

PETUNJUK PRAKTIKUM EL-3096

SISTEM MIKROPROSESOR



Laboratorium
Dasar
Teknik Elektro

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2010

PETUNJUK PRAKTIKUM SISTEM MIKROPROSESOR

Mervin T. Hutabarat

Waskita Adijarto

Harry Septanto

Laboratorium Dasar Teknik Elektro

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2010

KATA PENGANTAR

Puji dan syukur kami panjatkan pada Tuhan Yang Maha Esa karena rahmat-Nya telah memberikan kami kesempatan untuk menyusun Petunjuk Praktikum Sistem Mikroprosesor yang disesuaikan dengan Kurikulum Program Studi Teknik Elektro tahun 2008 ini. Petunjuk praktikum ini mengalami cukup banyak perubahan dibandingkan dengan petunjuk praktikum sejenis sebelumnya. Hal ini dilakukan sejalan dengan rencana Program Studi Teknik Elektro untuk mengupayakan Akreditasi ABET Internasional. Tuntutan pekerjaan mahasiswa dalam praktikum ini lebih tinggi dengan pengharagaan beban sks yang sesuai. Dalam melaksanakan praktikum ini, mahasiswa dituntut juga untuk menggunakan Buku Catatan Laboratorium dengan pola pencatatan sesuai baku yang berlaku sebagai bukti dalam perselisihan terkait pengajuan paten di negara maju guna melatih mahasiswa menjadi *engineer* yang baik.

Pada kesempatan ini, kami ingin menyampaikan terima kasih yang besar-besarnya pada semua pihak yang telah terlibat dalam penyusunan petunjuk praktikum ini. Secara khusus untuk anggota Tim Penyusun Petunjuk Praktikum Sistem Mikroprosesor, Mas Harry Septanto dan Pak Dr. Waskita Adijarto, yang sudah memberikan tenaga, pikiran dan waktunya untuk perbaikan praktikum dalam Program Studi Teknik Elektro ini. Ucapan terima kasih juga disampaikan untuk dukungan rekan-rekan teknisi di Laboratorium Dasar Teknik Elektro, Pak Suparyanto, Pak Sandra Irawan dan Mbak Nina Lestari.

Akhir kata, semoga semua usaha yang telah dilakukan berkontribusi pada dihasilkannya lulusan Program Studi Teknik Elektro sebagai *engineer* dengan standar internasional.

Bandung, Maret 2010

Tim Penyusun Petunjuk Praktikum Sistem Mikroprosesor

Ketua Tim,

Ir. Mervin T. Hutabarat, M.Sc., Ph.D.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
ATURAN UMUM LABORATORIUM	v
KELENGKAPAN	v
PERSIAPAN	v
PERGANTIAN JADWAL.....	vi
SANKSI	vii
PANDUAN UMUM KESELAMATAN DAN PENGGUNAAN PERALATAN LABORATORIUM	ix
KESELAMATAN	ix
PENGGUNAAN PERALATAN PRAKTIKUM.....	x
SANKSI	xi
MODUL I I/O DASAR	1
I.A PORT A SEBAGAI OUTPUT	3
TUGAS I.A.1 Menjalankan Contoh Program	3
TUGAS I.A.2	3
I.B FASILITAS DELAY PADA AVR GCC	3
TUGAS I.B.1 Menjalankan Contoh Program	3
TUGAS I.B.2	4
TUGAS I.B.3	4
TUGAS I.B.4	5
I.C PORT A SEBAGAI OUTPUT DAN PORT B SEBAGAI INPUT	5
TUGAS I.C.1	5
TUGAS I.C.2	5
MODUL II TIMER, COUNTER, INTERRUPT DAN EXTERNAL INTERRUPT	7
II.A DELAY DENGAN TIMER/ COUNTER0	10
TUGAS II.A.1 Menjalankan Contoh Program	10
TUGAS II.A.2	11
II.B EXTERNAL CLOCK SEBAGAI COUNTER.....	11

TUGAS II.B.1 Menjalankan Contoh Program.....	12
TUGAS II.B.2	12
II.C APLIKASI INTERRUPT DENGAN TIMER/COUNTER.....	13
TUGAS II.C.1 Menjalankan Contoh Program.....	13
TUGAS II.C.2	14
II.D EXTERNAL INTERRUPT.....	14
TUGAS II.D.1 Menjalankan Contoh Program	14
TUGAS II.D.2	15
MODUL III KOMUNIKASI USART DAN EVALUASI MODUL I-III.....	17
III.A MENERIMA DATA DARI PC.....	19
TUGAS III.A.1 Menjalankan Contoh Program	19
TUGAS III.A.2	20
III.B MENGIRIM DATA KE PC.....	20
TUGAS III.B.1 Menjalankan Contoh Program.....	20
TUGAS III.B.2	21
III.C PENGIRIMAN DAN PENERIMAAN DATA DARI DAN KE PC.....	21
TUGAS III.C.1 Menjalankan Contoh Program.....	21
TUGAS III.C.2	22
III.D EVALUASI MODUL I-III.....	22
TUGAS III.D.1	22
MODUL IV SEVEN SEGMENT, KEYPAD DAN LCD.....	25
IV.A SCANNING SEVEN SEGMENT	28
TUGAS IV.A.1 Menjalankan Contoh Program	28
TUGAS IV.A.2.....	29
IV.B SCANNING KEYPAD	30
TUGAS IV.B.1 Menjalankan Contoh Program	30
TUGAS IV.B.2.....	31
TUGAS IV.B.3.....	31
IV.C LCD	32
TUGAS IV.C.1 Menjalankan AVR LCD Lib Demo	32
TUGAS IV.C.2 Menjalankan Contoh Program	32
TUGAS IV.C.3.....	34

MODUL V PERANCANGAN APLIKASI	37
APENDIKS A PETUNJUK PENGGUNAAN WINAVR.....	A-1
APENDIKS B KIT PRAKTIKUM	B-1
SISTEM MINIMUM ATMEGA8535	B-1
DT-51 TRAINER BOARD	B-2
APENDIKS C KODE ASCII	C-1

ATURAN UMUM LABORATORIUM

KELENGKAPAN

Setiap praktikan wajib berpakaian lengkap, mengenakan celana panjang / rok, kemeja dan mengenakan sepatu. Praktikan wajib membawa kelengkapan berikut:

- Modul praktikum
- Buku Catatan Laboratorium (BCL)
- Alat tulis (dan kalkulator, jika diperlukan)
- *Name tag*
- Kartu Praktikum

PERSIAPAN

SEBELUM PRAKTIKUM

- Membaca dan memahami isi modul praktikum
- Mengerjakan hal-hal yang dapat dikerjakan sebelum praktikum dilaksanakan, misalnya mengerjakan soal perhitungan, menyalin *source code*, mengisi Kartu Praktikum dll.
- Mengerjakan Tugas Pendahuluan
- Mengisi daftar hadir
- Mengambil kunci loker dan melengkapi administrasi peminjaman kunci loker (tukarkan dengan kartu identitas: KTM/ SIM/ KTP)

MASUK LABORATORIUM

- PRAKTIKAN dan ASISTEN sudah harus masuk Laboratorium sebelum jam 08.00 (praktikum pagi) atau sebelum jam 13.30 (praktikum siang) waktu laboratorium
- Setelah jam 08.00 (praktikum pagi) atau jam 13.30 (praktikum siang) pintu laboratorium ditutup

SELAMA PRAKTIKUM

- Perhatikan dan kerjakan setiap percobaan dengan waktu sebaik-baiknya, diawali dengan kehadiran praktikan secara tepat waktu
- Kumpulkan Kartu Praktikum pada asisten

- Dokumentasikan pada BCL (lihat Petunjuk Penggunaan BCL) tentang hal-hal penting terkait percobaan yang sedang dilakukan
- Praktikan hanya diperbolehkan menggunakan kit dan peralatan praktikum yang sudah disediakan di meja masing-masing, apabila ada kekurangan atau terjadi kerusakan silahkan hubungi asisten atau teknisi labdasar untuk mengganti dengan yang baru

SETELAH PRAKTIKUM

- Rapikan kit dan peralatan praktikum serta bersihkan meja praktikum seperti keadaan semula
- Pastikan BCL telah ditandatangani oleh asisten
- Kembalikan kunci loker dan melengkapi administrasi pengembalian kunci loker (pastikan kartu identitas (KTM/ SIM/ KTP) diperoleh kembali)
- Kerjakan laporan (lihat Panduan Penyusunan Laporan)
- Kumpulkan laporan pada lemari (sesuai nama asistennya) di ruang Loker, sebelah ruang Lab Dasar. Waktu pengumpulan paling lambat jam 16.30, hari kerja berikutnya setelah praktikum

PERGANTIAN JADWAL

KASUS BIASA

- Pertukaran jadwal hanya dapat dilakukan per kelompok dengan modul yang sama
- Isi Form Pergantian Jadwal (dapat diperoleh di labdasar.ee.itb.ac.id), lalu tunjukkan pada asisten yang bersangkutan, Kordas yang bersangkutan atau TU Lab. Dasar untuk ditandatangani
- Serahkan Form Pergantian Jadwal yang sudah ditandatangani tadi pada asisten saat praktikum

KASUS SAKIT ATAU URUSAN MENDESAK PRIBADI LAINNYA

- Isi Form Pergantian Jadwal dengan melampirkan surat keterangan dokter (bagi yang sakit) atau surat terkait lainnya
- Khusus bagi yang sakit, harus melapor pada kesempatan pertama, yaitu ketika yang bersangkutan masuk kuliah pertama kali setelah sehat dengan menunjukkan surat keterangan sakit pada Kordas
- Form Pergantian Jadwal diserahkan pada TU Lab. Dasar
- Praktikan yang bersangkutan sebelum kesempatan jadwal praktikum selanjutnya harus meminta jadwal praktikum pengganti ke Kordas praktikum terkait

KASUS "KEPENTINGAN MASSAL"

- "Kepentingan massal" terjadi jika ada lebih dari 1/3 rombongan praktikan yang tidak dapat melaksanakan praktikum pada satu hari yang sama karena alasan yang terkait kegiatan akademis
- Isi Form Pergantian Jadwal dan serahkan pada TU Lab. Dasar secepatnya. Jadwal praktikum pengganti satu hari itu akan ditentukan kemudian oleh Kordas praktikum yang bersangkutan

SANKSI

Pengabaian aturan-aturan di atas dapat dikenakan sanksi sebagai berikut:

- Praktikan yang TERLAMBAT tidak diizinkan mengikuti praktikum dan dinyatakan TIDAK LULUS
- Praktikan yang TIDAK MEMBAWA KELENGKAPAN praktikum (Name Tag / BCL / Kartu Praktikum / Modul Praktikum) dikenakan sanksi nilai praktikum percobaan pada hari tersebut sama dengan NOL
- Segala bentuk kerusakan kit atau peralatan praktikum yang dikarenakan kesalahan prosedur atau *Human-Error* akan dikenakan sanksi untuk mengganti seluruh biaya kerusakannya
- Praktikan yang mengambil atau memindahkan kit dan peralatan praktikum dari meja lain serta tidak merapikan kembali meja praktikumnya akan dikenakan sanksi pemotongan nilai
- Pelanggaran akademik berupa *COPY-PASTE* laporan, tugas pendahuluan dan lainnya dinyatakan TIDAK LULUS praktikumnya
- ASISTEN yang BERHALANGAN HADIR harus memberikan kabar minimal sehari sebelum praktikum dan langsung mencari ASISTEN PENGGANTI.
- ASISTEN yang TERLAMBAT akan diberikan sanksi berupa PEMOTONGAN HONOR sesuai dengan keterlambatannya
- BATAS maksimum KETERLAMBATAN ASISTEN adalah 15 MENIT. Jika lebih, asisten tidak diperkenankan masuk laboratorium

PANDUAN UMUM KESELAMATAN DAN PENGGUNAAN PERALATAN LABORATORIUM

KESELAMATAN

Pada prinsipnya, untuk mewujudkan praktikum yang aman diperlukan partisipasi seluruh praktikan dan asisten pada praktikum yang bersangkutan. Dengan demikian, kepatuhan setiap praktikan terhadap uraian panduan pada bagian ini akan sangat membantu mewujudkan praktikum yang aman.

