial_Out
5      Lcall   AmbilKey
6      Lcall   KirimKey
7      Ajmp   Loop
8
9
```

The status bar indicates 'Line: 5 Col: 25 Modified'. The output window shows the following messages:

```
Assembled 720 lines in 0.610 sec with 2 errors 0 warnings found.
[Error] (testkeypad.asm: 5): Invalid expression "AMBILKEY": 'Lcall AmbilKey'
[Error] (testkeypad.asm: 6): Invalid expression "KIRIMKEY": 'Lcall KirimKey'
```

**Gambar 16 Posisi kesalahan pada Daftar Program**

### **Melacak kesalahan pada program yang telah didownload**

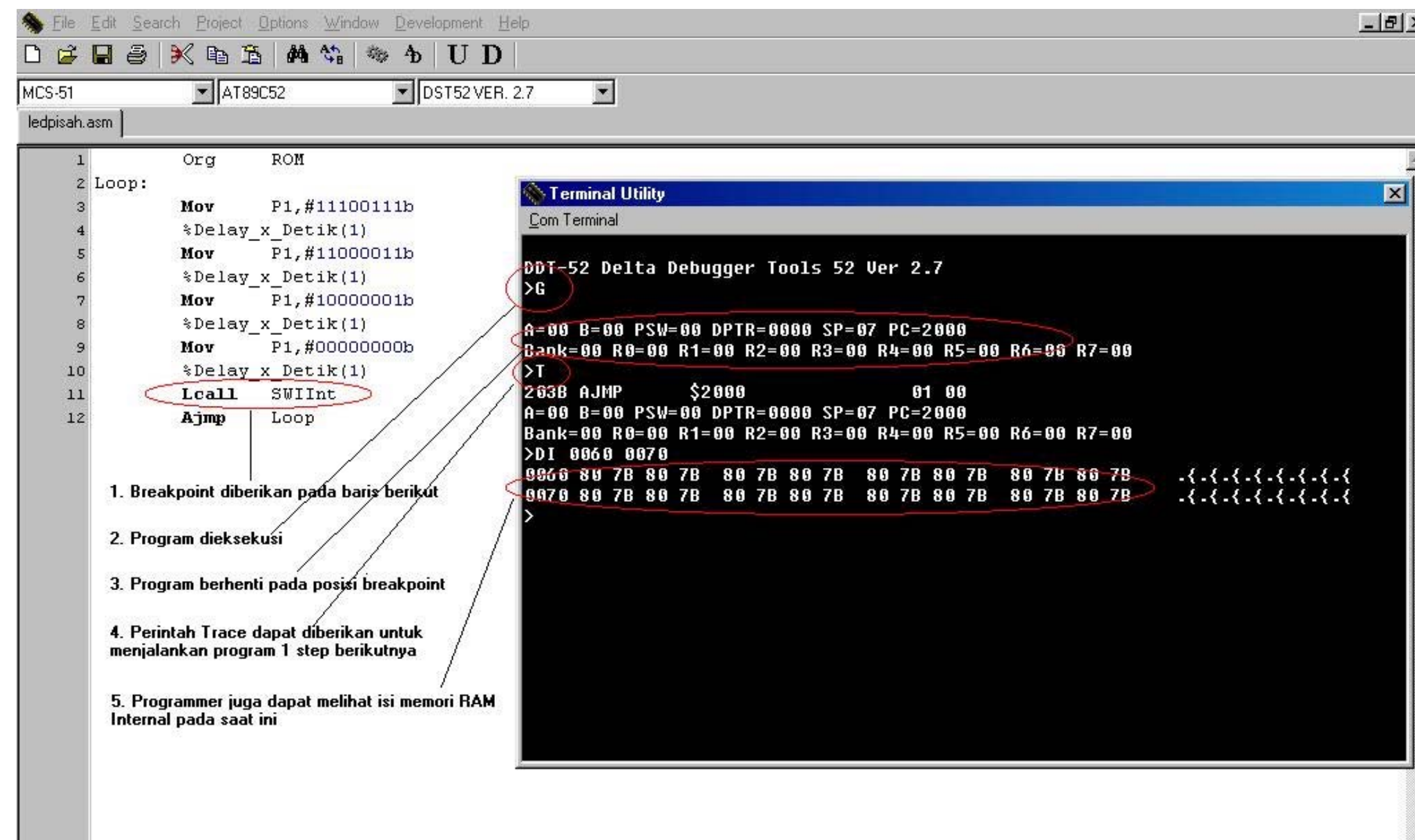
Proses ini hanya dapat dilakukan pada Modul DST-52 yang memiliki kernel DDT-52 Delta Debugger Tools 52 dan dilakukan pada Mode Monitor yaitu proses Realtime Debugging.

Sebagai contoh saat kita ingin mengetahui apakah aktifitas LED yang terhubung pada Port 1 sudah sesuai dengan keinginan kita. Proses ini dapat diketahui dengan memberikan Perintah Breakpoint pada lokasi-lokasi tertentu. Pada gambar 17 terlihat bahwa perintah ini diberikan setelah program memberikan logika 0 pada Port 1 yaitu setelah detik ketiga (Point 1).

Saat program dieksekusi/Run (Point 2), pembuat program dapat melihat pada LED yang terhubung pada Port 1 apakah aktifitas LED sudah sesuai

dengan keinginan. Fungsi Real Time pada proses debug ini akan membuat program benar-benar berhenti tepat pada detik ketiga dan menampilkan isi register maupun program counter saat tersebut (point 3).

Selanjutnya programmer juga dapat memajukan program ke step berikutnya dari program yang dihentikan tadi dengan fungsi Trace (point 4) atau melihat isi RAM Internal AT8951 pada saat itu (point 5).



Gambar 17 Breakpoint

## TOMBOL KONTROL PROGRAM DST-51 MONITOR

### D {aaaa} {aaaa}

Berfungsi untuk melihat isi memori eksternal (28C64) mulai dari alamat yang ditunjukkan oleh {aaaa} yang pertama dan {aaaa} kedua

Contoh:

D 2000 2100 <CR>

Monitor akan menampilkan isi dari alamat 2000H hingga 2100H dalam bentuk hexa

### DI {aaaa} {aaaa}

Berfungsi untuk melihat isi memori internal mulai dari alamat yang ditunjukkan oleh {aaaa} yang pertama dan {aaaa} kedua

Contoh:

DI 0030 0040 <CR>

Monitor akan menampilkan isi dari alamat 30H hingga 40H dalam bentuk hexa

**E {aaaa} <CR>**

Berfungsi untuk mengedit isi dari memori eksternal yang dimulai dari alamat {aaaa}

Contoh:

E 2000 <CR>

Monitor akan menampilkan isi dari alamat 2000 dan meminta anda memasukkan data baru ke alamat tersebut. Setelah penekanan <CR> atau enter maka alamat berikutnya akan ditampilkan untuk diedit juga.

**EI {aa} <CR>**

Berfungsi untuk mengedit isi dari memori internal yang ditunjuk oleh alamat {aa} sama seperti yang dilakukan oleh instruksi E

**F {aaaa} {aaaa} {dd} <CR>**

Berfungsi untuk mengisi data ke alamat yang ditunjukkan oleh {aaaa} yang pertama hingga {aaaa} kedua dengan data {dd}

Contoh:

