

PERCOBAAN

2

JUDUL :

DASAR INPUT/OUTPUT (2)
(PORT PPI DAN PORT 1 SEBAGAI INPUT/OUTPUT)
Menggunakan DT-51 MinSys

TUJUAN :

- Mengamati keluaran data berupa nyala LED setelah proses pemindahan data (akses eksternal) dari sebuah register ke alamat memori yang ditunjuk oleh Data Pointer atau I/O port (Port 1 maupun Port PPI).
- Memanfaatkan port input dan output pada minimum sistem DT-51 untuk transfer data secara parallel.
- Membuat program untuk kontrol nyala LED menggunakan SWITCH on/off (I/O)

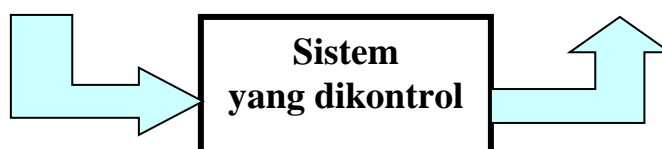
TEORI :

Pada percobaan sebelumnya (percobaan 1) telah dijelaskan fungsi dari masing-masing Port pada DT-51 MinSys yang bisa digunakan sebagai input/output, yaitu Port 1 mikrokontroler 89S51, serta Port A, Port B, dan Port C dari PPI 82C55.

Semua Port diatas dapat diimplementasikan pada system rangkaian kontrol elektronik yaitu sebuah sistem yang mempunyai piranti input dan piranti output. Piranti input berfungsi sebagai piranti penerima data-data dari sistem yang dikontrol, sedang piranti output berfungsi sebagai pemberi aksi kepada sistem yang dikontrol.

Piranti Input/Output dari sistem kontrol merupakan pintu gerbang hubungan ke sistem yang akan dikontrol. Data masukan melalui gerbang peripheral input maupun data keluaran melalui peripheral output adalah berupa level biner.

Sedangkan yang disebut sebagai sistem yang dikontrol diilustrasikan pada gambar 2-1. dari output rangkaian kontrol ke input rangkaian kontrol



Gambar 2-1. Blok diagram sistem yang dikontrol

Terminal input dapat terdiri dari sebuah jalur atau lebih (sekumpulan jalur) yang beroperasi pada logika digital. Demikian pula pada terminal output, dapat terdiri dari sebuah jalur atau lebih (sekumpulan jalur).

Pada percobaan berikut ini (percobaan 2), kita akan menggunakan Port PPI dan Port 1 pada DT-51 MinSys sebagai jalur input dan output. Rangkaian toggle switch dihubungkan pada jalur input sedangkan jalur output dihubungkan dengan LED.

Port PPI Sebagai Input

Sebagai masukan, port A, port B, atau port C dapat digunakan sebagai jalur untuk menerima data, seperti telah dijelaskan pada percobaan sebelumnya (Percobaan 1), bahwa port A dan port B hanya bisa diakses secara *byte* sedangkan port C bisa diakses secara *byte* maupun *bit* dengan cara mengatur format Bit Set/Reset pada control world. Pengertian dari akses per bit adalah mengakses tiap bit secara sendiri – sendiri, tidak secara keseluruhan (8 bit). Sedangkan akses byte adalah mengakses keseluruhan (8 bit) sekaligus.

➤ Port A Sebagai Input

Sebagai input, kita bisa memasukkan data, misalnya melalui Port A. Port A dari PPI 8255 pada DT-51 MinSys dihubungkan dengan piranti input yaitu toggle switch 8-bit (SW), sedangkan output menggunakan Port output DT-51 MinSys yaitu Port 1 dihubungkan dengan piranti output (LED).

Karena menggunakan Port A PPI sebagai input, maka control wordnya bernilai 90h (bit 4 control word bernilai “1”). Dengan pemberian nilai 90h pada control word berarti Port A berfungsi sebagai input, Port B dan Port C berfungsi sebagai output.

Contoh :

- inisialisasi:

```
MOV  DPTR, #2003h
MOV  A, #90h
MOVX @DPTR, A
```

➤ **Kombinasi Port A, B dan Port C Sebagai Input dan Output**

Berikutnya adalah menggunakan ketiga Port PPI sebagai input dan output. Rangkaian toggle switch (SW) dihubungkan dengan Port B DT-51 MinSys sebagai input, sedangkan output Port C DT-51 MinSys dihubungkan dengan rangkaian LED.