BAHAYA LISTRIK

- Perhatikan dan pelajari tempat-tempat sumber listrik (stop-kontak dan *circuit breaker*) dan cara menyala-matikannya. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, laporkan pada asisten
- Hindari daerah atau benda yang berpotensi menimbulkan bahaya listrik (sengatan listrik/ *strum*) secara tidak disengaja, misalnya kabel jala-jala yang terkelupas dll.
- Tidak melakukan sesuatu yang dapat menimbulkan bahaya listrik pada diri sendiri atau orang lain
- Keringkan bagian tubuh yang basah karena, misalnya, keringat atau sisa air wudhu
- Selalu waspada terhadap bahaya listrik pada setiap aktivitas praktikum

Kecelakaan akibat bahaya listrik yang sering terjadi adalah tersengat arus listrik. Berikut ini adalah hal-hal yang harus diikuti praktikan jika hal itu terjadi:

- Jangan panik
- Matikan semua peralatan elektronik dan sumber listrik di meja masing-masing dan di meja praktikan yang tersengat arus listrik
- Bantu praktikan yang tersengat arus listrik untuk melepaskan diri dari sumber listrik
- Beritahukan dan minta bantuan asisten, praktikan lain dan orang di sekitar anda tentang terjadinya kecelakaan akibat bahaya listrik

BAHAYA API ATAU PANAS BERLEBIH

- Jangan membawa benda-benda mudah terbakar (korek api, gas dll.) ke dalam ruang praktikum bila tidak disyaratkan dalam modul praktikum

- Jangan melakukan sesuatu yang dapat menimbulkan api, percikan api atau panas yang berlebihan
- Jangan melakukan sesuatu yang dapat menimbulkan bahaya api atau panas berlebih pada diri sendiri atau orang lain
- Selalu waspada terhadap bahaya api atau panas berlebih pada setiap aktivitas praktikum

Berikut ini adalah hal-hal yang harus diikuti praktikan jika menghadapi bahaya api atau panas berlebih:

- Jangan panik
- Beritahukan dan minta bantuan asisten, praktikan lain dan orang di sekitar anda tentang terjadinya bahaya api atau panas berlebih
- Matikan semua peralatan elektronik dan sumber listrik di meja masing-masing
- Menjauh dari ruang praktikum

BAHAYA BENDA TAJAM DAN LOGAM

- Dilarang membawa benda tajam (pisau, gunting dan sejenisnya) ke ruang praktikum bila tidak diperlukan untuk pelaksanaan percobaan
- Dilarang memakai perhiasan dari logam misalnya cincin, kalung, gelang dll.
- Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
- Tidak melakukan sesuatu yang dapat menimbulkan luka pada diri sendiri atau orang lain

LAIN-LAIN

- Dilarang membawa makanan dan minuman ke dalam ruang praktikum

PENGUNAAN PERALATAN PRAKTIKUM

Berikut ini adalah panduan yang harus dipatuhi ketika menggunakan alat-alat praktikum:

- Sebelum menggunakan alat-alat praktikum, pahami petunjuk penggunaan alat itu. Petunjuk penggunaan beberapa alat dapat didownload di <http://labdasar.ee.itb.ac.id>
- Perhatikan dan patuhi peringatan (*warning*) yang biasa tertera pada badan alat
- Pahami fungsi atau peruntukan alat-alat praktikum dan gunakanlah alat-alat tersebut hanya untuk aktivitas yang sesuai fungsi atau peruntukannya. Menggunakan alat praktikum di luar fungsi atau peruntukannya dapat menimbulkan kerusakan pada alat tersebut dan bahaya keselamatan praktikan

- Pahami *rating* dan jangkauan kerja alat-alat praktikum dan gunakanlah alat-alat tersebut sesuai *rating* dan jangkauan kerjanya. Menggunakan alat praktikum di luar *rating* dan jangkauan kerjanya dapat menimbulkan kerusakan pada alat tersebut dan bahaya keselamatan praktikan
- Pastikan seluruh peralatan praktikum yang digunakan aman dari benda/ logam tajam, api/ panas berlebih atau lainnya yang dapat mengakibatkan kerusakan pada alat tersebut
- Tidak melakukan aktifitas yang dapat menyebabkan kotor, coretan, goresan atau sejenisnya pada badan alat-alat praktikum yang digunakan

SANKSI

Pengabaian uraian panduan di atas dapat dikenakan sanksi tidak lulus mata kuliah praktikum yang bersangkutan.

MODUL I I/O DASAR

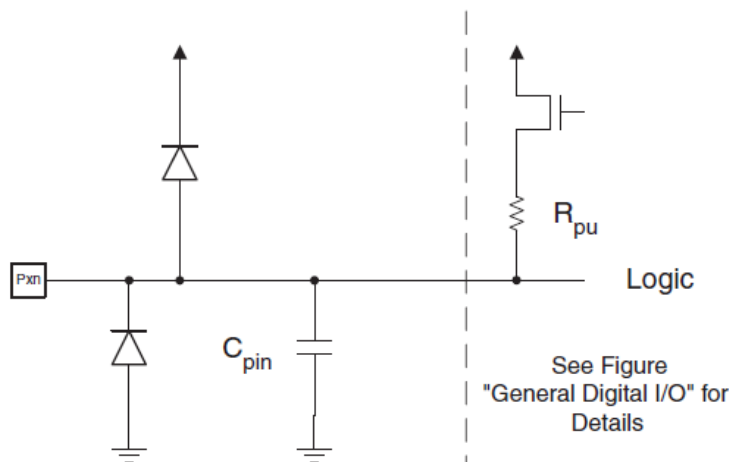
TUJUAN

- Praktikan memahami datasheet ATmega 8535
- Praktikan mampu membuat aplikasi input dan output pada AVR dengan menggunakan bahasa pemrograman C pada WinAVR
- Praktikan mampu mengatur fuse AVR dengan AVRdude

KONFIGURASI I/O ATMEGA 8535

Gambar dan table berikut ini dikutipkan dari datasheet ATmega 8535:

Skematik I/O (Hal. 51)



All registers and bit references in this section are written in general form. A lower case “x” represents the numbering letter for the port, and a lower case “n” represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn. The physical I/O Registers and bit locations are listed in “Register Description for I/O-Ports” on page 66.

Three I/O memory address locations are allocated for each port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. In addition, the Pull-up Disable – PUD bit in SFIOR disables the pull-up function for all pins in all ports when set.

Konfigurasi Pin Port (hal. 53)

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

PUD pada SFIOR (hal. 59)

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	–	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Rekomendasi untuk inisialisasi pada pin input (hal. 56)

If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, floating inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset, Active mode and Idle mode).

The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pull-down. Connecting unused pins directly to V_{CC} or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

Deskripsi register (misalnya Port A) (hal. 66)

Port A Data Register – PORTA

Bit	7	6	5	4	3	2	1	0	
	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Data Direction Register – DDRA

Bit	7	6	5	4	3	2	1	0	
	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port A Input Pins Address – PINA

Bit	7	6	5	4	3	2	1	0	
	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

I.A PORT A SEBAGAI OUTPUT

PERSIAPAN

- Sebelum Power Supply dihubungkan pada kit praktikum, atur terlebih dahulu pada tegangan 9volt. Kemudian, hubungkan pada kit praktikum
- Hubungkan Sismin dengan PC menggunakan kabel DB-9
- Hubungkan **PORT A** pada **PORT OUTPUT**

TUGAS I.A.1 Menjalankan Contoh Program

Tanpa memperhatikan pengaturan fuse, jalankan contoh program di bawah ini, analisis dan buatlah flowchart-nya.

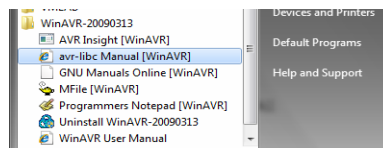
```
#include <avr/io.h>
int main(void)
{
    DDRA = 0xFF; //inisialisasi PORTA sebagai OUTPUT
    PORTA = (1<<PA6) | (1<<PA4) | (1<<PA2) | (1<<PA0); //nilai OUTPUT
    return 0;
}
```

TUGAS I.A.2

Modifikasi program di atas sehingga menghasilkan keluaran led (OUTPUT LED DISLAY) sesuai dengan representasi biner dari nomor kelompok Anda.

I.B FASILITAS DELAY PADA AVR GCC

AVR GCC menyediakan fungsi-fungsi untuk keperluan delay. Penjelasan selengkapnya diuraikan di dalam [avr-libc Manual \[WinAVR\]](#).



TUGAS I.B.1 Menjalankan Contoh Program

Pastikan pengaturan fusenya, lalu jalankan contoh program di bawah ini, analisis dan buatlah flowchart-nya.

```
#include <avr/io.h>
#define F_CPU 1000000UL // Internal Clock
#include <util/delay.h>
```

```

int main(void)
{
    DDRA = 0xFF;
    PORTA = 0xFF;
    while(1)
    {
        PORTA = 0b00110011;
        _delay_ms(500);
        PORTA = 0b11001100;
        _delay_ms(500);
    }
    return 0;
}

```

TUGAS I.B.2

Modifikasi contoh program Tugas 1.B. 1 sehingga menghasilkan keluaran led (OUTPUT LED DISLAY) secara bergantian sesuai dengan representasi biner dari tiga-angka-terakhir NIM anggota kelompok Anda.

TUGAS I.B.3

Jalankan contoh program di bawah ini, analisis dan buatlah flowchart-nya.

```

#include <avr/io.h>
#define F_CPU 1000000UL // 1 MHz
#include <util/delay.h>
int main(void)
{
    unsigned char i = 0xFE;
    DDRA = 0xFF;
    PORTA = i;
    while(1)
    {
        PORTA = i;
        _delay_ms(120);
        i=(i<<1)|(i>>7);
    }
    return 0;
}

```

TUGAS I.B.4

Modifikasi contoh program Tugas I.B.3 sehingga hasilnya tidak menunjukkan nyala-mati lampu yang memutar. Tetapi, setelah sampai ujung kiri akan kembali ke kanan dan sebaliknya.

I.C PORT A SEBAGAI OUTPUT DAN PORT B SEBAGAI INPUT

PERSIAPAN

Pastikan PERSIAPAN pada I.A terpenuhi; kemudian, hubungkan **PORT B** pada **PORT INPUT**

TUGAS I.C.1

Jalankan contoh program di bawah ini, analisis dan buatlah flowchart-nya.

```
#include <avr/io.h>
int main(void)
{
    unsigned char i;
    DDRA = 0xFF; //inisialisasi PORTA sebagai OUTPUT
    DDRB = 0x00; //inisialisasi PORTB sebagai INPUT
    SFIOR = 0<<PUD; //aktivasi resistor pull-up internal
    PORTA = 0xFF;
    while(1)
    {
        i = PINB;
        PORTA = i;
    }
    return 0;
}
```

TUGAS I.C.2

Modifikasi program di atas sehingga kondisi berikut ini terpenuhi:

- Apabila input samadengan representasi biner tiga angka NIM terakhir Anda, maka kondisi led seperti pada Tugas I.B.2;
- Apabila input samadengan representasi biner tiga angka NIM terakhir teman sekelompok Anda, maka kondisi led seperti pada Tugas I.B.4;
- Apabila input tidak-samadengan kedua representasi di atas, maka kondisi led seperti pada Tugas I.A.2 .

MODUL II TIMER, COUNTER, INTERRUPT DAN EXTERNAL INTERRUPT

TUJUAN

- Praktikan memahami datasheet ATmega 8535
- Praktikan mampu membuat aplikasi Timer/ Counter dan Interrupt pada AVR dengan menggunakan bahasa pemrograman C pada WinAVR
- Praktikan mampu membuat aplikasi External Interrupt pada AVR dengan menggunakan bahasa pemrograman C pada WinAVR

KONFIGURASI TIMER/COUNTER DAN INTERRUPT

Gambar dan table berikut ini dikutipkan dari datasheet ATmega 8525:

- Deskripsi register Timer/Counter 8 bit (Timer/Counter 0) (Hal. 71)

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The FOC0 bit is only active when the WGM00 bit specifies a non-PWM mode. However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0 is written when operating in PWM mode. When writing a logical one to the FOC0 bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0 output is changed according to its COM01:0 bits setting. Note that the FOC0 bit is implemented as a strobe. Therefore it is the value present in the COM01:0 bits that determines the effect of the forced compare.

A FOC0 strobe will not generate any interrupt, nor will it clear the timer in CTC mode using OCR0 as TOP.

The FOC0 bit is always read as zero.

- Konfigurasi mode operasi (hal. 83)

Mode	WGM01 (CTC0)	WGM00 (PWM0)	Mode of Operation	TOP	Update of OCR0	TOV0 Flag Set on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Lihat pula hal. 72 untuk memahami definisi MAX, BOTOM dan TOP.

- Pengaturan Perilaku pin OCO (misalnya untuk mode non-PWM) (hal. 84)

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Toggle OC0 on Compare Match
1	0	Clear OC0 on Compare Match
1	1	Set OC0 on Compare Match

- Pengaturan sumber clock (hal. 85)

Table 43. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/counter stopped).
0	0	1	$\text{clk}_{I/O}$ (No prescaling)
0	1	0	$\text{clk}_{I/O}/8$ (From prescaler)
0	1	1	$\text{clk}_{I/O}/64$ (From prescaler)
1	0	0	$\text{clk}_{I/O}/256$ (From prescaler)
1	0	1	$\text{clk}_{I/O}/1024$ (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Pahami Timing Diagram!

- Register-register penting lain (hal. 85)

Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	OCR0[7:0]								OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 1 – OCIE0: Timer/Counter0 Output Compare Match Interrupt Enable

Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable

Bit	7	6	5	4	3	2	1	0	
	OCF2	TOV2	ICF1	OCF1A	OCF1B	TOV1	OCF0	TOV0	TIFR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 1 – OCF0: Output Compare Flag 0

Bit 0 – TOV0: Timer/Counter0 Overflow Flag

KONFIGURASI EXTERNAL INTERRUPT

Register-register untuk keperluan external interrupt diuraikan pada hal. 68-70 di datasheet ATmega8535. Di sini diberikan kutipan-kutipannya.