F 2000 2100 FF <CR>

Maka data FF akan diisikan ke alamat 2000 hingga 2100 yang ada di memori eksternal

I

Maka akan ditampilkan informasi-informasi umum yang kami sertakan

?

Help, akan tampil daftar Tombol-tombol Kontrol Program DST-51 Monitor.

**TOMBOL KONTROL DDT-52 DEBUGGER TOOLS**

**D {aaaa} {aaaa}**

Berfungsi untuk melihat isi memori eksternal (28C64) mulai dari alamat yang ditunjukkan oleh {aaaa} yang pertama dan {aaaa} kedua

Contoh:

D 2000 2100 <CR>

Monitor akan menampilkan isi dari alamat 2000H hingga 2100H dalam bentuk hexa

**DI {aaaa} {aaaa}**

Berfungsi untuk melihat isi memori internal mulai dari alamat yang ditunjukkan oleh {aaaa} yang pertama dan {aaaa} kedua

Contoh:

DI 0030 0040 <CR>

Monitor akan menampilkan isi dari alamat 30H hingga 40H dalam bentuk hexa

**D8 [aaaa] [aaaa]**

Berfungsi untuk melihat isi memori Serial EEPROM 2400 - 2416 (Modul SEE-24)

D8 0030 0040 <CR>

**D16 [aaaa] [aaaa]**

Berfungsi untuk melihat isi memori Serial EEPROM 2432 – 24512 (Modul SEE-24)

D16 0030 0040 <CR>

**DL [aaaa] [aaaa]**

Berfungsi untuk melihat isi memori LCD Grafik PG24644

DL 0030 0040 <CR>

**E {aaaa} <CR>**

Berfungsi untuk mengedit isi dari memori eksternal yang dimulai dari alamat {aaaa}

Contoh:

E 2000 <CR>

Monitor akan menampilkan isi dari alamat 2000 dan meminta anda memasukkan data baru ke alamat tersebut. Setelah penekanan <CR> atau enter maka alamat berikutnya akan ditampilkan untuk diedit juga.

**EI {aaaa} <CR>**

Berfungsi untuk mengedit isi dari memori internal yang ditunjuk oleh alamat {aa} sama seperti yang dilakukan oleh instruksi E

**E8 {aaaa} <CR>**

Berfungsi untuk mengedit isi dari memori Serial EEPROM 2400-2416 yang ditunjuk oleh alamat {aaaa} sama seperti yang dilakukan oleh instruksi E

**E16 {aaaa} <CR>**

Berfungsi untuk mengedit isi dari memori Serial EEPROM 2432-24512 yang ditunjuk oleh alamat {aaaa} sama seperti yang dilakukan oleh instruksi E

**EL {aaaa} <CR>**

Berfungsi untuk mengedit isi dari memori LCD Grafik PG-24644 yang ditunjuk oleh alamat {aaaa} sama seperti yang dilakukan oleh instruksi E

**F {aaaa} {aaaa} {dd} <CR>**

Berfungsi untuk mengisi data ke alamat yang ditunjukkan oleh {aaaa} yang pertama hingga {aaaa} kedua dengan data {dd}

Contoh:

F 2000 2100 FF <CR>

Maka data FF akan diisikan ke alamat 2000 hingga 2100 yang ada di memori eksternal

**F8 {aaaa} {aaaa} {dd} <CR>**

Berfungsi untuk mengisi data ke alamat yang ditunjukkan oleh {aaaa} yang pertama dari Serial EEPROM 2400 – 2416 hingga {aaaa} kedua dengan data {dd}

Contoh:

F8 2000 2100 FF <CR>

Maka data FF akan diisikan ke alamat 2000 hingga 2100 yang ada di memori eksternal

**F16 {aaaa} {aaaa} {dd} <CR>**

Berfungsi untuk mengisi data ke alamat yang ditunjukkan oleh {aaaa} yang pertama



dari Serial EEPROM 2464 – 24512 hingga {aaaa} kedua dengan data {dd}

Contoh:

```
F16 2000 2100 FF <CR>
```

Maka data FF akan diisikan ke alamat 2000 hingga 2100 yang ada di memori eksternal

**FL {aaaa} {aaaa} {dd} <CR>**

Berfungsi untuk mengisi data ke alamat yang ditunjukkan oleh {aaaa} yang pertama LCD Grafik PG24644 hingga {aaaa} kedua dengan data {dd}

Contoh:

```
FL 0000 0100 FF <CR>
```

Maka data FF akan diisikan ke alamat 0000 hingga 0100 yang ada di memori LCD Grafik

## **BUILT IN ROUTINES DST-5x**

Built In Routines adalah rutin-rutin yang telah disediakan untuk mempermudah dan mempercepat pengguna dalam merancang program. Anda dapat melihat lebih detail mengenai rutin-rutin tersebut dalam CD yang ada dalam paket ini.

Built In Routines ini diaktifkan dengan memanggil rutin tersebut menggunakan perintah LCALL Nama\_Rutin

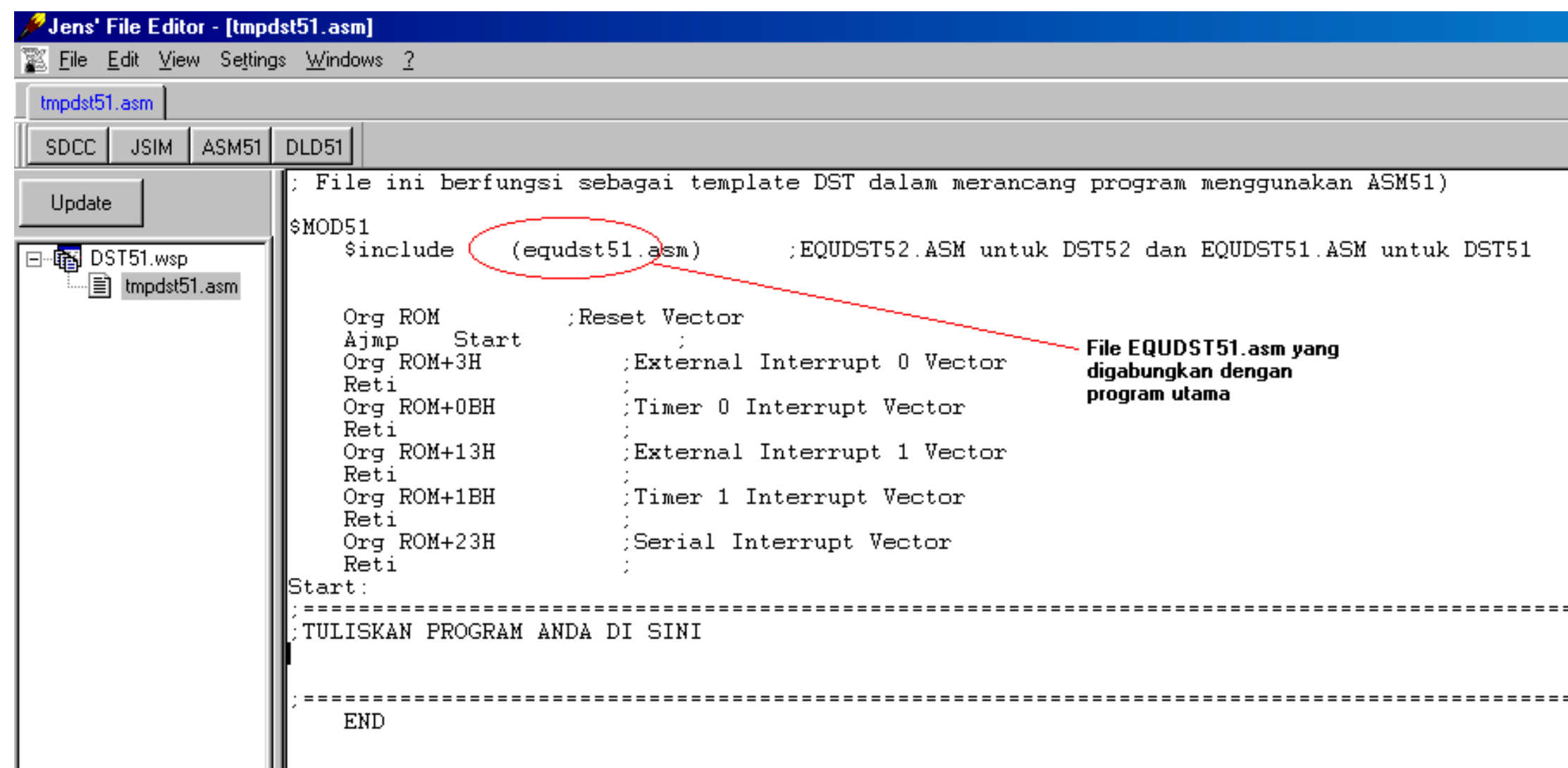
Pada awal program, built in routine tersebut harus di definisikan sebagai berikut:

```
Nama_Rutin EQU xxxx, di mana xxxx = alamat rutin
```

DST-51 dilengkapi dengan lebih dari 50 built in routine yang tersedia dalam Flash PEROM AT89C51

Agar pengguna tidak perlu mendeklarasikan lagi alamat-alamat ini, maka cukup dilakukan dengan menggabungkan file EQU DST51.asm (untuk DST-51) dan EQU DST52.asm (untuk DST-52) ke dalam program utama anda dengan fasilitas include.

File EQU DST51 maupun EQU DST52 sudah berisi deklarasi rutin-rutin ini beserta deskripsinya.



**Gambar 18 Penggabungan file EQU DST51.asm pada program utama**

Sedangkan pada Delta Micro Studio, pengguna hanya perlu mengatur template pada DST-52 V2.7 untuk Modul DST-52 atau DST-51 V2.6 untuk Modul DST-51.

### **Rutin Fungsi Umum**

HexASCII2 EQU 00FAH

Konversi Hexa ke ASCII, melakukan konversi dari bilangan Hexa pada Acc A menjadi ASCII di mana nibble bawah berada di Acc B dan nibble atas berada pada Acc A. Contoh: Acc A=1BH -> Acc B = 31H dan Acc A = 42H

HexASCII1 EQU 0106H

Konversi Hexa ke ASCII, sama dengan HexASCII2 namun hanya untuk nibble bawah. Contoh: Acc A = 21H -> Acc A = 31H (ASCII dari 01H)

ASCII\_Hex EQU 0113H

Konversi ASCII ke Hexa, melakukan konversi dari bilangan ASCII yang disimpan di Acc A dan Acc B menjadi Hexa di Acc A. Nibble atas berada di Acc A dan nibble bawah berada di Acc B. Contoh: Acc A = 32H dan Acc B = 31H -> Acc A = 21H

ASCII\_Hex1 EQU 0123H

Konversi ASCII ke Hexa untuk 1 byte. Contoh: Acc A = 41H -> Acc A = 0AH

Dec\_DPTR EQU 0130H

Rutin untuk decrement nilai DPTR. DPTR = DPTR - 1 (MCS51 hanya mempunyai instruksi Inc DPTR)

Capitalize EQU 013CH

Rutin untuk mengubah ASCII huruf kecil pada Acc A menjadi ASCII huruf besar pada Acc A di mana karakter-karakter lain seperti numerik, tanda baca dll tidak dikonversi

Space\_Code EQU 014EH

Rutin untuk mengeluarkan karakter spasi pada serial port

Writeext\_memory EQU 0157H

Rutin untuk menulis data pada Acc A ke alamat memori eksternal EEPROM AT28C64 yang ditunjuk oleh DPTR. Untuk penggunaan EEPROM AT28C64 sebagai memori eksternal, proses penulisan hanya dapat dilakukan dengan menggunakan rutin ini. Hal ini disebabkan karena hanya rutin program dari luar AT28C64 saja yang dapat menulis ke dalamnya di mana dalam hal ini berada pada Flash PEROM AT89C51

Contoh Penggunaan:

Writeext\_memory EQU 0157H

Org 2000H

Mov A,#01H

Mov DPTR,#2800H

Lcall Writeext\_memory

.....

Program ini akan menuliskan data 01H ke alamat 2800H yang ada pada AT28C64

### **Rutin-rutin delay dengan menggunakan crystal 11.0592 MHz**

Delay\_1detik EQU 015CH

Delay\_500mS EQU 0165H

Delay\_100mS EQU 016EH

Delay\_75mS EQU 0177H

Delay\_5mS EQU 0180H

ASCII\_Out EQU 0197H

**Rutin untuk mengirim nilai ASCII dari Acc A ke Serial Port**

**Out\_DPTR EQU 01A0H**

Rutin untuk mengirim nilai DPTR ke Serial Port

**Enter\_Code EQU 01ABH**

Rutin untuk mengirim karakter enter ke serial port

**Init\_Serial EQU 01B8H**

Rutin untuk inisialisasi serial port 9600 bps, 8 data

**Serial\_Out EQU 01C8H**

Rutin untuk mengirimkan nilai Acc A ke serial port

**Serial\_In EQU 01D0H**

Rutin untuk mengambil data dari serial port dan menyimpan di Acc A

**KirimPesan\_Serial EQU 01D8H**

Rutin untuk mengirimkan karakter serial mulai dari alamat yang ditunjuk oleh DPTR ke serial port hingga ditemukan karakter 0FH

Contoh Penggunaan:

```
Mov DPTR,#Pesan
Lcall KirimPesan_Serial
```

.....

Pesan:

```
DB 'Delta Electronic',0FH
```

**Filter EQU 01E4H**

Rutin untuk memastikan bahwa data di Acc A berada di antara nilai R7 sebagai nilai maksimum dan R6 sebagai nilai minimum

Contoh Penggunaan:

```
Mov R7,#10H
Mov R6,#08H
Lcall Filter
Jc Diluar_Batas
```

Dalam\_Batas:

.....

Diluar\_Batas:

.....

Bila nilai Acc A berada antara 08H hingga 10H maka carry flag akan clear dan bila nilai Acc A berada di luar 08H hingga 10H maka carry flag akan set.

#### **Rutin-rutin yang berhubungan dengan interface**

HD44780

**GeserDisplay\_Kanan EQU 0208H**

Menggeser tampilan LCD HD44780 ke kanan

**GeserDisplay\_Kiri EQU 020FH**

Menggeser tampilan LCD HD44780 ke kiri

**PosisiAwal\_LCD EQU 0216H**

Memindah posisi cursor LCD ke posisi awal

**GeserCursor\_Kiri EQU 021DH**

Memindah posisi cursor LCD ke kiri

**GeserCursor\_Kanan EQU 0224H**

Memindah posisi cursor LCD ke kanan

**KirimPesan\_LCD EQU 022BH**

Sama dengan rutin KirimPesan\_Serial namun hasil ditampilkan pada LCD HD44780

**Init\_LCD EQU 0237H**

Inisial LCD HD44780

**Kirim\_Perintah EQU 0276H**

Rutin-rutin untuk mengirim perintah (instruksi) ke LCD HD44780. Digunakan untuk fungsi-fungsi spesial LCD.

**Kirim\_Karakter EQU 0295H**

Rutin-rutin untuk mengirim karakter yang tersimpan di Acc A ke LCD HD44780 (Karakter dalam ASCII)

**Rutin-rutin untuk menentukan baris LCD yang aktif**

**Baris2\_LCD EQU 029FH**  
**Baris1\_LCD EQU 02A4H**

**Keypad (digunakan untuk Keypad Interface KP-43865)**  
**Ambil\_Keypad EQU 02C3H**

**TestPPI EQU 0340H**

Rutin untuk mengirimkan logika 0 ke Port PPI dan menggeser logika tersebut pada port. Rutin ini digunakan untuk menguji port PPI dengan bantuan LED Tester (Rangkaian yang terdiri dari 8 buah LED dan connector ke Port I/O DST-51)

**Tulis\_SEE EQU 0397H**

Rutin untuk menulis data ke Serial EEPROM 24C00 - 24C16. Rutin ini hanya dapat digunakan bila anda menggunakan IC Serial EEPROM yang terpasang pada posisi

- SDA (Serial EEPROM) P2.0
- SCL (Serial EEPROM) P2.1

Data yang ditulis disiapkan di Akumulator

Device Address di R7

Word Address di Register B

Contoh Penggunaan:

```
Mov    A,#05H      ;Data yang ditulis adalah 5
Mov    B,#00H      ;Word Address = 0
Mov    R7,#0A0H    ;Slave Address di alamat A0
Lcall  Tulis_SEE   ;Data 5 akan tersimpan di alamat 0 dari Serial
EEPROM
```

**Baca\_SEE EQU 03C9H**

Device Address di R7

Word Address di Register B

Hasil tersimpan di Akumulator

```
Mov    B,#00H      ;Word Address = 0
Mov    R7,#0A0H    ;Slave Address di alamat A0
Lcall  Baca_SEE    ;Data di alamat 0 dari Serial EEPROM dibaca dan
hasilnya
;tersimpan di Akumulator
```

### **Rutin-rutin LCD Grafik**

#### **InitLCDGRP EQU 0486H**

Pada rutin ini, modul LCD diinisialisasi sebagai berikut:

- Grafik Home alamat 1E0H
- Grafik Area sebesar 40 kolom
- Teks Area sebesar 40 kolom
- Teks Home di alamat 00H
- EXOR antara Grafik dan Teks
- Cursor 1 baris
- Grafik ON
- Teks ON

#### **SetTextHome EQU 04BEH**

Rutin ini digunakan untuk mengatur posisi awal teks. Input dari rutin ini adalah register A dan B di mana register A untuk byte rendah dan B untuk byte tinggi.

Contoh:

```
Mov  A,#01H
Mov  B,#02H
Acall SetTextHome
```

Pada contoh berikut, posisi awal teks akan menempati alamat 0201H dari modul LCD

#### **SetGrafikHome EQU 04C4H**

Rutin ini digunakan untuk mengatur posisi awal grafik. Input dari rutin ini adalah register A dan B di mana register A untuk byte rendah dan B untuk byte tinggi.

Contoh:

```
Mov  A,#01H
Mov  B,#02H
Acall SetGrafikHome
```

Pada contoh berikut, posisi awal grafik akan menempati alamat 0201H dari modul LCD

#### **SetGrafikArea EQU 04CAH**

Rutin ini digunakan untuk mengatur area grafik. Input dari rutin ini adalah register A dan B di mana register A untuk byte rendah dan B untuk byte tinggi.

Contoh:

```
Mov  A,#040
Mov  B,#00
```

Acall SetGrafikArea

Pada contoh berikut, grafik akan menempati area sebesar 40 kolom

**SetTextArea EQU 04D0H**

Rutin ini digunakan untuk mengatur area teks. Input dari rutin ini adalah register A dan B di mana register A untuk byte rendah dan B untuk byte tinggi.

Contoh:

Mov A,#030

Mov B,#00

Acall SetTextArea

Pada contoh berikut, text akan menempati area sebesar 30 kolom

**TextON EQU 04E0H**

Rutin ini digunakan untuk mengaktifkan mode teks

**GrafikON EQU 04E9H**

Rutin ini digunakan untuk mengaktifkan mode grafik

**CursorONBlinkON EQU 04F2H**

Rutin ini digunakan untuk mengaktifkan cursor dan blinking

**CursorONBlinkOFF EQU 0513H**

Rutin ini digunakan untuk mengaktifkan cursor tanpa blinking

**TextOFF EQU 051CH**

Rutin ini digunakan untuk menonaktifkan mode teks

**GrafikOFF EQU 0525H**

Rutin ini digunakan untuk menonaktifkan mode grafik

**KirimPerintah EQU 0548H**

Rutin ini digunakan untuk mengirimkan data ke register perintah dari LCD

**TungguStatus1EQU 0575H**

Rutin ini digunakan untuk menunggu status LCD apakah modul tersebut telah siap untuk menerima data lebih lanjut setelah proses penulisan atau pembacaan data dilakukan

**TungguStatusAWR EQU 058BH**



Rutin ini digunakan untuk menunggu status LCD apakah modul tersebut telah siap untuk menerima data lebih lanjut setelah proses penulisan secara otomatis dilakukan

### **TungguStatusAR EQU 05A1H**

Rutin ini digunakan untuk menunggu status LCD apakah modul tersebut telah siap untuk menerima data lebih lanjut setelah proses pembacaan secara otomatis dilakukan

### **SetPolaCursor EQU 05B7H**

Rutin ini digunakan untuk mengatur pola cursor.

Contoh:

```
Mov  A,#0A7H      ; Cursor 8 baris
Acall SetPolaCursor ;
```

### **SetPosisiCursor EQU 05C0H**

Rutin ini digunakan untuk mengatur posisi cursor dengan input dari register A dan B

Contoh:

```
Mov  A,#00H
Mov  B,#01H
Acall SetPosisiCursor
```

Cursor akan menempati kolom pertama baris kedua dari modul LCD

### **SetPointerAlamat EQU 05C7H**

Rutin ini digunakan untuk mengatur posisi alamat data LCD yang akan diakses baik dalam mode grafik maupun teks. Input dari rutin ini adalah register A dan B

Contoh:

```
Mov  A,#00H
Mov  B,#20H
Acall SetPointerAlamat
```

Akses akan dilakukan ke alamat 2000H dari modul LCD tersebut

### **HapusMemori EQU 05CEH**

Rutin ini digunakan untuk menghapus seluruh isi data pada memori LCD

### **KirimKarakterLCD EQU 05FDH**

Rutin ini digunakan untuk menampilkan karakter ke layar LCD di mana input dari rutin ini adalah R7 dalam bentuk ASCII. Sebelum rutin ini diaktifkan maka pointer alamat harus diatur terlebih dahulu.

Contoh:

```
Mov  A,#01H
Mov  B,#00H
Acall SetPointerAlamat
Mov  R7,#'A'
Acall KirimKarakterLCD
```

Karakter A akan tampil pada kolom dua baris pertama modul LCD

### **KirimPesanLCD EQU 0609H**

Rutin ini berfungsi untuk mengirimkan pesan dari suatu lokasi memori yang ditunjuk oleh DPTR dan diakhiri dengan data 0 ke layar LCD.

Contoh:

```
Mov  DPTR,#Pesan
Acall KirimPesanLCD
.....
```

Pesan:

```
DB   'Delta Electronic,'00H
```

### **Rutin Akses PC Hyperterminal**

#### **GetBitmapHPT EQU 06C0H**

Rutin ini berfungsi untuk menunggu kiriman file bitmap monochrome dari Hyperterminal PC

### **BUILT IN ROUTINE DST-52 (BIOS versi 2.7)**

#### **Hex\_ASCII2 EQU 002EH**

```
;Konversi Hexa ke 2 byte ASCII
;- Akumulator diisi bilangan hexa yang akan
dikonversi
;Hasil:
;- Akumulator menyimpan nibble atas
;- Register B menyimpan nibble bawah
```

```

Hex_ASCII1                EQU  003AH
;Konversi Hexa ke 1 byte ASCII
;- Akumulator diisi bilangan hexa yang akan
dikonversi
;Hasil:
;- Akumulator menyimpan nibble bawah

ASCII_HEX                 EQU  0047H
;Konversi 2 byte ASCII ke hexa
;- Akumulator diisi nibble atas
;- Register B diisi nibble bawah
;Hasil:
;- Akumulator menyimpan hasil konversi

Dec_DPTR                  EQU  0064H
;Mengurangi isi dari Register DPTR

Capitalize                EQU  0070H
;Mengubah semua jenis huruf kecil menjadi huruf besar

SPACE_CODE                EQU  0082H
;Mengirim kode spasi ke port serial

Writeext_Memory           EQU  008BH
;Menulis data ke memori EEPROM
;- Akumulator berisi data yang akan ditulis
;- DPTR berisi alamat dari data yang akan ditulis

Delay_1detik              EQU  0090H
;Menunda waktu selama 1 detik (timer 0 digunakan)

Delay_500mS               EQU  0099H
;Menunda waktu selama 500 mS (timer 0 digunakan)

Delay_100mS               EQU  00A2H
;Menunda waktu selama 100 mS (timer 0 digunakan)

Delay_75mS                EQU  00ABH
;Menunda waktu selama 75 mS (timer 0 digunakan)

```

```

Delay_5mS                EQU  00B4H
;Menunda waktu selama 5 mS (timer 0 digunakan)

ASCII_Out                EQU  00CBH
;Mengirim data ke port serial dalam 2 byte ASCII
;- Akumulator diisi dengan data yang akan dikirim

Out_DPTR                 EQU  00D4H
;Mengirim nilai DPTR dalam bentuk 4 byte ASCII ke
port serial

Enter_Code               EQU  00DFH
;Mengirim kode <CR> ke port serial

Init_Serial              EQU  00ECH
;Inisialisasi Port Serial 9600 bps

Serial_Out               EQU  00FCH
;Kirim nilai akumulator ke port serial

Serial_In                EQU  0104H
;Ambil data dari port serial dan simpan di akumulator

KirimPesan_Serial       EQU  010CH
;Kirim data di alamat yang ditunjuk oleh DPTR hingga
data 0F
;- DPTR diisi dengan alamat dari data yang akan
dikirim
;- Memori yang diakses adalah memori CODE/PROGRAM

Filter                   EQU  0118H
;Menentukan besaran data apakah berada di dalam atau
di luar nilai setpoint
;- R7 diisi nilai max
;- R6 diisi nilai min
;- besaran data di akumulator
;Hasil:

```

;- Carry flag akan set bila besaran berada di luar nilai setpoint

;=====

; SERIAL EEPROM

;=====

**Tulis\_SEE16b** **EQU 01A0H**

;Menulis data ke Serial EEPROM 16 bit

;- R7 diisi Device Address

;- Register B diisi Word Address 1

;- R6 diisi Word Address 2

;- Akumulator diisi data yang akan ditulis

**Tulis\_SEE8b** **EQU 01B7H**

;Menulis data ke Serial EEPROM 8 bit

;- R7 diisi Device Address

;- Register B diisi Word Address 1

;- Akumulator diisi data yang akan ditulis

**Baca\_SEE16b** **EQU 01CEH**

;Membaca data ke Serial EEPROM 16 bit

;- R7 diisi Device Address

;- Register B diisi Word Address 1

;- Akumulator berisi data yang telah dibaca

**Baca\_SEE8b** **EQU 01E5H**

;Membaca data ke Serial EEPROM 8 bit

;- R7 diisi Device Address

;- Register B diisi Word Address 1

;- Akumulator berisi data yang telah dibaca

**PageSEE8bWrite** **EQU 027CH**

;Menulis data ke Serial EEPROM 8 bit secara page mode (Flag F0 digunakan)

;- R7 diisi Device Address

;- Register B diisi Word Address 1

;- Akumulator diisi data yang akan ditulis

