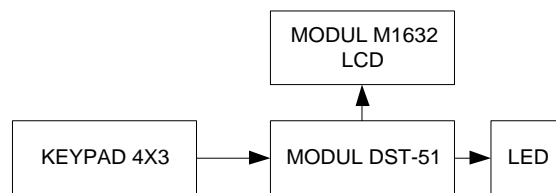


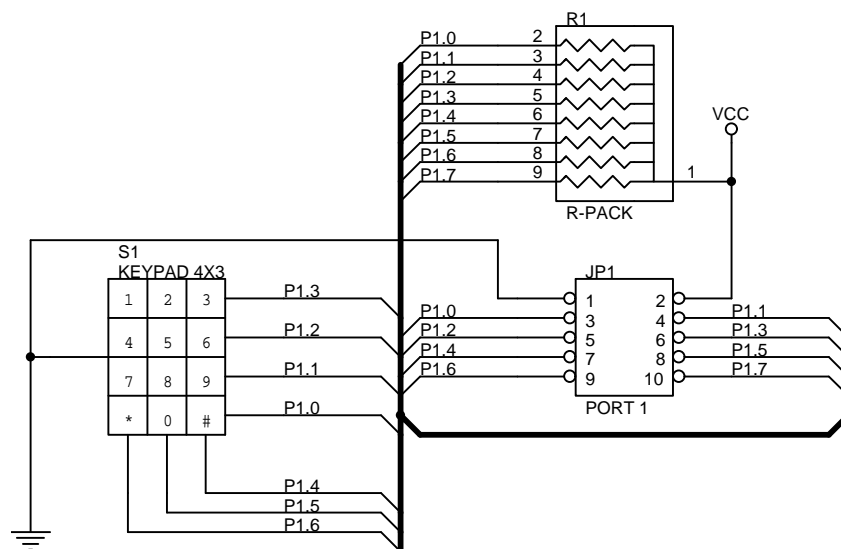
PROGRAMMABLE TIMER DENGAN TAMPILAN M1632 LCD MENGGUNAKAN MODUL DST-51

Perangkat timer adalah merupakan sebuah perangkat yang seringkali digunakan untuk sebuah sistem elektronik. Artikel berikut ini akan menjelaskan bagaimana sebuah perangkat *programmable timer* (perangkat timer yang dapat diatur) dibangun dengan menggunakan Modul DST-51. Pemrograman timer dilakukan secara digital dengan menggunakan keypad 4x3 dengan tampilan Modul LCD M1632.



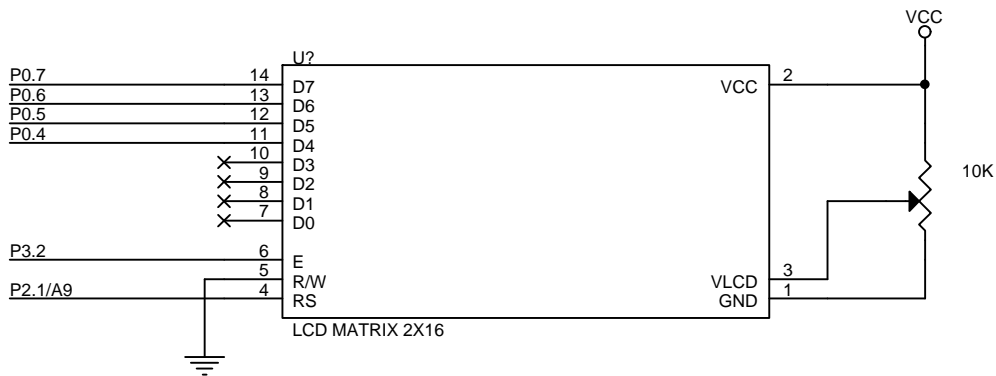
Gambar 1
Blok Diagram Programmable Timer

Keypad 4x3 yaitu keypad dengan 12 tombol dengan konfigurasi 4 baris dan 3 kolom berfungsi sebagai pemberi masukan data pada timer untuk proses pengaturan waktu. Data waktu yang dimasukkan akan tampil pada layar LCD dan timer akan segera bekerja menghitung mundur. Pada saat perhitungan timer mencapai nilai 0 (time out), maka LCD akan menampilkan pesan time out. Selain pesan time out, Modul DST-51 juga akan mengaktifkan LED yang terhubung pada P3.2/INT0 di mana pada aplikasinya, I/O ini juga dapat digunakan untuk terhubung dengan rangkaian pengendali triac, relay atau peralatan-peralatan lainnya.