Bit	7	6	5	4	3	2	1	0	
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

Bit	7	6	5	4	3	2	1	0	
	–	ISC2	–	–	WDRF	BORF	EXTRF	PORF	MCUCSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0		See Bit Description				
Bit	7	6	5	4	3	2	1	0	
	INT1	INT0	INT2	–	–	–	IVSEL	IVCE	GICR
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – INT1: External Interrupt Request 1 Enable

Bit 6 – INT0: External Interrupt Request 0 Enable

Bit 5 – INT2: External Interrupt Request 2 Enable

Bit	7	6	5	4	3	2	1	0	
	INTF1	INTF0	INTF2	–	–	–	–	–	GIFR
Read/Write	R/W	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Bit 7 – INTF1: External Interrupt Flag 1

Bit 6 – INTF0: External Interrupt Flag 0

Bit 5 – INTF2: External Interrupt Flag 2

II.A DELAY DENGAN TIMER/ COUNTER0

PERSIAPAN

- Sebelum Power Supply dihubungkan pada kit praktikum, atur terlebih dahulu pada tegangan 9volt. Kemudian, hubungkan pada kit praktikum
- Hubungkan Sismin dengan PC menggunakan kabel DB9
- Hubungkan **PORT A** (ATMega8535) pada **PORT OUTPUT** (Training Board)

TUGAS II.A.1 Menjalankan Contoh Program

Jalankan contoh program di bawah ini. Analisis dan buatlah flowchart-nya.

```
#include <avr/io.h>
void init_tim(void);
void Delay(void);
int main(void)
```

```

{
    init_tim();
    DDRA=0xFF;
    while (1)
    {
        PORTA = 0x0F;
        Delay();
        PORTA = 0xF0;
        Delay();
    }
    return 0;
}

void init_tim(void)
{
    //mode operasi dan sumber clock
    TCCR0=0b00000101;
    //overflow interrupt disable
    TIMSK=(0<<TOIE0);
}

void Delay(void)
{
    unsigned int n;
    //overflow interrupt enable
    TIMSK=(1<<TOIE0);
    //penentuan nilai awal counter
    TCNT0=0x00;
    //menunggu hingga overflow flag (TOV0) set
    loop_until_bit_is_set(TIFR,0);
    //meng-nol-kan flag TOV0
    TIFR=(1<<TOV0);
    //overflow interrupt disable
    TIMSK=(0<<TOIE0);
}

```

TUGAS II.A.2

Modifikasi program di atas sehingga antara 4bit LSB dengan 4 bit MSB berkedip secara bergantian dengan waktu periode kedip berkisar x detik, dimana $x = (\text{modulus } 9 \text{ dari nomor kelompok}) + 1$.

II.B EXTERNAL CLOCK SEBAGAI COUNTER

PERSIAPAN

Hubungkan prosesor ATmega8535 dengan Training Board (pada **PORT CONTROL**) sehingga pin **T1** pada Prosesor ATmega8535 (lihat datasheet ATmega 8535) terhubung dengan sebuah Interrupt Input (**IS1, IS2, IS3** atau **IS4**) pada Training Board.

TUGAS II.B.1 Menjalankan Contoh Program

Jalankan contoh program di bawah ini, analisis dan buatlah flowchart-nya.

```
#include<avr/io.h>
#define F_CPU 1000000UL // 1 MHz
#include <util/delay.h>

void init_tim(void);
int main(void)
{
    init_tim();
    DDRA=0xFF;
    while (1)
    {
        TCNT1H=0x00;
        TCNT1L=0x00;
        OCR1AH=0;
        OCR1AL=3;
        while(bit_is_clear(TIFR,4))
        {PORTA = 0x0F;}
        TIFR=_BV(4);
        PORTA = 0xF0;
        _delay_ms(350);
    }
    return 0;
}

void init_tim(void)
{
    TCCR1A=0b11000000;
    TCCR1B=0b00000110;
    TIMSK=(1<<OCIE1A);
}
```

TUGAS II.B.2

Modifikasi program diatas sehingga LED menyala semua selama x detik ketika tombol ditekan sejumlah x kali, dimana x = (modulus 9 dari nomor kelompok) + 1. Keterangan: fasilitas delay AVR-gcc tidak digunakan.

II.C APLIKASI INTERRUPT DENGAN TIMER/COUNTER

PERSIAPAN

Pastikan **PORT A** (ATMega8535) terhubung dengan **PORT OUTPUT** (Training Board)

TUGAS II.C.1 Menjalankan Contoh Program

Jalankan contoh program di bawah ini, analisis dan buatlah flowchart-nya.

```
#include <avr/io.h>
#define F_CPU 1000000UL
#include <util/delay.h>
#include <avr/interrupt.h>

void inti_int0(void);
ISR(TIMER0_OVF_vect)
{
    unsigned char i;
    for (i=0;i<2;++i)
    {
        PORTA=0x0F;
        _delay_ms(500);
        PORTA=0xF0;
        _delay_ms(500);
    }
    TIFR=(1<<TOV0);
    TIMSK=0b000000;
}

int main(void)
{
    init_int();
    DDRA=0xFF;
    while (1)
    {
        PORTA = 0xFF;
        _delay_ms(500);
        PORTA = 0b10101010;
        _delay_ms(500);
        TIMSK=0b000001;
    }
    return 0;
}

void init_int(void)
{
    TIMSK=0b000000;
    TCCR0=0b00000101;
```

```

    TCNT0=0x00;
    sei();
}

```

TUGAS II.C.2

Modifikasi program Tugas II.C.2 menggunakan Timer/Counter1

II.D EXTERNAL INTERRUPT

PERSIAPAN

Hubungkan prosesor ATmega8535 dengan Training Board (pada **PORT CONTROL**) sehingga pin **INT0** pada Prosesor ATmega8535 terhubung dengan sebuah Interrupt Input (**IS1, IS2, IS3** atau **IS4**) pada Training Board.

TUGAS II.D.1 Menjalankan Contoh Program

Jalankan contoh program di bawah ini, analisis dan buatlah flowchart-nya.

```

#include <avr/io.h>
#define F_CPU 1000000UL
#include <util/delay.h>
#include <avr/interrupt.h>

void Init_Ext_Int(void);
ISR(INT0_vect)
{
    unsigned char i=0x01,n;
    for(n=0;n<11;++n)
    {
        PORTA = i;
        _delay_ms(500);
        i=(i<<1)|(i>>7);
    }
}

int main(void)
{
    Init_Ext_Int();
    sei();
    DDRA=0xFF;
    while (1)
    {
        PORTA = 0b10101010;
        _delay_ms(500);
        PORTA = 0b01010101;
        _delay_ms(500);
    }
    return 0;
}

void Init_Ext_Int(void)
{
    MCUCR=0x02;
}

```

```
}  
    MCUCSR=0x00 ;  
    GICR=0x40 ;  
    GIFR=0x40 ;  
}
```

TUGAS II.D.2

Buatlah program dengan kondisi:

- Program utama akan menampilkan data "0FFh" bergantian dengan data "00h", masing-masing selama ½ detik.
- Jika terjadi *interrupt* dari INT0, program akan menampilkan data "99h" bergantian dengan data "66h", masing-masing selama ½ detik. Setelah itu kembali ke program utama.
- Jika terjadi *interrupt* dari INT1, program akan menampilkan data "0Ah" bergantian dengan data "55h", masing-masing selama ½ detik.. Setelah itu kembali ke program utama

Keterangan: gunakan fasilitas delay AVR-gcc.

MODUL III KOMUNIKASI USART DAN EVALUASI MODUL I-III

TUJUAN

- Praktikan memahami datasheet ATmega 8535
- Praktikan memahami pengesetan fuse yang berkaitan dengan penggunaan besar dan jenis kristal
- Praktikan mampu membuat aplikasi komunikasi USART

KONFIGURASI TIMER/COUNTER DAN INTERRUPT

Gambar dan table berikut ini dikutipkan dari datasheet ATmega 8525:

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Pengaturan sumber clock (hal. 85)

Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	0	0	0	0	1	1	0	

Pahami Timing Diagram!

- Register-register penting lain (hal. 85)

Table 64. UMSEL Bit Settings

UMSEL	Mode
0	Asynchronous Operation
1	Synchronous Operation

Table 65. UPM Bits Settings

UPM1	UPM0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

Table 66. USBS Bit Settings

USBS	Stop Bit(s)
0	1-bit
1	2-bit

Table 67. UCSZ Bits Settings

UCSZ2	UCSZ1	UCSZ0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

Table 68. UCPOL Bit Settings

UCPOL	Transmitted Data Changed (Output of TxD Pin)	Received Data Sampled (Input on RxD Pin)
0	Rising XCK Edge	Falling XCK Edge
1	Falling XCK Edge	Rising XCK Edge

Bit	15	14	13	12	11	10	9	8	
	URSEL		–	–	–	UBRR[11:8]			UBRRH
	UBRR[7:0]								UBRRL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Baud Rate (bps)	f _{osc} = 8.0000 MHz				f _{osc} = 11.0592 MHz				f _{osc} = 14.7456 MHz			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	207	0.2%	416	-0.1%	287	0.0%	575	0.0%	383	0.0%	767	0.0%
4800	103	0.2%	207	0.2%	143	0.0%	287	0.0%	191	0.0%	383	0.0%
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4k	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2k	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8k	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4k	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6k	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8k	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2k	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4k	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250k	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	0	0.0%	1	0.0%	–	–	2	-7.8%	1	-7.8%	3	-7.8%
1M	–	–	0	0.0%	–	–	–	–	0	-7.8%	1	-7.8%
Max ⁽¹⁾	0.5 Mbps		1 Mbps		691.2 kbps		1.3824 Mbps		921.6 kbps		1.8432 Mbps	

III.A

MENERIMA DATA DARI PC

PERSIAPAN

- Sebelum Power Supply dihubungkan pada kit praktikum, atur terlebih dahulu pada tegangan 9volt. Kemudian, hubungkan pada kit praktikum
- Hubungkan **PORT C** (ATMega8535) dengan **PORT OUTPUT** (Training Board)
- Hubungkan Sismin ATMega 8535 dengan PC menggunakan kabel DB9
- Jalankan HyperTerminal (Cara menyiapkan Hyperterminal dapat dipelajari melalui situs ini: <http://extremeelectronics.co.in/avr-tutorials/rs232-communication-the-level-conversion/>)

TUGAS III.A.1 Menjalankan Contoh Program

Jalankan contoh program di bawah ini. Analisis dan buatlah flowchart-nya.

Keterangan:

- Program berikut ini dijalankan dengan kondisi ATmega8535 menggunakan sumber clock Kristal 8Mhz.
- Amati OUTPUT LED setiap tombol keyboard PC ditekan

```
#include<avr/io.h>

int main (void)
{
    UCSRA=0x00;
    UCSRB=0x10;
    UCSRC=0x86;
    UBRRL=0x33;
    while (1)
    {
        loop_until_bit_is_set(UCSRA,7);
        PORTC=UDR;
        UCSRA|=_BV(7);
    }
    return 0;
}
```

TUGAS III.A.2

Modifikasi contoh program Tugas III.A.1. sehingga kondisinya demikian:

- Baudrate = 2400bps
- Mengaktifkan fasilitas double USART transmission speed

III.B MENGIRIM DATA KE PC

TUGAS III.B.1 Menjalankan Contoh Program

Jalankan contoh program di bawah ini, analisis dan buatlah flowchart-nya.

Keterangan:

- Program berikut ini dijalankan dengan kondisi ATmega8535 menggunakan sumber clock Kristal 8Mhz.

- Amati tampilan pada HyperTerminal

```
#include<avr/io.h>
int main (void)
{
    UCSRA=0x00;
    UCSRB=0x08;
    UCSRC=0x86;
    UBRRL=0x33;
    char kal[]=" Praktikum Sistem Mikroprosesor ";
    unsigned char i=0;
    for (i=0;kal[i]!='\0';++i)
    {
        loop_until_bit_is_set(UCSRA,5);
        UDR=kal[i];
    }
    return 0;
}
```

TUGAS III.B.2

Modifikasi program diatas sehingga LED menyala semua selama x detik ketika tombol ditekan sejumlah x kali, dimana $x = (\text{modulus } 9 \text{ dari nomor kelompok}) + 1$. Keterangan: fasilitas delay AVR-gcc tidak digunakan.

III.C

PENGIRIMAN DAN PENERIMAAN DATA DARI DAN KE PC

PERSIAPAN

- Hubungkan **PORT C** (ATMega8535) dengan **PORT OUTPUT** (Training Board)
- Hubungkan Sismin ATMega 8535 dengan PC menggunakan kabel DB9
- Jalankan HyperTerminal

TUGAS III.C.1 Menjalankan Contoh Program

Jalankan contoh program di bawah ini, analisis dan buatlah flowchart-nya.