```

PageSEE16bWrite                EQU  0285H
;Menulis data ke Serial EEPROM 16 bit secara page
mode (Flag F0 digunakan)
;- R7 diisi Device Address
;- Register B diisi Word Address 1
;- Akumulator diisi data yang akan ditulis

TulisDPTRSEE8b                  EQU  02F2H
;Menulis data ke Serial EEPROM dengan menggunakan
DPTR sebagai pointer
;- DPTR diisi alamat yang akan ditulis
;- Akumulator diisi data yang akan ditulis

TulisDPTRSEE16b                EQU  02F7H
;Menulis data ke Serial EEPROM dengan menggunakan
DPTR sebagai pointer
;- DPTR diisi alamat yang akan ditulis
;- Akumulator diisi data yang akan ditulis

DATA_SEE    EQU  3AH
;Konstanta Data_SEE terletak pada alamat 3AH RAM
Internal

Perbandingan16bit              EQU  0396H
;Membandingkan data 16 bit
; DPTR = PTR, A = 0
; DPTR < PTR, C = 1
; DPTR > PTR, C = 0
;Untuk Built In Routine
; PTR+1 = 3EH
; PTR = 3FH
;=====
;HD44780
;=====

GeserDisplay_Kanan             EQU  05BDH
;Menggeser tampilan LCD HD44780 ke kanan

GeserDisplay_Kiri             EQU  05C4H
;Menggeser tampilan LCD HD44780 ke kiri

```

```

Posisi_Awal EQU 05CBH
;Mengatur posisi cursor LCD ke posisi awal

GeserCursor_Kiri EQU 05D2H
;Mengeser Cursor LCD ke kiri

GeserCursor_Kanan EQU 05D9H
;Mengeser Cursor LCD ke kanan

KirimPesan_LCD EQU 05E0H
;Mengirim data di alamat yang ditunjuk oleh DPTR ke
LCD hingga data 0F
;- DPTR diisi dengan alamat awal data yang dikirim
;- Akhir data adalah 0FH

Init_LCD EQU 05ECH
;Inisialisasi LCD

Kirim_Perintah EQU 062BH
;Mengirim data ke register perintah LCD
;- Data diisi di akumulator

Kirim_Karakter EQU 064AH
;Mengirim data ke register data LCD
;- Data diisi di akumulator

Baris2 EQU 0654H
;Memindah posisi cursor ke baris 2

;===== LCD GRAFIK =====
InitLCDGRP EQU 0678H
;Inisialisasi LCD Grafik

SetTextHome EQU 06B7H
;Mengatur posisi awal teks
;- Akumulator diisi nibble tinggi alamat memori text
;- Register B diisi nibble rendah alamat memori text

```

```

SetGrafikHome                EQU    06BDH
;Mengatur posisi awal tampilan grafik
;- Akumulator diisi nibble tinggi alamat memori
grafik
;- Register B diisi nibble rendah alamat memori
grafik

SetGrafikArea                EQU    06C3H
;Mengatur area tampilan Grafik
;- Akumulator diisi baris dari LCD grafik
;- Register B diisi kolom dari LCD Grafik

SetTextArea                  EQU    06C9H
;Mengatur area tampilan Text
;- Akumulator diisi jumlah baris text
;- Register B diisi jumlah kolom text