Untuk menentukan inisialisasi PPI supaya Port B sebagai input dan Port C sebagai output, maka control word bernilai 82h (bit 1 = “1”, bit 0 dan 3 = “0”).

Contoh :

- inisialisasi:

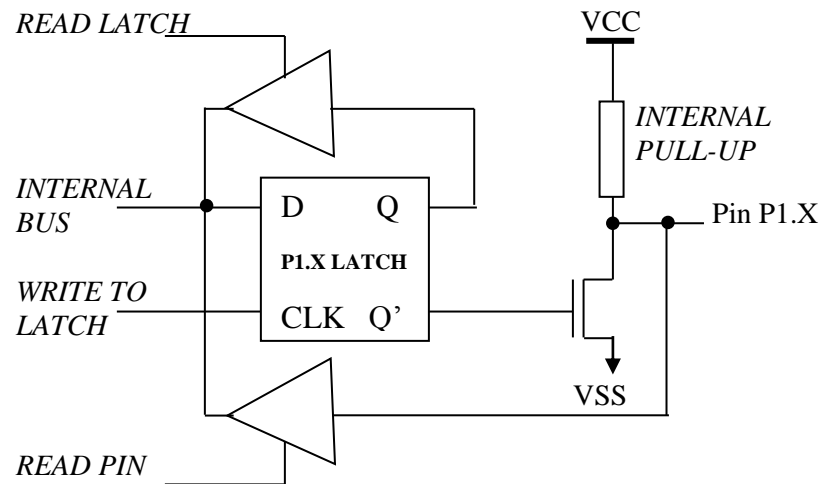
```
MOV  DPTR, #2003h
MOV  A, #82h
MOVX @DPTR, A
```

Port 1 Sebagai Input

Port 1 merupakan Port I/O dwi-arah (bidirectional) yang dilengkapi dengan *internal pull-up* (gambar 2-2). Penyangga keluaran Port 1 mampu memberikan/menyerap arus empat masukan TTL (sekitar 1,6 mA).

Jika “1” dituliskan ke kaki-kaki Port 1, maka masing-masing kaki akan di-pulled high dengan pull-up internal sehingga dapat digunakan sebagai masukan, jika kaki-kaki Port 1 dihubungkan dengan ground (di-pull low), maka masing-masing kaki akan memberikan arus (source) karena di-pulled high secara internal. Port 1 juga menerima lambat rendah (low byte) selama pemrograman dan verifikasi flash.

Sebagai input, Port 1 dapat dimasuki suatu data yang berasal dari piranti input yaitu toggle switch (SW). Sedangkan output bisa menggunakan Port PPI yang terdapat pada DT-51 MinSys.



Gambar 2-2. Rangkaian Internal Port 1

PERALATAN :

- Modul DT-51
- PC beserta software
 - Debugger
 - Downloader
 - Editor
- Toggle switch dan LED sebagai modul I/O

PROSEDUR :