Keterangan:

- Program berikut ini dijalankan dengan kondisi ATMega8535 menggunakan sumber clock Kristal 8Mhz.
- Amati tampilan pada HyperTerminal

```
#include<avr/io.h>
```

```

#include<avr/interrupt.h>
#define F_CPU 8000000UL // 8 MHz
#include <util/delay.h>

ISR(USART_RX_vect)
{
    unsigned char kar;
    kar = UDR;
    PORTC = kar;
    loop_until_bit_is_set(UCSRA,5);
    UDR = kar;
}

int main (void)
{
    UCSRA=0x00;
    UCSRB=0x98;
    UCSRC=0x86;
    UBRRL=0x33;
    sei();

    while(1)
    {
        if bit_is_set(UCSRA,3)
        {
            loop_until_bit_is_set(UCSRA,5);
        }
    }
    return 0;
}

```

TUGAS III.C.2

Buatlah aplikasi seperti contoh program di atas, namun ATmega8535 bekerja dengan menggunakan sumber clock internal 1Mhz

III.D EVALUASI MODUL I-III

TUGAS III.D.1

Buatlah sebuah program yang berfungsi untuk memeriksa *password* yang diberikan melalui komunikasi serial dan *toggle switch* (SWITCH INPUT) dengan spesifikasi sebagai berikut:

- Ada dua buah kata kunci yang akan diberikan kepada sistem;
- Pada saat *idle*, yaitu Mode Idle, OUTPUT LED DISPLAY menampilkan nyala sebuah LED seperti pada Tugas I.B.4
- Jika sebuah tombol (INTERRUPT INPUT) ditekan, maka sistem akan berpindah pada Mode Kata Kunci 1, dimana sistem siap menerima kata kunci 1;

- Pada Mode Kata Kunci 1, sistem hanya menyediakan waktu 5 detik untuk menerima input kata kunci yang direpresentasikan oleh SWITCH INPUT. Sementara itu, OUTPUT LED DISPLAY akan merepresentasikan kenaikan waktu per-detik (dimulai dari detik ke-nol) dengan nyala led yang terus bertambah (dari hanya nyala led Bit0, lalu Bit0 dan Bit1 hingga led Bit0 sampaidengan Bit4 menyala). Setelah 5 detik berlalu, program mengecek apakah kata kunci yang direpresentasikan dengan SWITCH INPUT apakah benar atau salah. Jika salah, maka sistem kembali ke Mode Idle.
- Jika kata kunci 1 benar, semua led pada “OUTPUT LED DISPLAY” akan menyala sejenak (500ms) kemudian mati;
- Setelah itu sistem akan berpindah pada Mode Counter;
- Pada Mode Counter, ada sebuah tombol yang digunakan sebagai penghitung. OUTPUT LED DISPLAY menampilkan representasi biner dari jumlah penekanan tombol (COUNTER INPUT). Jika tombol itu telah ditekan sebanyak 5 kali, maka program akan berpindah ke Mode Kata Kunci 2.
- Pada Mode Kata Kunci 2, sistem akan menerima input kata kunci (sebuah karakter) dari komunikasi serial dalam jangka waktu 5 detik. Sementara itu, OUTPUT LED DISPLAY akan merepresentasikan kenaikan waktu per-detik (dimulai dari detik ke-nol) dengan nyala led yang terus bertambah (dari hanya nyala led Bit0, lalu Bit0 dan Bit1 hingga led Bit0 sampaidengan Bit4 menyala). Program mengecek apakah kata kunci yang dimasukkan melalui HyperTerminal tadi adalah benar atau salah? Jika salah, akan kembali ke Mode Idle;
- Jika kata kunci 2 benar, program akan membuat seluruh led pada OUTPUT LED DISPLAY nyala dan mati, bergantian tiap detik, sebanyak 2 kali. Setelah itu kembali ke Mode Idle

Catatan:

- Port, mode timer dan kombinasi kata kunci yang digunakan bebas.

MODUL IV SEVEN SEGMENT, KEYPAD DAN LCD

TUJUAN

- Praktikan memahami datasheet ATmega 8535
- Praktikan mampu membuat aplikasi Seven Segment, Keypad dan LCD

SCANNING SEVEN SEGMENT

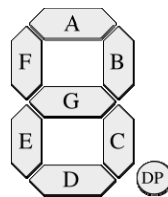
Salah satu teknik untuk menjalankan perangkat Seven Segment adalah teknik Scanning. Dengan teknik scanning ini, sejumlah seven segmen dapat dijalankan secara bergantian (dengan frekuensi tertentu) melalui jumlah port yang minimum.

Pada praktikum ini akan dipelajari teknik scanning untuk menjalankan dua buah seven segment pada DT-51 Tutorial Board. Aturan untuk menjalankan kedua seven segmen tersebut adalah sebagai berikut:

- Untuk menjalankan *seven segment* 1 (berlabel DIGIT1) maka pin DO1 (dari Port "I/P S KEY") harus diberi logika "0";
- Sedangkan untuk dapat menjalankan *seven segment* 2 (DIGIT1) maka pin DO2 (dari port "I/P S KEY") harus diberi logika "0".

Dengan demikian, untuk menyalakan masing-masing seven segemen tersebut, diatur logika "0" untuk pin D01 dan D02 secara bergantian.

Selain dua aturan di atas, untuk menampilkan angka (0 – 9) dan titik (.) dengan seven segment pada DT-51 Tutorial Board tersebut yaitu dengan mengatur logika pin dari port "Data 7S". Rangkaian seven segment tersebut berisfat *active high* (logika "1" = nyala, logika "0" = padam), dengan urutan dari MSB ke LSB adalah DP (titik), G, F, E, D, C, B, A.



SCANNING KEYPAD

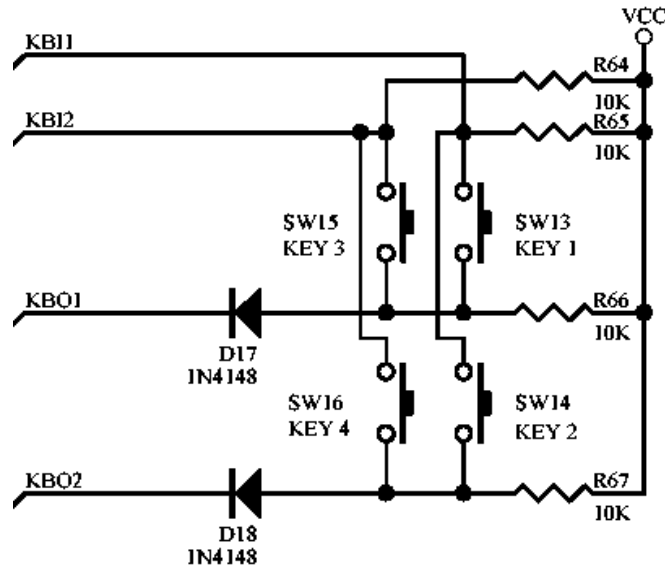
Pada prinsipnya, teknik scanning pada keypad mirip dengan teknik scanning pada seven segmen. Namun, pada scanning keypad, selain dilakukan pengaturan logika pin secara bergantian, juga dilakukan secara bergantian pemeriksaan/ pembacaan pin oleh mikrokontroler untuk mengetahui tombol keypad mana yang sedang ditekan.

DT-51 Tutorial Board memiliki keypad 2x2 dengan kondisi sebagai berikut:

- Pin penerima output dari mikrokontroler adalah pin KBO1 (dari port "I/P S KEY") dan KBO2 (dari port "I/P S KEY") yang bersifat *active low*.
- Sedangkan pin yang memberikan input pada mikrokontroler adalah KBI1 (dari port "I/P S KEY") dan KBI2 (dari port "I/P S KEY") yang bersifat *active low*.

Berikut ini adalah tahap-tahap untuk menjalankan keypad 2x2:

- Pertama, berikan logika "0" pada pin KBO1 (sementara pin KBO2 berlogika "1"). Lalu, periksa logika dari data input KBI1. Jika logika pin KBI1 berubah dari "1" ke "0" berarti KEY1 telah ditekan (perhatikan rangkain listriknya).
 - Selanjutnya KBI2 diperiksa; jika logikanya berubah dari "1" ke "0" berarti KEY3 telah ditekan.
- Kedua, berikan logika "0" pada KBO2 (sementara pin KBO1 berlogika "1"). Kemudian tiap-tiap data *input*-nya diperiksa, seperti pemeriksaan pada tahap pertama.



LCD

LCD yang digunakan dalam praktikum ini adalah LCD yang berbasis pada HD44780 LCD controller. Sebelum praktikum, praktikan diharuskan untuk membaca dan memahami proses untuk menjalankan LCD yang diuraikan pada datasheet HD44780 tersebut. Datasheet HD44780 disediakan di <http://labdasar.ee.itb.ac.id>

Pada praktikum ini, percobaan tentang LCD akan dilakukan dengan menggunakan library beserta demo project-nya yang dapat didownload free di sini: http://winavr.scienceprog.com/download/AVR_LCD_Lib_Demo.zip

Hal pertama yang harus diperhatikan dalam menggunakan library tersebut untuk percobaan pada praktikum ini adalah bahwa pada praktikum ini LCD dijalankan dengan menggunakan Port C. Sementara pada library tersebut, LCD dijalankan dengan Port D. Pengubahan dari Port D ke Port C dilakukan dengan mengedit 4 baris perintah pada file lcd_lib.h menjadi sebagai berikut:

```
#define LDP PORTC
#define LCP PORTC
#define LDDR DDRC
#define LCDR DDRC
```

IV.A SCANNING SEVEN SEGMENT

PERSIAPAN

- Sebelum Power Supply dihubungkan pada kit praktikum, atur terlebih dahulu pada tegangan 9volt. Kemudian, hubungkan pada kit praktikum
- Hubungkan Sismin ATmega 8535 dengan PC menggunakan kabel DB9
- Hubungkan **PORT A** DT-51 MinSys dengan **“DATA 7S”** DT-51 Trainer Board.
- Hubungkan **PORT D** DT-51 MinSys dengan **“I/P S KEY”** DT-51 Trainer Board.

TUGAS IV.A.1 Menjalankan Contoh Program

```
#include <avr/io.h>
#define F_CPU 8000000UL
#include <util/delay.h>

char angka;
char kodehex;
unsigned int SSegmen(char angka);

int main (void)
{
    DDRA = 0xFF;
    DDRD = 0xFF;

    while (1)
    {
        PORTD =(0<<PC7) | (1<<PC6);
        PORTA =SSegmen('7') | (1<<7);
        _delay_ms(1);
        PORTD =(1<<PC7) | (0<<PC6);
        PORTA =SSegmen('2');
        _delay_ms(1);
    }
    return 0;
}
```

```
unsigned int SSegmen(char angka)
{
    switch (angka)
    {
        case '0':
            kodehex = 0x3f;
            break;
        case '1':
            kodehex = 0x06;
            break;
        case '2':
            kodehex = 0x5b;
            break;
        case '3':
            kodehex = 0x4f;
            break;
        case '4':
            kodehex = 0x66;
            break;
        case '5':
            kodehex = 0x6d;
            break;
        case '6':
            kodehex = 0x7d;
            break;
        case '7':
            kodehex = 0x07;
            break;
        case '8':
            kodehex = 0x7f;
            break;
        case '9':
            kodehex = 0x6f;
            break;
    }
    return kodehex;
}
```

TUGAS IV.A.2

Modifikasi Contoh Program di atas sehingga berjalan dengan kondisi berikut:

- Pertama, *seven segment* akan menampilkan "5.0"
- 1 detik kemudian, *seven segment* akan menampilkan "4.0"
- 1 detik kemudian, *seven segment* akan menampilkan "3.0."
- 1 detik kemudian, *seven segment* akan menampilkan "2.0."
- 1 detik kemudian, *seven segment* akan menampilkan "1.0"
- 1 detik kemudian, program kembali menampilkan "0.0" yang berkedip-kedip 3x dengan selang waktu setengah detik. Kemudian padam.

IV.B SCANNING KEYPAD

PERSIAPAN

- Hubungkan **PORT A** pada "**PORT OUTPUT**" DT-51 Trainer Board.
- Hubungkan **PORT D** pada "**I/P S KEY**" DT-51 Trainer Board.

TUGAS IV.B.1 Menjalankan Contoh Program

Jalankan contoh program di bawah ini, analisis dan buatlah flowchart-nya.

```
#include <avr/io.h>
#define F_CPU 8000000UL
#include <util/delay.h>

int main (void)
{
    DDRA=0xFF;
    PORTA=0x00;
    DDRD=(1<<PD4) | (1<<PD5);
    SFIOR=(0<<PUD);
    while(1)
    {
        PORTD=(1<<PD4) | (0<<PD5);
        if(bit_is_clear(PIND,0))
            PORTA=0b00000001;
```

```

        _delay_ms(50);
        if(bit_is_clear(PIND,1))
            PORTA=0b00001111;
        _delay_ms(50);

        PORTD=(0<<PD4)|(1<<PD5);
        if(bit_is_clear(PIND,0))
            PORTA=0b00000011;
        _delay_ms(50);
        if(bit_is_clear(PIND,1))
            PORTA=0b00000111;
        _delay_ms(50);
    }
    return 0;
}

```

TUGAS IV.B.2

Modifikasi Contoh Program diatas sehingga memenuhi kondisi berikut ini:

- Ketika tombol keypad belum ditekan samasekali, seven segment akan menampilkan angka "00"
- Jika SW1 ditekan maka seven segment akan menampilkan angka "11"
- Jika SW2 ditekan maka seven segment akan menampilkan angka "22"
- Jika SW3 ditekan maka seven segment akan menampilkan angka "33"
- Jika SW4 ditekan maka seven segment akan menampilkan angka "44".