TextON                       EQU    06DBH
;Mengaktifkan tampilan teks

GrafikON                     EQU    06E4H
;Mengaktifkan tampilan grafik

CursorONBlinkON              EQU    06EDH
;Mengaktifkan Cursor dan blinking

CursorONBlinkOFF            EQU    0710H
;Mengaktifkan cursor dan no blink

TextOFF                      EQU    071AH
;Menonaktifkan teks

GrafikOFF                    EQU    0724H
;Menonaktifkan grafik

TulisDataLCD                 EQU    072EH
;Menulis data ke memori LCD grafik
;- Register B diisi nibble tinggi dari alamat memori
LCD

```



;- Akumulator diisi nibble rendah dari alamat memori  
LCD  
;- Panggil subroutine SetPointerAlamat  
;- R7 diisi data yang akan ditulis

**BacaDataLCD** **EQU 0742H**

;Membaca data dari memori LCD Grafik  
;- Register B diisi nibble tinggi dari alamat memori  
LCD  
;- Akumulator diisi nibble rendah dari alamat memori  
LCD  
;- Panggil subroutine SetPointerAlamat

**KirimPerintah** **EQU 0752H**

;Mengirim data ke register perintah dari LCD Grafik  
;==== DATA-DATA PERINTAH =====  
TXHOME EQU 40H  
TXAREA EQU 41H  
GRHOME EQU 42H  
GRAREA EQU 43H  
SETCURSOR EQU 21H  
OFFSET EQU 22H  
ADPSET EQU 24H  
AWRON EQU 0B0H  
AWROFF EQU 0B2H  
ARON EQU 0B1H  
AROFF EQU AWROFF

**TungguStatus1** **EQU 077AH**

;Menunggu status penulisan data ke LCD grafik selesai

**TungguStatusAWR** **EQU 0790H**

;Menunggu status penulisan data secara otomatis ke  
LCD Grafik selesai

**TungguStatusAR** **EQU 07A6H**

;Menunggu status membaca data secara otomatis ke LCD  
Grafik selesai

**SetPolaCursor** EQU 07BCH

;Mengatur pola cursor LCD

Cursor1Line EQU 0A0H

Cursor2Line EQU 0A1H

Cursor3Line EQU 0A2H

Cursor4Line EQU 0A3H

Cursor5Line EQU 0A5H

Cursor6Line EQU 0A6H

Cursor7Line EQU 0A7H

**SetPosisiCursor** EQU 07C6H

;Mengatur posisi cursor LCD

;- Register B diisi baris

;- Akumulator diisi kolom

**SetPointerAlamat** EQU 07CFH

;- Register B diisi alamat nibble tinggi

;- Akumulator diisi alamat nibble rendah

**HapusMemori** EQU 07D8H

;Menghapus memori teks dan memori grafik

CommandKarakter EQU 41H

; Konstanta Command Karakter di RAM Internal alamat  
4FH

**KirimKarakterLCD** EQU 080AH

;Mengirim karakter ke LCD Grafik

;- R7 diisi data karakter

;- Akumulator diisi alamat nibble tinggi

;- Register B diisi alamat nibble rendah

;- Panggil subroutine set pointer alamat

;- Tulis CommandKarakter dengan data

TulisADPplus EQU 0C0H

BacaADPplus EQU 0C1H

TulisADPminus EQU 0C2H

BacaADPminus EQU 0C3H

TulisADP EQU 0C4H  
BacaADP EQU 0C5H

CMDP EQU 8200H  
DP EQU 8000H

**KirimPesanLCD EQU 0817H**

;Mengirim data di alamat yang ditunjuk oleh DPTR ke  
LCD hingga data 0F

- ;- DPTR diisi dengan alamat awal data yang dikirim
- ;- Akhir data adalah 0FH
- ;- Data di memori data (sinyal RD)

**BacaMemoriLCD EQU 0831H**

;Membaca data dari memori LCD dengan DPTR sebagai  
pointer

- ;- DPTR sebagai pointer alamat
- ;- Data yang dibaca disimpan di akumulator

**TulisMemoriLCD EQU 084BH**

;Menulis data dari memori LCD dengan DPTR sebagai  
pointer

- ;- DPTR sebagai pointer alamat
- ;- Akumulator sebagai data yang akan ditulis

**AmbilScanCode EQU 0865H**

;Mengambil 1 byte scan code keyboard

- ;- Nilai scan code berada pada akumulator

**KirimScanCode EQU 08A3H**

;Mengirim 1 byte scan code keyboard

- ;- Nilai scan code berada pada akumulator

**InitKeyboard EQU 08C4H**

;Inisialisasi Keyboard

**CapslockLED EQU 08DDH**

Mengaktifkan LED Caps Lock Keyboard

**SWIInt****EQU 0F41H**

Fasilitas yang mirip dengan breakpoint, namun dalam hal ini breakpoint tidak ditentukan pada mode terminal, melainkan dengan menyisipkan perintah pemanggilan terhadap program di alamat yang ditunjuk SWIInt dan terbentuk breakpoint baru saat program diisikan ke mikrokontroler.

Dapatkan informasi lebih detail atau konsultasi gratis mengenai DST-51 dan teknik-teknik interfacing di website kami

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

DST5x is Trademark by **Delta Electronic**

Delta Studio is Trademark by **Delta Electronic**

ASM51 is Trademark by **Metalink Corporation**

## **History Update**

### **Program Monitor:**

#### HISTORY

28 September 2001 : Penyempurnaan Mode Stand Alone  
(Versi 2.0)

2 Oktober : Penyempurnaan fasilitas Dump dan Dump  
Internal

: - Penambahan kecepatan fasilitas Fill  
External

Memori (khusus AT28C64)

: - Penambahan rutin konversi hexa ke ASCII  
untuk 1 nibble (Versi 2.1)

13 Oktober : Penyempurnaan rutin pulse receive dengan  
anti bouncing (Versi 2.2)

1 Desember 2001 : - Perbaiki program keypad

- Ganti SED1200 dg HD44780

- Perbaiki program keyboard

- Penambahan rutin delay 5 mS dan delay 1 detik (Versi  
2.21)

30 Desember 2001:

- Penambahan rutin delay

- Rubah sistem Built In Routine untuk kit dengan  
alamat yg dapat diatur sebelum dipanggil

- Call Progress Detector untuk DF-88 (Versi 2.22)

2 Juni 2002:

- Penambahan rutin Serial EEPROM (Versi 2.3)

30 Juni 2002:

- LCD M1632 kompatibel DST-51 2.1 (Versi 2.31)

18 Juli 2002:

- Penambahan rutin memori testing (Versi 2.32)

5 April 2003:

- Penambahan rutin LCD Grafik

- Penambahan rutin Test PPI

20 April 2003: DST51 2.4

- Rutin Keyboard keluar

- Penambahan rutin pengambilan file dari hyperterminal

25 Agustus 2003:

- DST-52 dengan program DDT-52 diluncurkan

### **Board Module:**

12 Juni 2001 DST-51 Versi 2.0

- Keyboard Connector
- On Cable RS232
- 2 LCD SED1200 Port
- AVR90S8515 Mode

7 Juli 2002 DST-51 Versi 2.1

- Hitachi & Other M1632 LCD Port

23 Desember 2007 DST-51 V3.7

- LED Logic Tester
- I2C Serial EEPROM