Kombinasi Port A, Port B, Port C dan Port 1 sebagai Input/Output

1. Hardware

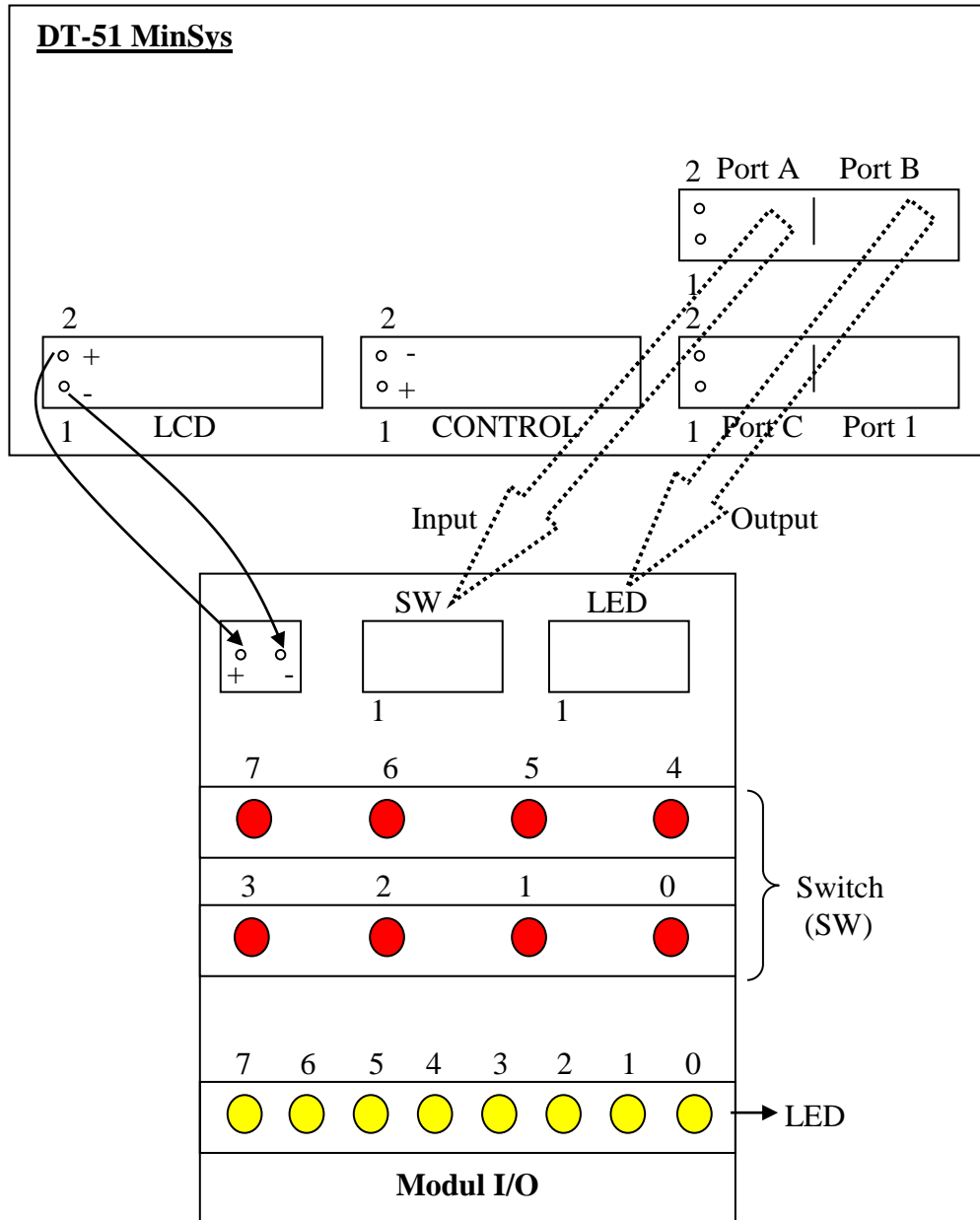
Input

- Hubungkan konektor (kabel pita) Port A, Port B pada DT-51 ke Modul toggle switch jika input yang digunakan adalah Port A atau Port B. Tetapi bila inputnya adalah Port 1 atau Port C maka *hubungkan konektor (kabel pita) Port 1 atau Port C pada DT-51 ke Modul toggle switch.*

Output

- Hubungkan konektor (kabel pita) Port 1, Port C pada DT-51 ke *Modul Display*

LED jika output yang digunakan adalah Port 1 atau Port C. Tetapi bila outputnya adalah Port A atau Port B maka hubungkan konektor (kabel pita) Port A atau Port B pada DT-51 ke Modul Display LED (lihat gambar 2-3).



Gambar 2-3. Koneksi DT-51 MinSys dengan Modul I/O

Selanjutnya

- Hubungkan kabel serial dari konektor DB-9 pada DT-51 ke PC pada Port serial.
- Hubungkan catu daya 9 Volt AC pada modul DT-51.

- Pin Konektor Select pada posisi Pin 1 dan 2 untuk mode *download program*, pada posisi Pin 2 dan 3 untuk mode *stand alone* , setelah download program berhasil.