TUGAS IV.B.3

Modifikasi Contoh Program di atas sehingga keypad 4x4 pada Kit Praktikum dan seven segment bekerja sebagai berikut:

- Ketika tombol keypad belum ditekan samasekali, seven segment tidak menampilkan apapun.
- Penekanan tombol dengan label angka pada keypad direpresentasikan dengan tampilan pada seven segment. Contoh, penekanan tombol keypad dengan label "3" direpresentasikan dengan tampilan "3.0" pada seven segment.
- Jika tombol berlabel "A" ditekan maka seven segment tidak menampilkan apapun.

IV.C

LCD

PERSIAPAN

- Hubungkan **PORT LCD** pada Sismin dengan **PORT LCD** pada LCD
- Hubungkan **PORT D** pada “**I/P S KEY**” DT-51 Trainer Board.

TUGAS IV.C.1 Menjalankan AVR LCD Lib Demo

Jalankan AVR LCD Lib Demo dengan membuka terlebih dahulu `avrlcd.pnproj` dan kemudian adaptasi Makefile.

Berikan penjelasan singkat tentang contoh program tersebut, termasuk alur program pada file `main.c`

TUGAS IV.C.2 Menjalankan Contoh Program

Contoh program di bawah ini merupakan program pada file `main.c` yang dimodifikasi.

```
#include <avr/io.h>
#include <avr/pgmspace.h>
#include <util/delay.h>
#include "lcd_lib.h"

//Strings stored in AVR Flash memory
const uint8_t LCDtomboll[] PROGMEM="***Tekan Tombol Scanning Keypad***\0";

//delay 1s
void delay1s(void)
{
    uint8_t i;
    for(i=0;i<100;i++)
    {
        _delay_ms(10);
    }
}

int main(void)
{
```



```

LCDinit();//init LCD bit, dual line, cursor right
LCDclr();//clears LCD

DDRA=0xFF;
PORTA=0x00;
DDRD=(1<<PD4)|(1<<PD5)|(1<<PC7)|(1<<PC6);
SFIOR=(0<<PUD);

delay1s();
LCDGotoXY(0, 1);
delay1s();
while(1)//loop demos
{
    CopyStringtoLCD(LCDtomboll, 0, 0);
    LCDshiftRight(1);

    PORTD =(1<<PD4)|(0<<PD5)|(1<<PC7)|(1<<PC6);
    if(bit_is_clear(PIND,0))
    {
        LCDclr();
        LCDGotoXY(8, 1);
        LCDsendChar('&');
        delay1s();
        LCDclr();
    }

    _delay_ms(50);
    if(bit_is_clear(PIND,1))
    {
        LCDclr();
        LCDGotoXY(8, 1);
        LCDsendChar('a');
        delay1s();
        LCDclr();
    }
}

```

```

    }
    _delay_ms(50);

    PORTD =(0<<PD4)|(1<<PD5)|(1<<PC7)|(1<<PC6);
    if(bit_is_clear(PIND,0))
    {
        LCDclr();
        LCDGotoXY(8, 1);
        LCDsendChar('5');
        delay1s();
        LCDclr();
    }
    _delay_ms(50);
    if(bit_is_clear(PIND,1))
    {
        LCDclr();
        LCDGotoXY(8, 1);
        LCDsendChar('7');
        delay1s();
        LCDclr();
    }
    _delay_ms(50);

}

return 0;
}

```

TUGAS IV.C.3

Modifikasi contoh program di atas sehingga keypad 4x4 dan LCD bekerja sebagai berikut:

- Sebelum ada tombol yang ditekan, LCD menampilkan tulisan “Percobaan LCD”
- Setiap penekanan tombol berlabel angka, LCD merepresentasikan angka tersebut.

- Jika secara berurutan ditekan, misalnya, tombol dengan label “1”, “3” dan “2” maka LCD menampilkan “132”.
- Setelah penekanan tombol keypad sebanyak 4x dan kelipatannya, layar LCD bergeser 2 karakter ke kiri.

MODUL V PERANCANGAN APLIKASI

Tugas Modul V adalah melakukan perancangan suatu aplikasi sederhana yang memanfaatkan fasilitas mikrokontroler AVR AT8535. Percobaan ini dilakukan per-kelompok secara mandiri dalam waktu 2 minggu.

APENDIKS A PETUNJUK PENGGUNAAN WINAVR

WinAVR (winavr.sourceforge.net) adalah seperangkat software opensource, termasuk di dalamnya adalah Programmers Notepad (<http://sourceforge.net/projects/pnotepad/>) dan AVRdude (<http://savannah.nongnu.org/projects/avrdude/>), untuk membuat, meng-compile dan mendownload program Atmel AVR dalam bahasa C.

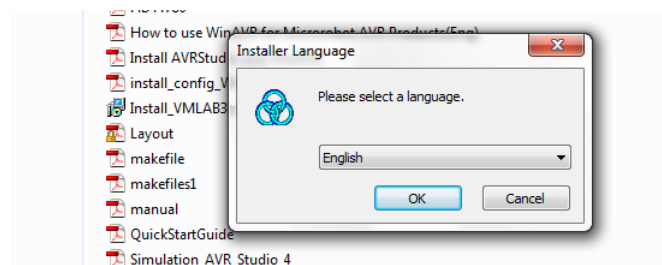
Pada situs winavr.sourceforge.net, selain disediakan installer WinAVR yang dapat didownload secara gratis, juga disediakan manual WinAVR. Namun demikian, berikut ini akan diulas secara singkat, mulai instalasi WinAVR, membuat program, hingga bagaimana memprogram ke dalam prosesor AVR. Langkah-langkah praktis disajikan secara berkesinambungan dari 1 hingga 39.

LANGKAH-LANGKAH MENG-INSTALL WINAVR PADA PC

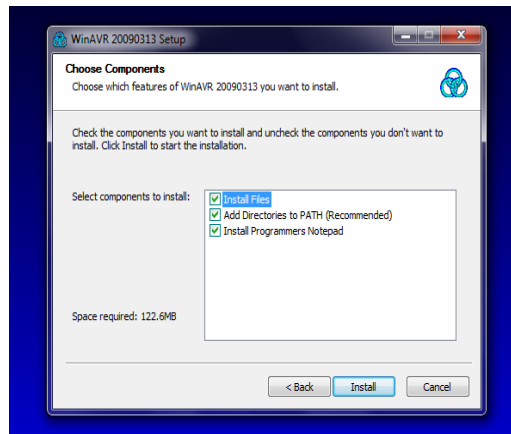
1. Jalankan program installer WinAVR; pada contoh ini digunakan WinAVR versi 20090313 (tampak pada nama file-nya)

twonavr	1/10/2010 12:59 AM	WinRAR ZIP archive	91 KB
VMLABusercomp	1/11/2010 5:03 PM	Adobe Acrobat D...	365 KB
WinAVR-20090313-install	1/7/2010 2:26 PM	Application	22,094 KB

2. Installer WinAVR kemudian akan menunjukkan dialog berikut:



3. Lalu, tekan OK jika diinginkan bahasa Inggris sebagai bahasa pengantarnya.
4. Proses instalasi selanjutnya adalah berisi Lisence Agreement yang harus kita ikuti, kemudian pemilihan lokasi instalasi WinAVR
5. Selanjutnya, pastikan seluruh (tiga) komponen dipilih untuk diinstal. Pemilihan ini ditandai dengan tanda centang (✓)



6. Ikuti hingga selesai.

LANGKAH-LANGKAH MENGINSTALL DRIVER GIVEIO

Driver GiveIO harus diinstall terlebih dahulu agar pemrograman dengan AVRdude dapat dilakukan.

7. Jalankan Command Prompt dengan mengetikkan “cmd” pada Kotak Run



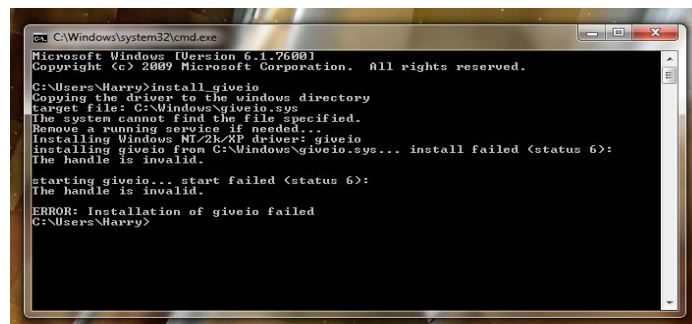
8. Lalu, tekan Enter (untuk Win XP) atau Ctrl+Shift+Enter (untuk Win Vista atau Win 7), sehingga Command Prompt yang muncul dengan mode Administrator



9. Kemudian, ketik “install_giveio” dan tekan Enter

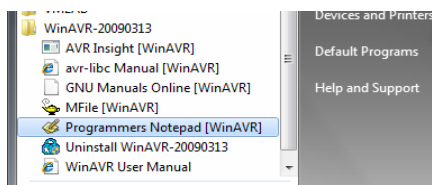


10. Apabila pada Command Prompt ditampilkan sebagaimana gambar di atas, maka instalasi driver GiveIO telah berhasil. Namun, apabila tidak seperti itu, kemungkinan besar karena Command Prompt tidak berjalan dalam mode Administrator. Berikut contoh instalasi GiveIO yang gagal:



LANGKAH-LANGKAH MEMBUAT PROGRAM DENGAN PROGRAMMERS NOTEPAD [WINAVR]

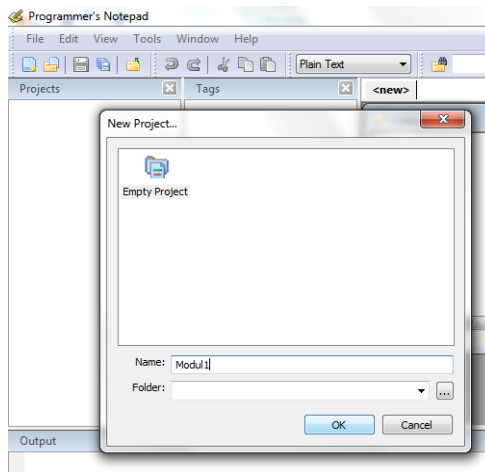
11. Jalankan Programmers Notepad [WinAVR]



12. Buatlah New Project: File → New → Project

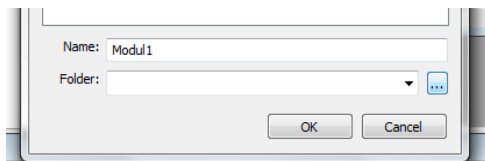


13. Beri nama New Project tersebut, misalnya “Modul1”

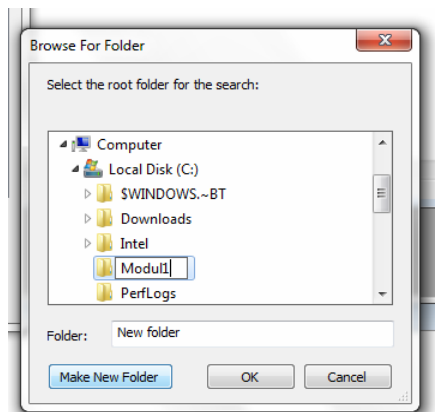


14. Pilih Folder dimana New Project akan disimpan:

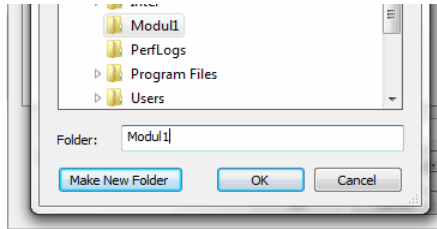
a. Klik Kotak “...”