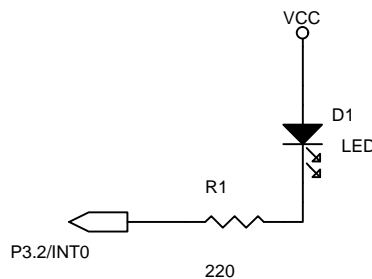
Gambar 2
Antar Muka Keypad 4x3

Gambar 2 menunjukkan bagian antar muka keypad lengkap dengan konektor penghubung ke DST-51 (JP1) yang berupa amphenol 5x2. Keypad yang digunakan dalam hal ini adalah **Keypad 4x3 lengkap dengan common** seperti yang pernah dibahas pada **Teknik Antar Muka Keypad 4x3 dengan DST-51**. Pada gambar 2 tampak common dari keypad tersebut terhubung ke ground.



Gambar 3 menunjukkan bagian antar muka Modul LCD M1632 dengan DST-51 di mana proses antar muka dilakukan dengan mode antar muka 4 bit. Potensio 10K yang terhubung pada kaki nomor 3 VLCD berfungsi sebagai pengatur kontras dari layar LCD.

Gambar 3
Antar Muka M1632 LCD



Gambar 4
Antar Muka LED

Gambar 4 adalah LED yang terhubung ke P3.2/INT0 melalui R1 sebagai pembatas arus. Pada saat timer selesai menghitung, maka kondisi logika P3.2/INT0 akan berlogika 0 sehingga arus mengalir dari VCC melalui LED D1 menuju ke P3.2/INT0 melalui R1 dan mengaktifkan LED tersebut. Bagian ini pada aplikasinya, juga dapat dihubungkan pada rangkaian pengendali relay, triac atau perangkat-perangkat lain yang akan dikontrol oleh sistem timer ini.

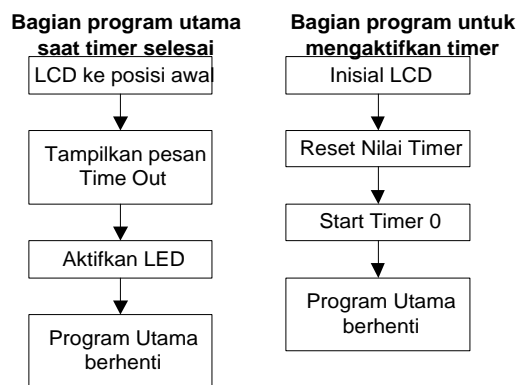
Pada bagian perangkat lunak, secara garis besar terdiri dari 2 bagian yaitu program utama dan program layanan interupsi. Sedangkan program utama sendiri juga terdiri dari bagian untuk mengaktifkan timer dan bagian yang bekerja saat timer selesai menghitung. Pada bagian untuk mengaktifkan timer, proses dimulai dengan inisialisasi LCD dan Reset Nilai Timer di mana pada bagian Reset Nilai Timer, nilai-nilai waktu

tersebut akan dimasukkan melalui Keypad 4x3 dan tampilan LCD M1632. Setelah nilai timer terisi sesuai keinginan pengguna, maka timer diaktifkan sehingga proses penghitungan waktu terjadi. Proses penghitungan waktu dilakukan dengan menggunakan layanan interupsi timer 0 (gambar 6).

Layanan interupsi berjalan secara terpisah (*independent*) sehingga program utama dapat berhenti saja pada satu lokasi tertentu ataupun menjalankan aplikasi lain selama aplikasi tersebut tidak menggunakan timer 0 dari AT8951.

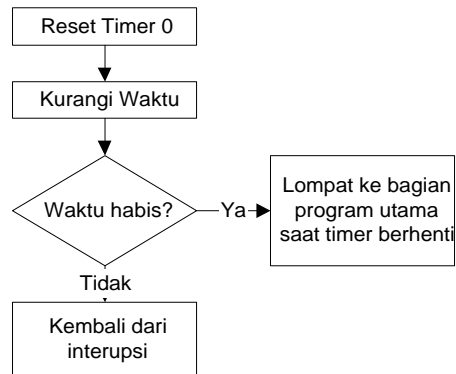
Apabila timer telah selesai menghitung, maka program akan masuk ke **bagian program utama saat timer selesai** di mana pesan “time out” ditampilkan ke LCD dan LED aktif. Pada aplikasi ini, memang terlihat program berhenti setelah LED aktif, namun pengguna juga dapat memodifikasi dengan menambahkan program lain yang bekerja setelah itu.

Contohnya apabila pengguna ingin mengirimkan pesan “time out” tersebut ke port serial, maka pengguna dapat menambahkan program pengiriman data ke port serial di antara bagian Aktifkan LED dan Program Utama Berhenti. Rutin pengiriman pesan melalui port serial pada Modul DST-51 dapat dilakukan dengan memanggil rutin siap pakai **KirimPesan_Serial**.



Gambar 5
Diagram Alir Utama

Proses penghitungan timer dilakukan pada bagian layanan interupsi dengan proses perhitungan mundur dari nilai awal yang sebelumnya telah dimasukkan melalui keypad. Selama penghitungan waktu belum selesai, maka program akan kembali dari interupsi. Artinya kembali ke lokasi di mana proses interupsi terjadi dan melanjutkan jalannya program di sana sampai interupsi berikutnya terjadi. Namun apabila proses penghitungan waktu telah selesai, maka program akan melompat ke bagian dari program utama saat timer selesai.