2. Software

Editor :

- Tulislah program pada editor DOS prompt, Notepad atau lainnya.
- Program / file baru tersebut, simpan dalam direktori(folder) DT51 dengan extension *[namafile].ASM*.
- Exit

Debugger :

- Compile file baru dengan perintah *C:\DT51\ASM51 [nama file].ASM* pada DOS Prompt, dengan terlebih dahulu pindah ke direktori DT51.
- Maka akan terbentuk file-file yang berextension .HEX, .OBJ, dan .LST disamping itu juga muncul pesan, bahwa file yang telah dicompile tersebut terdapat kesalahan atau tidak.
- Bila ada kesalahan, bisa dilihat letak kesalahannya dengan membuka file yang berekstension .LST, melalui *C:\DT51\edit*.
- Untuk memperbaiki kesalahan (error), buka file yang berekstension .ASM. Jangan lupa disimpan kemudian dicompile kembali.
- Exit

Downloader :

- Untuk mendownload program yang telah sukses dicompile, maka gunakan perintah *C:\DT51\DT51L [nama file].HEX*
- Apabila proses download selesai, maka akan muncul pesan Download Succeeded.
- Bila gagal, maka akan muncul beberapa prosedur untuk megecek kembali kesalahan tersebut.

PROGRAM PERCOBAAN :

➤ Akses Port PPI

- Program 2.1 Port A Input, Port B & C Output

```
$MOD51
;=====
; Program Baca data di Switch dan
; tampilkan/tulis di LED
;=====
                ORG 4000H

                MOV     DPTR,#2003H ;inisialisasi
                MOV     A,#90H      ;PA In, PB&PC Out
                MOVX    @DPTR,A
LOOP:           MOV     DPTR,#2000H ;baca Sw input
                MOVX    A,@DPTR
                MOV     DPTR,#2002H ;tulis data
                MOVX    @DPTR,A
                LJMP    LOOP
                END
```

- Program 2.2 Port C Input, Port A & B Output

```
$MOD51
;=====
; Program Baca data di Sw 6 untuk reset dan
; Sw 7 untuk tampilkan data di LED (dengan Delay)
;=====
                ORG 4000H
                MOV     DPTR,#2003H ;inisialisasi PPI
                MOV     A,#10001001B ;PC In, PA&PB Out
                MOVX    @DPTR,A

START:          MOV     DPTR,#2002H ;Baca Sw Input
                MOVX    A,@DPTR
                CJNE    A,#01000000B,CEK_1 ;cek Sw6 on
                SJMP    RESET
CEK_1:          CJNE    A,#10000000B,START ;cek Sw7 on
                MOV     A,#10100101B
                MOV     DPTR,#2001H ;Output data
                MOVX    @DPTR,A
                ACALL   DELAY
                MOV     A,#10000001B
                MOVX    @DPTR,A
                ACALL   DELAY
                LJMP    START
RESET:          MOV     A,#00000000B
                MOV     DPTR,#2001H ;Output data
                MOVX    @DPTR,A
                ACALL   DELAY
```

```

RET
DELAY:  MOV     R1,#3H
DEL1:   MOV     R2,#0FFH
DEL2:   MOV     R3,#0FFH
        DJNZ   R3,$
        DJNZ   R2,DEL2
        DJNZ   R1,DEL1
        RET
        END

```

- Program 2.3 Port B Input, Port A & C Output

```

$MOD51

;=====
; Program kontrol arah gerakan nyala LED
; (rotasi kanan)menggunakan switch On/Off (1/0)
;=====
                                ORG 4000H
                                MOV  DPTR,#2003H    ;inisialisasi
                                MOV  A,#82H         ;PB In, PA&PC Out
                                MOVX @DPTR,A

MAIN:  MOV  DPTR,#2001H    ;baca Sw input
        MOVX A,@DPTR
        CJNE A,#40H,CEK_LAGI
EXIT:  MOV  A,#00000000B
        MOV  DPTR,#2002H    ;Output data
        MOVX @DPTR,A
        ACALL DELAY
        SJMP MAIN

CEK_LAGI: CJNE A,#80H,MAIN
        MOV  A,#80H
LOOP:  MOV  DPTR,#2002H    ;Output
        MOVX @DPTR,A
        ACALL DELAY
        RR  A              ;Rotasi kanan
        SJMP LOOP

DELAY:  MOV  R1,#3H
DEL1:   MOV  R2,#0FFH
DEL2:   MOV  R3,#0FFH
        DJNZ R3,$
        DJNZ R2,DEL2
        DJNZ R1,DEL1
        RET
        END