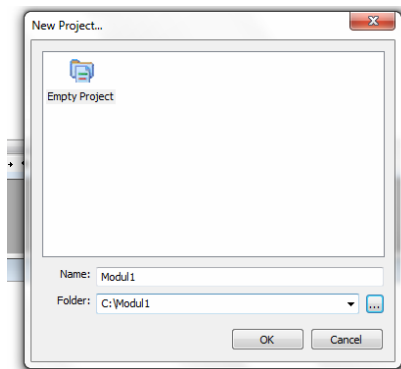
b. Misalkan folder baru ditempatkan pada Drive C, kemudian, klik “Make New Folder” lalu beri nama folder “Modul1”



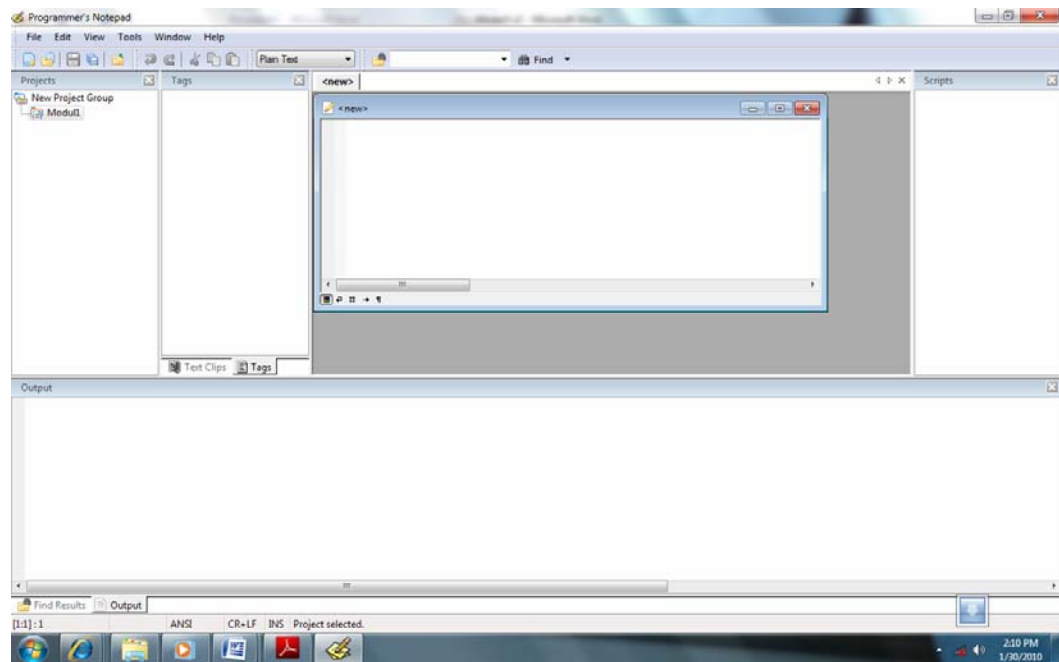
c. Kemudian pilih folder baru tersebut sehingga “Folder: New folder” berubah menjadi “Folder: Modul1”



d. Akhiri dengan klik OK

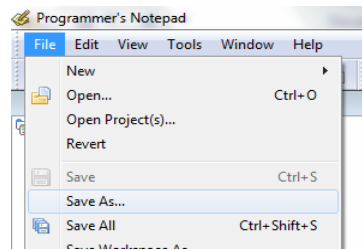


e. Klik OK sekali lagi maka project baru dengan nama “Modul1” telah siap.

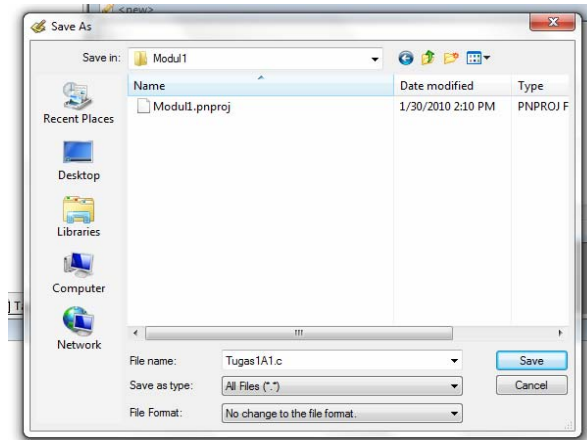


15. Selanjutnya adalah menyiapkan file editor dengan nama “Tugas1A1.c”

a. Klik: File → Save As...



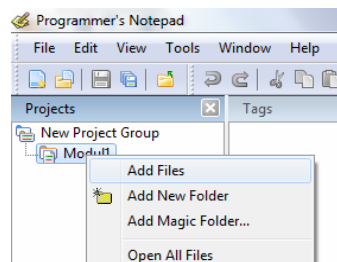
b. Beri nama file “Tugas1A1.c”



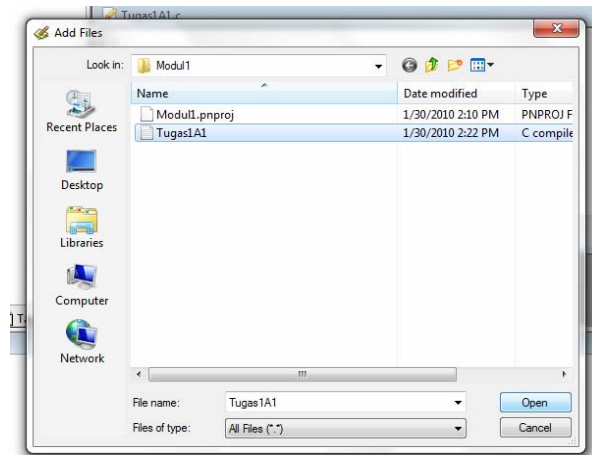
c. Pastikan file “Tugas1A1.c” berada pada folder baru yang tadi dibuat, yaitu folder “C:\Modul1”. Kemudian, akhiri dengan klik Save

16. Ikutsertakan file “Tugas1A1.c” pada project “Modul1”

a. Klik kanan (Porject)(New Project Group)Modul1→pilih Add Files



b. Pilih file “Tugas1A1.c” tadi



c. Akhiri dengan klik Open



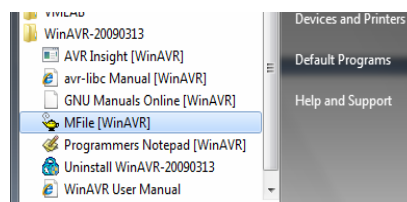
17. Ketikkan baris-baris program Tugas1A1 sebagaimana gambar berikut:



LANGKAH-LANGKAH MENYIAPKAN MAKEFILE

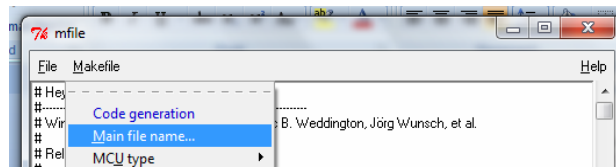
Program yang telah diketik tadi tidak dapat di-compile jika belum disiapkan file “Makefile” yang berada pada folder yang sama dengan file “Tugas1A1.c”, yaitu pada folder “C:\Modul1”.

18. Jalankan MFile [WinAVR]

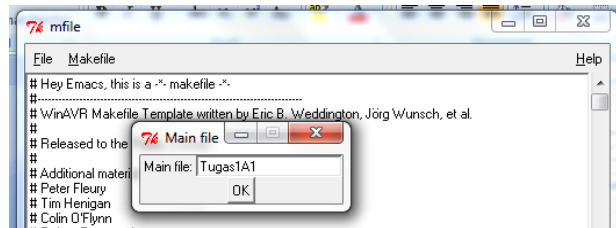


19. Atur “TARGET = Tugas1A1” dengan langkah:

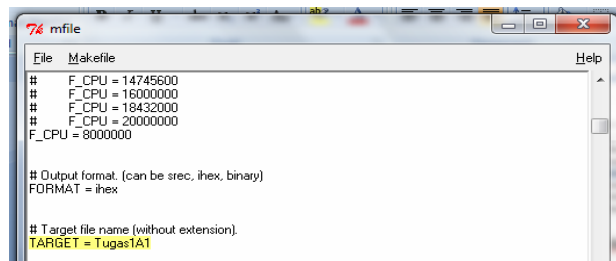
a. Makefile → klik Main file name...



- b. Pada kotak Main file → Ketikan nama “Tugas1A1”, sebagaimana gambar berikut:

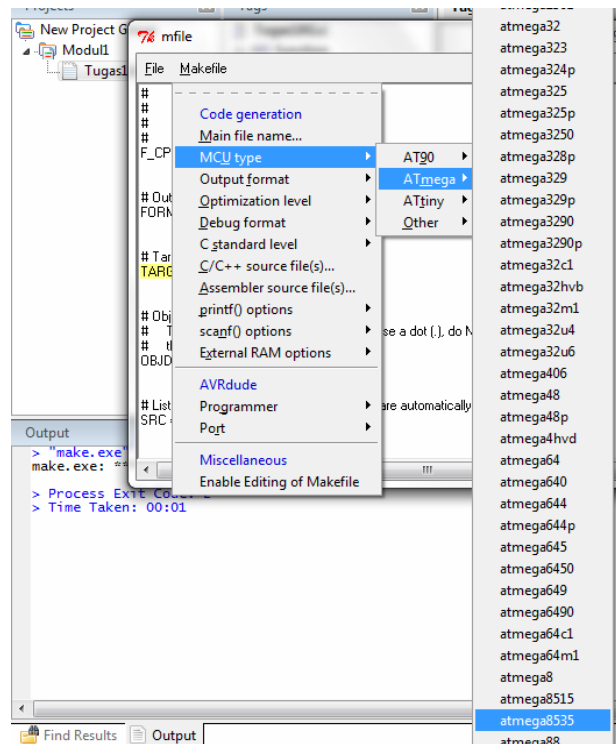


- c. Akhiri dengan klik OK

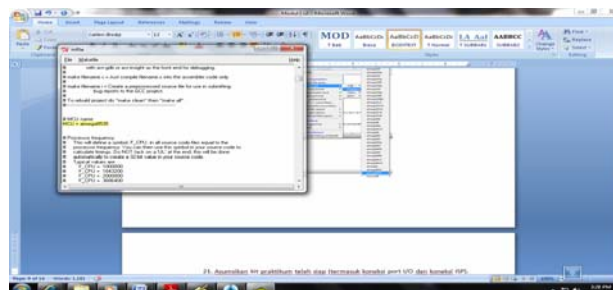


20. Pilih jenis prosesor: Makefile→MCU type →ATmega

- a. Pilih atmega8535

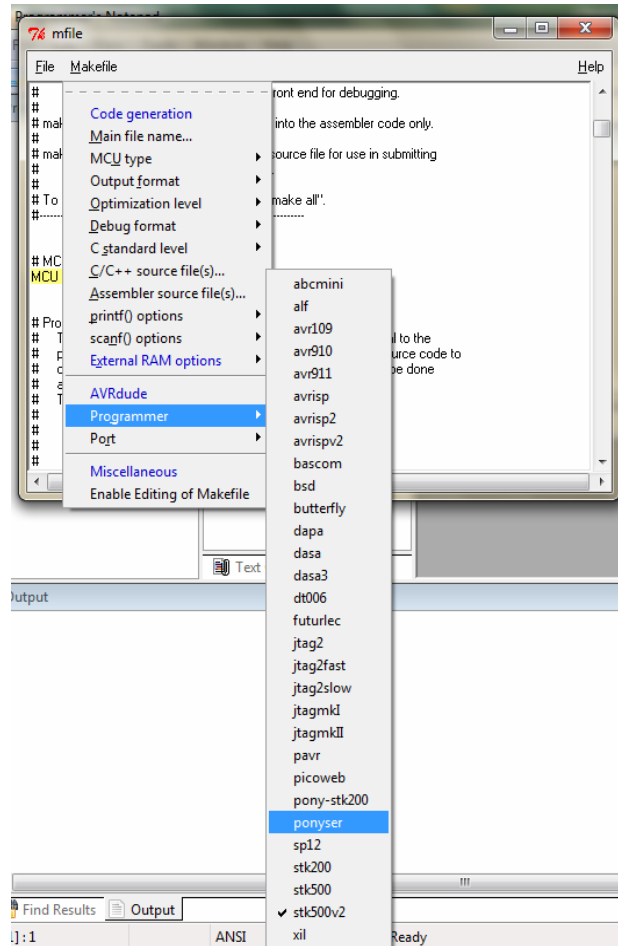


b. Tampilan setelah dipilih atmega8535

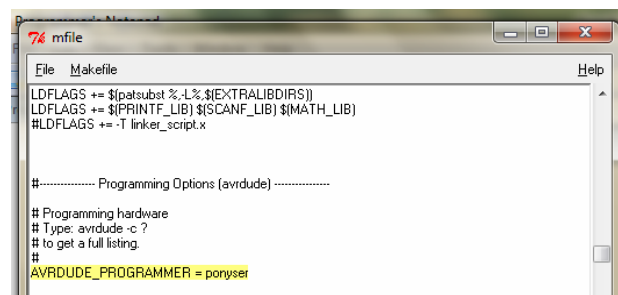


21. Pilih programmer yang digunakan pada praktikum ini: Makefile→Programmer

- a. Pilih ponyser (skematik programmer ini dapat dilihat di http://www.lancos.com/e2p/siprog_base.png)