Gambar 6
Diagram Alir Layanan Interupsi

Program Utama:

```

$MOD51
;*****
; PROGRAM UTAMA TIMER
;*****

ROM EQU 2000H

Org ROM ;Reset Vector
Ajmp Start ;
Org ROM+3H ;External Interrupt 0 Vector
Reti ;
Org ROM+0BH ;Timer 0 Interrupt Vector
Ajmp Timer0_Interrupt
Org ROM+13H ;External Interrupt 1 Vector
Reti ;
Org ROM+1BH ;Timer 1 Interrupt Vector
Reti ;
Org ROM+23H ;Serial Interrupt Vector
Reti ;

Start:
Lcall Init_LCD ;Inisial LCD
Acall ResetNilaiTimer ;Isi nilai timer
Acall StartTimer0 ;Aktifkan timer
Ajmp $

TimerSelesai:
Lcall Posisi_Awal ;LCD ke posisi awal
Mov DPTR,#PesanTimeOut ;Tampilkan pesan time out
Lcall KirimPesan_LCD ;
Clr P3.2
Ajmp $

PesanTimeOut:
DB 'Time Out',0FH
  
```

Pada potongan program 1 adalah merupakan inti dari program timer ini, di mana setiap kali interupsi terjadi, maka nilai waktu yang berada di variabel nilai timer berkurang satu hingga nilai waktu mencapai nol. Pada saat nilai waktu mencapai nol maka program harus melompat ke bagian dari program utama yang menunjukkan bahwa timer telah selesai, yaitu pada label timer selesai.

Hal ini dilakukan dengan memanipulasi isi stack pointer dengan label timer selesai sehingga instruksi RETI akan mengalihkan jalannya program menuju ke label tersebut (perhatikan bagian untuk mengembalikan nilai-nilai register)

Potongan Program 1 Layanan Interupsi Timer

```

;*****
; RUTIN INTERUPSI TIMER 0
; - Terjadi setiap 10 ms
; - Nilai Timer berkurang setiap 10 ms
;*****

Timer0_Interrupt:
    Acall ResetTimer0
    Acall KurangiWaktu
    Setb  ET0
    Reti

;*****
; RUTIN PENGURANGAN WAKTU DENGAN FORMASI 24 HOURS
;*****

KurangiWaktu:
    Mov    R0,#NilaiTimer           ;Mili detik - 1
    Dec   @R0                       ;
    Mov   A,@R0                     ;
    Cjne  A,#-1,TidakReset          ;
    Mov   @R0,#99                   ;Mili detik = -1 -> Mili detik = 99
    Acall DisplayTimer              ;Tampilkan nilai timer
    Inc   R0                         ;Detik - 1
    Dec   @R0                       ;
    Mov   A,@R0                     ;

    Cjne  A,#-1,TidakReset          ;Detik = -1 -> Detik = 59
    Mov   @R0,#59                   ;
    Inc   R0                         ;
    Dec   @R0                       ;Detik - 1
    Mov   A,@R0                     ;

    Cjne  A,#-1,TidakReset          ;Menit = -1 -> Menit = 59
    Mov   @R0,#59                   ;
    Inc   R0                         ;
    Dec   @R0                       ;Menit - 1
    Mov   A,@R0                     ;

    Cjne  A,#-1,TidakReset          ;Jam = -1 -> Time Out
    Clr   TR0                       ;Timer berhenti

;*****

```

```

;BAGIAN UNTUK MENGEMBALIKAN NILAI-NILAI REGISTER KE KONDISI SEMULA
;*****
    Mov    RTemp+5,DPH    ;
    Mov    RTemp+4,DPL    ;
    Mov    RTemp+3,00     ;
    Mov    RTemp+2,A      ;Simpan nilai-nilai register di register
    Mov    RTemp+1,B      ;temporary
    Mov    RTemp,PSW      ;

    Mov    B,StackTimer
    Mov    A,SP
    Clr    C
    Subb   A,B

LoopRecover:
    Pop    00H
    Djnz   ACC,LoopRecover
    Pop    00H
    Pop    00H
    Mov    DPTR,#TimerSelesai;
    Push   DPL             ;Label Timer Selesai diisi ke stack
    Push   DPH             ;
    Mov    PSW,RTemp       ;Ambil kembali nilai-nilai register
    Mov    B,RTemp+1       ;dari register temporary
    Mov    A,RTemp+2       ;
    Mov    00H,RTemp+3     ;
    Mov    DPL,RTemp+4     ;
    Mov    DPH,RTemp+5     ;

TidakReset:
    Ret

```

Program lengkap dari timer ini dapat didownload dari www.delta-electronic.com bagian software dengan nama file ptimer.zip.