```


- Program 2.4 Port B Input, Port A & C Output

```

$MOD51
;=====
; Program kontrol kecepatan display LED
; menggunakan switch On/Off (1/0)
;=====
                ORG    4000H
                MOV     DPTR,#2003H    ;inisialisasi
                MOV     A,#82H        ;PB In, PA&PC Out
                MOVX    @DPTR,A

MAIN:           MOV     R0,#01H
                MOV     DPTR,#2001H    ;baca Sw input
                MOVX    A,@DPTR
                CJNE    A,#80H,T_1     ;cek Sw 7
LOOP:           MOV     A,R0
                MOV     DPTR,#2002H    ;Output data
                MOVX    @DPTR,A
                ACALL   DELAY
                INC     R0
                AJMP    LOOP
T_1:            CJNE    A,#40H,T_2     ;cek Sw 6
LOOP1:          MOV     A,R0
                MOV     DPTR,#2002H    ;Output data
                MOVX    @DPTR,A
                ACALL   DELAY1
                INC     R0
                AJMP    LOOP1

T_2:            CJNE    A,#20H,MAIN    ;cek Sw 5
LOOP2:          MOV     A,R0
                MOV     DPTR,#2002H    ;Output data
                MOVX    @DPTR,A
                ACALL   DELAY2
                INC     R0
                AJMP    LOOP2

DELAY:          MOV     DPTR,#DAT
                MOVX    A,@DPTR
                MOV     R1,A
                SJMP    DEL1
DELAY1:         MOV     DPTR,#DAT
                INC     DPTR
                MOVX    A,DPTR
                MOV     R1,A
                SJMP    DEL1
DELAY2:         MOV     DPTR,#DAT
                INC     DPTR
                INC     DPTR
                MOVX    A,@DPTR
                MOV     R1,A
DEL1:           MOV     R2,#5FH
DEL2:           MOV     R3,#0FFH

```

```

                                DJNZ R3,$
                                DJNZ R2,DEL2
                                DJNZ R1,DEL1
                                RET
    DAT:                        DB    0FH,08H,03H
                                END

```

➤ Akses Port 1

- Program 2.5 Port 1 DT-51 sebagai Input, Port A output

```

$MOD51
;=====
; Program menyalakan satu LED
; menggunakan switch On/Off (1/0)
;=====

                                CSEG
                                ORG  4000H
                                JMP  START

                                ORG  4100H
START:                          MOV    SP,#30H
INIT:                            MOV    DPTR,#2003H
                                MOV    A,#80H
                                MOVX   @DPTR,A

LABEL1:                          JB     P1.0,DISPLAY

DISPLAY2:                        MOV    DPTR,#2000H
                                MOV    A,#00H
                                MOVX   @DPTR,A
                                SJMP   LABEL1
DISPLAY:                          MOV    DPTR,#2000H
                                MOV    A,#01H
                                MOVX   @DPTR,A
                                SJMP   LABEL1
                                END

```

TUGAS :

1. Sempurnakan program 2.4 diatas,
 - agar perpindahan saklar 7, 6 dan 5 untuk menjalankan program dengan kecepatan tertentu tanpa harus di download ulang (hanya 1x download).
 - Jika saklar yang On selain saklar 7, 6 dan 5 atau semua saklar Off, program akan reset.

2. Sempurnakan program 2.3 diatas, dengan ketentuan sebagai berikut :
 - ketika saklar 7 saja yang On, rotasi kanan dilaksanakan mulai dari LED bit ke 7 menuju ke LED bit ke 0.
 - Jika saklar yang On saklar 6, maka rotasi dihentikan/reset (semua LED padam).
 - Jika saklar 7 On lagi, maka rotasi kanan dilaksanakan kembali.
 3. Buat program untuk kontrol kecepatan nyala LED (rotasi kanan dan kiri) menggunakan switch input (Sw7-Sw0), jika Sw7 On, maka delay-nya paling besar/lambat dan bila Sw0 On delay-nya paling kecil/cepat.
 4. Buat program dengan input toggle switch di Port C dan output LED di Port B. Semua LED dimulai dari keadaan padam.
 - Jika Port C.5 = On, nyala LED di Port B akan bertambah 1 secara biner.
 - Jika Port C.1 = On, nyala LED di Port B akan berkurang 1 secara biner.
 - Jika Port C.0 = On, semua LED di Port B akan padam atau reset.
- Petunjuk* : gunakan perintah INC dan DEC untuk menambah atau mengurangi dan gunakan prosedur “delay” untuk mencegah agar proses penambahan atau pengurangan bisa berlangsung secara perlahan dan berurutan.