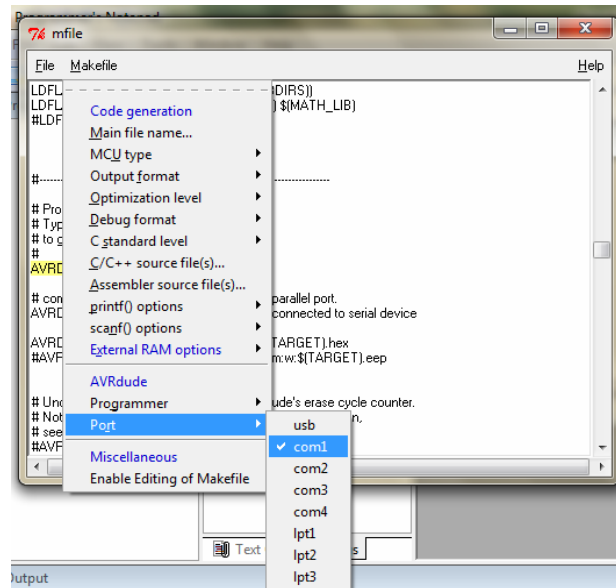


- b. Tampilan setelah dipilih ponyser

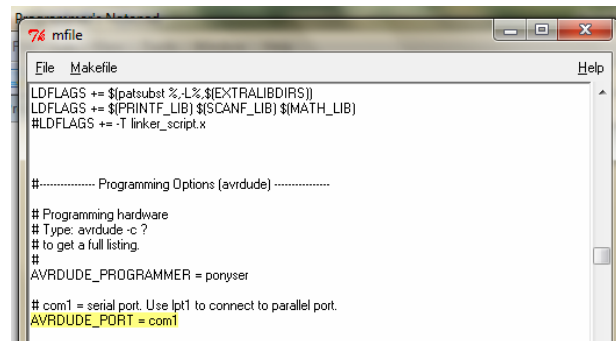


22. Pilih Port yang digunakan oleh programmer tadi: Makefile→Port

a. Pastikan/ pilih com1

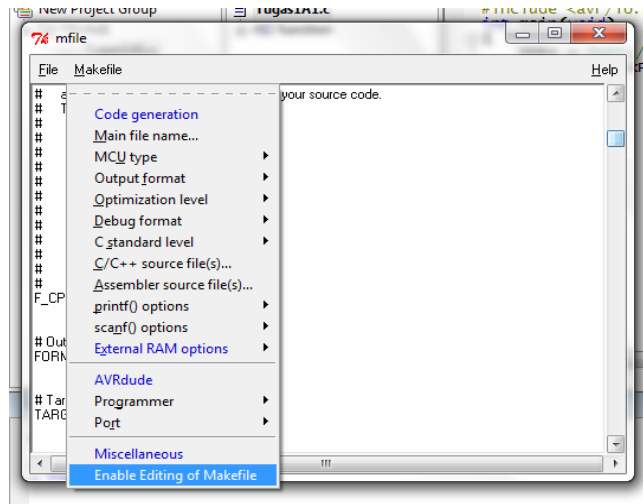


b. Tampilan setelah dipilih com1

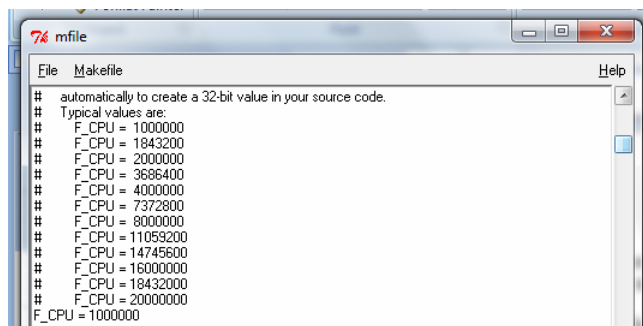


23. Atur nilai Kristal yang digunakan dengan cara mengedit “Makefile” dengan mengetikan nilai Kristal tersebut

a. Makefile → klik Enable Editing of Makefile

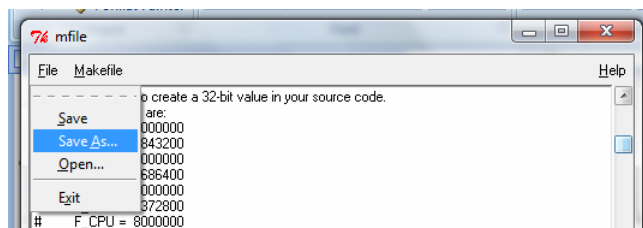


b. Lalu, edit “Makefile” dengan mengetikan nilai Kristal yang sesuai, misalnya 1 Mhz, sehingga tertulis “F_CPU = 1000000”

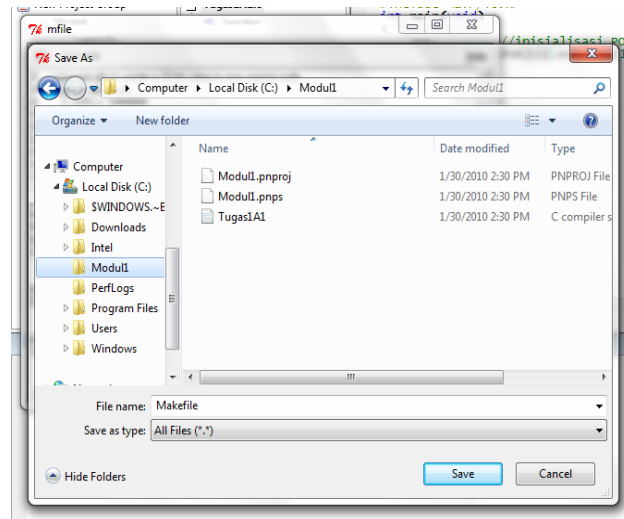


24. Simpan file “Makefile” tersebut di folder file “Tugas1A1.c” tadi berada

a. File → Save As ...

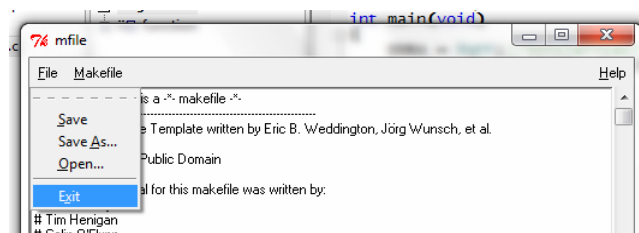


b. Pastikan berada pada folder “C:\Modul1”



c. Akhiri dengan klik Save

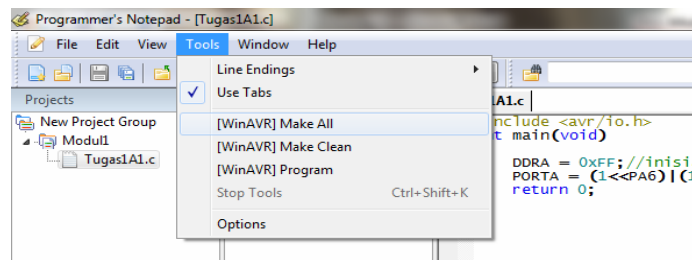
25. Terakhir, tutup program Mfile: File→Exit



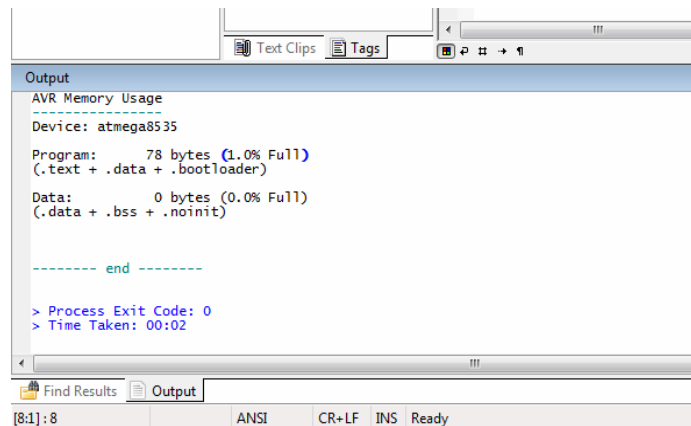
MENG-COMPILE PROGRAM DAN MEMBUAT FILE *.HEX

Berikut ini adalah langkah untuk meng-compile sekaligus membuat file *.hex (pada kasus ini adalah file “Tugas1A1.hex”) yang akan didownload ke ATmega8535

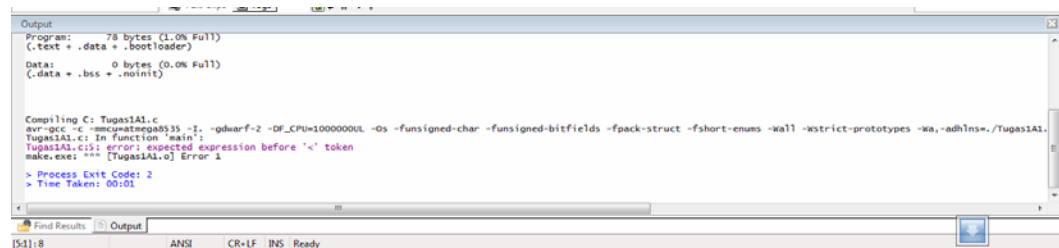
26. Pilih Tools → [WinAVR] Make All



27. Apabila tidak ada kesalahan pada program, maka pada Output ditampilkan “Process Exit Code : 0” seperti gambar berikut:



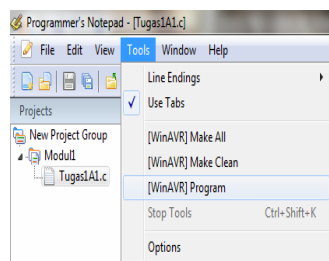
28. Apabila terjadi kesalahan, pada Output diberikan informasi kesalahan tersebut. Berikut ini adalah contohnya:



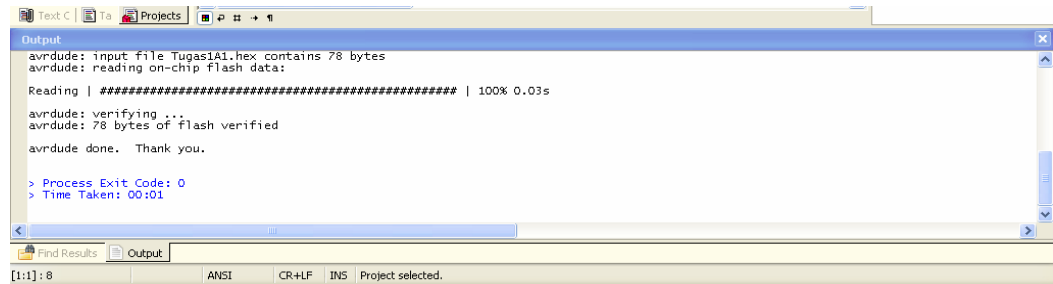
29. Perbaiki kesalahan yang terjadi, kemudian kembali ke langkah 26.

LANGKAH-LANGKAH MENDOWNLOAD *.HEX KE PROSESOR ATMEGA8535

30. Pastikan programmer telah terhubung dengan Sismin AVR dan programmer telah terhubung dengan PC
31. Pilih Tools → [WinAVR] Programm



32. Apabila proses memprogram/ mendownload “Tugas1A1.hex” ke prosesor ATMEGA8535 berhasil, maka pada Output ditampilkan “Process Exit Code : 0” seperti gambar berikut:

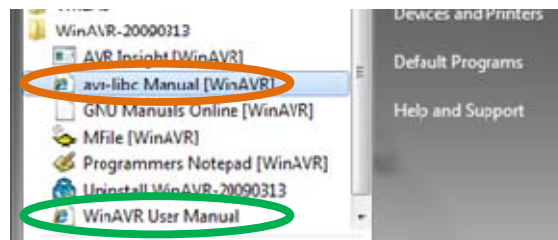


33. Apabila proses memprogram tidak berhasil, periksa kembali koneksi antara programmer dan PC serta programmer dan Sismin AVR; periksa juga kemungkinan lainnya. Kemudian, kembali ke langkah 30.

34. Setelah proses memprogram sukses, amatilah hasilnya.

MANUAL WINAVR DAN DESKRIPSI LIBRARY AVR-GCC

Manual WinAVR dan deskripsi library pada AVR-GCC dapat dilihat secara offline dengan menjalankan WinAVR User Manual (untuk manual WinAVR) dan menjalankan avr-libc Manual [WinAVR] (untuk deskripsi library pada AVR-GCC).



MENGATUR FUSE DENGAN AVRDUDE

Dengan menjalankan langkah 31. di atas, sesungguhnya prosesor tersebut diprogram dengan AVRDUDE (<http://www.nongnu.org/avrdude/>). Selain dapat memprogram seperti contoh di atas, AVRDUDE juga dapat digunakan untuk membaca dan mengatur fuse prosesor AVR.

Deskripsi Fuse ATmega8535

Berikut ini adalah cuplikan deskripsi fuse ATmega8535 (Datasheet ATmega8535 hal. 238-239, http://www.atmel.com/dyn/Products/product_card.asp?part_id=2008):

Fuse High Byte	Bit No	Description	Default Value
S8535C	7	Select AT90S8535 compatibility mode	1 (unprogrammed)
WDTON	6	WDT always on	1 (unprogrammed, WDT enabled by WDTCR)
SPIEN ⁽¹⁾	5	Enable Serial Program and Data Downloading	0 (programmed, SPI prog. enabled)
CKOPT ⁽²⁾	4	Oscillator options	1 (unprogrammed)
EESAVE	3	EEPROM memory is preserved through the Chip Erase	1 (unprogrammed, EEPROM not preserved)
BOOTSZ1	2	Select Boot Size (see Table 93 for details)	0 (programmed) ⁽³⁾
BOOTSZ0	1	Select Boot Size (see Table 93 for details)	0 (programmed) ⁽³⁾
BOOTRST	0	Select Reset Vector	1 (unprogrammed)

Fuse Low Byte	Bit no	Description	Default Value
BODLEVEL	7	Brown out detector trigger level	1 (unprogrammed)
BODEN	6	Brown out detector enable	1 (unprogrammed, BOD disabled)
SUT1	5	Select start-up time	1 (unprogrammed) ⁽¹⁾
SUT0	4	Select start-up time	0 (programmed) ⁽¹⁾
CKSEL3	3	Select Clock source	0 (programmed) ⁽²⁾
CKSEL2	2	Select Clock source	0 (programmed) ⁽²⁾
CKSEL1	1	Select Clock source	0 (programmed) ⁽²⁾
CKSEL0	0	Select Clock source	1 (unprogrammed) ⁽²⁾

Penggunaan Sumber dan Nilai Clock dengan Pengaturan Fuse

Prosesor ATMega8535 memiliki konfigurasi fuse awal (*default*) seperti pada table di atas sehingga bekerja dengan clock internal 1Mhz (tinjau konfigurasi Fuse Low Byte); lihat pula datasheet ATMega8535 hal. 29-30.

Pada praktikum ini akan digunakan pula sumber clock eksternal (kristal) 8Mhz (lihat Hardware Sismin), sehingga pengaturan ulang fuse harus dilakukan; lihat pula datasheet ATMega8535 hal 25-27.

Pada bagian ini akan diberikan petunjuk untuk membaca dan mengubah pengaturan fuse pada ATMega8535 dengan AVRdude

Langkah-Langkah Membaca Pengaturan Fuse ATMega8535 dengan AVRdude

35. Jalankan Command Prompt dengan mengikuti Langkah 7 dan 8 di atas

36. Pada Command Prompt, ketik perintah berikut, diikuti dengan tekan Enter:

```
Avrdude -c stk200 -p atmega8535 -P lpt1 -v
```

```
C:\Documents and Settings\LABDASAR>avrdude -c stk200 -p atmega8535 -P lpt1 -v
```

37. Berikut adalah contoh potongan tampilan hasilnya:

```
avrdude: AVR device initialized and ready to accept instructions
Reading ! ##### : 100% 0.00s
avrdude: Device signature = 0x1e9308
avrdude: safemode: lfuse reads as E1
avrdude: safemode: hfuse reads as D9
avrdude: safemode: lfuse reads as E1
avrdude: safemode: hfuse reads as D9
avrdude: safemode: Fuses OK
avrdude done. Thank you.

C:\Documents and Settings\LABDASAR>
```

Langkah-Langkah Mengubah Pengaturan Fuse ATmega8535 dengan AVRdude

38. Agar prosesor bekerja sumber clock dari clock eksternal (Kristal) 8MHz, ketik perintah berikut, diikuti dengan tekan Enter:

```
Avrdude -c stk200 -p atmega8535 -P lpt1 -U hfuse:w:0xD9:m -U lfuse:w:0xEF:m
```

```
C:\Documents and Settings\LABDASAR>avrdude -c stk200 -p atmega8535 -P lpt1 -U hfuse:w:0xD9:m -U lfuse:w:0xEF:m
```

MEMAHAMI KODE-KODE PADA AVRDUDE

39. Untuk memahami arti kode-kode , misalnya -c, -P dsb., pada Command Prompt ketik “avrdude” diikuti dengan tekan:

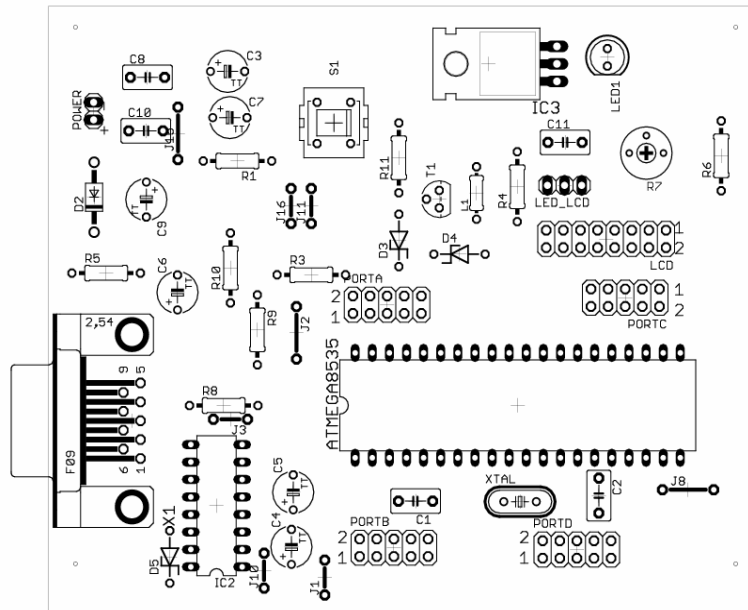
```
C:\Documents and Settings\LABDASAR>avrdude
Usage: avrdude [options]
Options:
  -p <partno>          Required. Specify AVR device.
  -b <baudrate>        Override RS-232 baud rate.
  -B <bitclock>        Specify JTAG/SPIK50002 bit clock period (us).
  -C <config-file>     Specify location of configuration file.
  -c <programmer>      Specify programmer type.
  -D                   Disable auto erase for flash memory
  -i <delay>           ISP Clock Delay (in microseconds)
  -P <port>            Specify connection port.
  -F                   Override invalid signature check.
  -e                   Perform a chip erase.
  -O                   Perform RC oscillator calibration (see AVR053).
  -U <memtype>:r:w:w:<filename>[:format]
                        Memory operation specification.
                        Multiple -U options are allowed, each request
                        is performed in the order specified.
  -n                   Do not write anything to the device.
  -U                   Do not verify.
  -u                   Disable safemode, default when running from a script.
  -s                   Silent safemode operation, will not ask you if
                        fuses should be changed back.
  -t                   Enter terminal mode.
  -E <exitspec>[:l,<exitspec>] List programmer exit specifications.
  -x <extended_param>  Pass <extended_param> to programmer.
  -v                   Count # erase cycles in EEPROM.
  -V <number>          Initialize erase cycle # in EEPROM.
  -v                   Verbose output. -v -v for more.
  -q                   Quell progress output. -q -q for less.
  -?                   Display this usage.

avrdude project: <URL:http://savannah.nongnu.org/projects/avrdude>
```


APENDIKS B KIT PRAKTIKUM

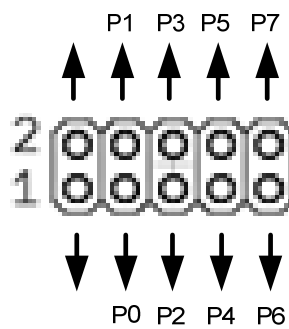
SISTEM MINIMUM ATMEGA8535

TATA LETAK

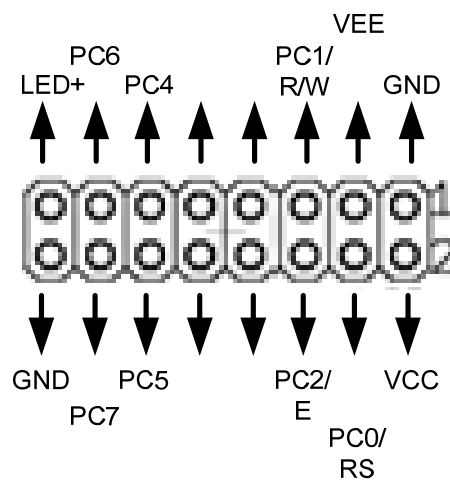


INFORMASI PIN

PORT I/O

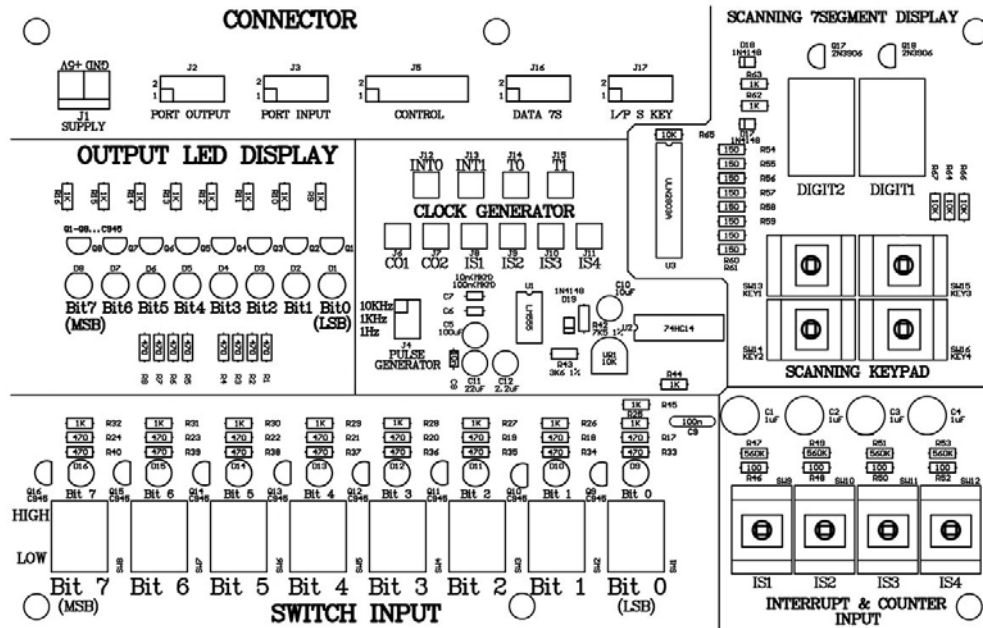


PORT LCD

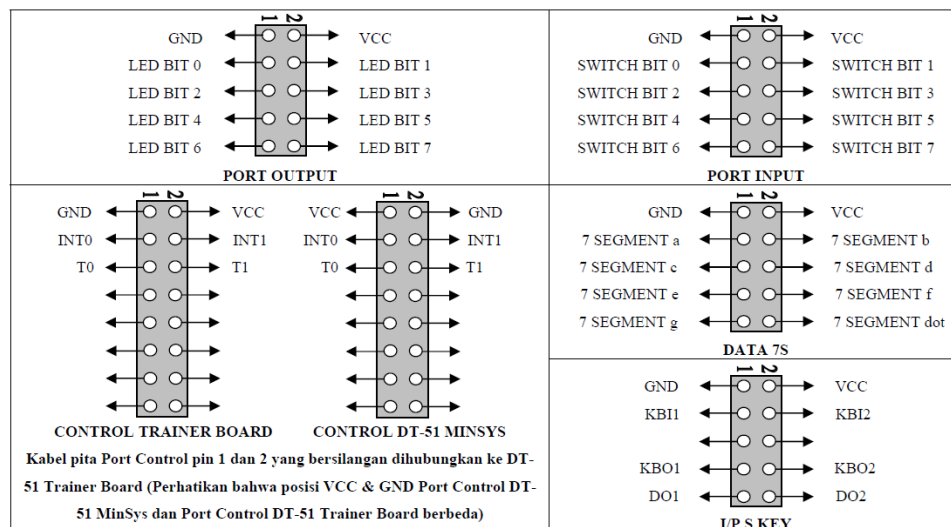


DT-51 TRAINER BOARD

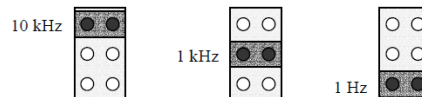
TATA LETAK



KONEKTOR



Setting Jumper J4 untuk Pulse Generator:



Output Pulse pada CO1 dan CO2 (inverted)

APENDIKS C KODE ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

128	Ç	144	É	161	í	177	☐	193	⊥	209	〒	225	ß	241	±
129	ü	145	æ	162	ó	178	☐	194	⊥	210	⊥	226	Γ	242	≥
130	é	146	Æ	163	ú	179		195	⊥	211	⊥	227	π	243	≤
131	â	147	ô	164	ñ	180	†	196	—	212	⊥	228	Σ	244	∫
132	ä	148	ö	165	Ñ	181	†	197	†	213	⊥	229	σ	245	∫
133	à	149	ò	166	•	182	‡	198	†	214	⊥	230	μ	246	÷
134	â	150	û	167	°	183	‡	199	‡	215	‡	231	τ	247	≈
135	ç	151	ù	168	¿	184	‡	200	⊥	216	‡	232	Φ	248	°
136	ê	152	—	169	—	185	‡	201	⊥	217	‡	233	Θ	249	•
137	ë	153	Ö	170	¬	186	‡	202	⊥	218	‡	234	Ω	250	•
138	è	154	Û	171	½	187	‡	203	⊥	219	■	235	δ	251	√
139	ï	156	£	172	¼	188	‡	204	‡	220	■	236	∞	252	—
140	î	157	¥	173	¡	189	‡	205	=	221	■	237	φ	253	²
141	ì	158	—	174	«	190	‡	206	‡	222	■	238	ε	254	■
142	À	159	f	175	»	191	‡	207	⊥	223	■	239	∩	255	
143	Ä	160	á	176	☐	192	⊥	208	⊥	224	α	240	=		

Source: www.LookupTables.com