

JOBSHEET II

ANTARMUKA MIKROKONTROLER DENGAN TOGGLE SWITCH

TUJUAN

- Mengetahui dan memahami cara mengantarmukakan mikrokontroler dengan rangkaian input saklar toggle.
- Mengetahui dan memahami bagaimana memrogram mikrokontroler untuk membaca data input dari saklar toggle.

SAKLAR TOGGLE

Saklar toggle adalah salah satu saklar elektrik yang digerakkan secara manual oleh batang mekanik. Saklar toggle tersedia dalam berbagai bentuk dan ukuran, serta digunakan dalam berbagai aplikasi. Gambar dibawah memperlihatkan bentuk fisik dan symbol saklar toggle.

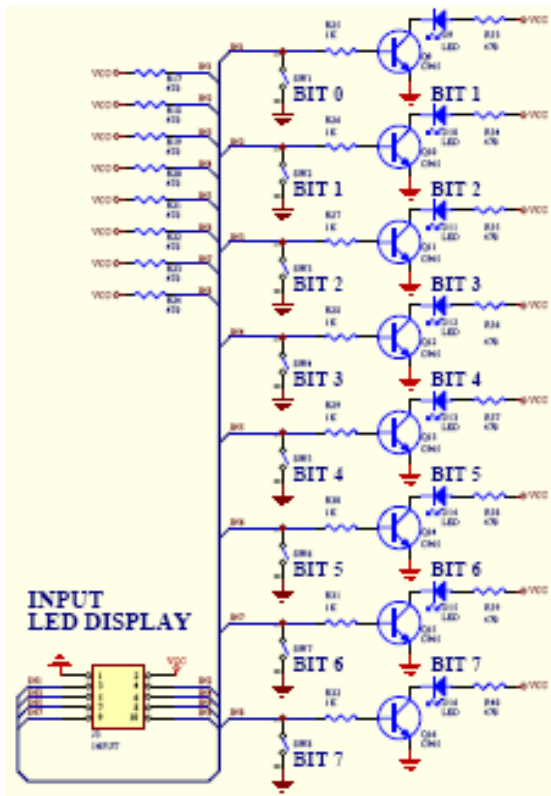


(a) Salah satu toggle switch

Toggle switch



(b) Simbol elektronik



Dalam modul I/O yang dipakai dalam praktikum, saklar toggle yang digunakan diberi rangkaian tambahan berupa rangkaian LED untuk melihat secara langsung logika yang ada pada saklar. Rangkaian LED sama dengan rangkaian dalam Modul I. Kedelapan rangkaian saklar tersebut dihubungkan ke sebuah soket jumper bernama INPUT.

PERINTAH DASAR MEMBACA DATA

- Sebelum membaca data, perlu dibuat deklarasi variabel untuk data yang dimasukkan. Data bisa bertipe char. Deklarasi variabel diletakkan di variabel lokal pada main program.

```
void main (void)
{
//Declare your local variables here
unsigned char data_in;
...

```

- Untuk membaca data byte ke PORTX (X=A, B, C, D) digunakan statement
`data_in = PINX;`
contoh: `data_in = PINB;`
- Untuk membaca data bit ke PORTX.Y (X=A, B, C, D dan Y=0, 1, 2, 3, 4, 5, 6, 7) digunakan statement

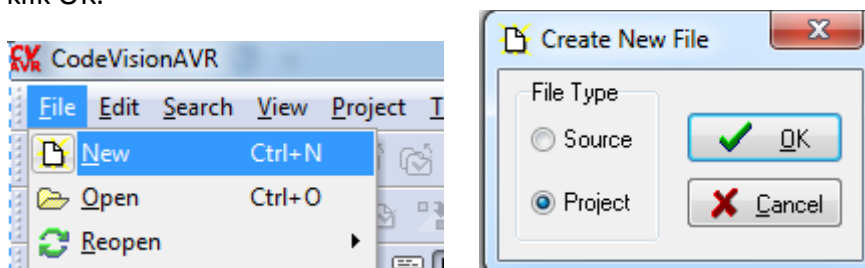
```
data_in = PINX.Y;
contoh: data_in = PINB.1;
```

ALAT DAN BAHAN YANG DIGUNAKAN

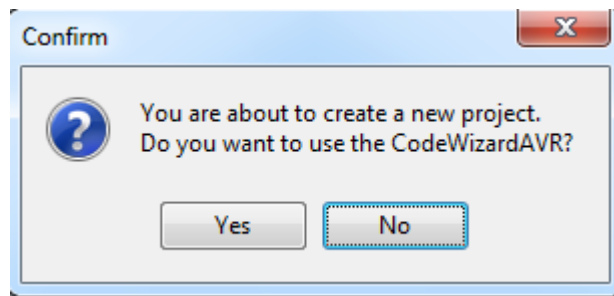
- 1 set PC/Laptop yang sudah berisi program Code Vision dan Khazama
- 1 buah catu daya DC +5V
- 1 buah multimeter
- 1 buah ISP Downloader AVR
- 1 buah sistem minimum AVR
- 1 buah I/O
- 1 buah kabel printer USB
- 2 buah kabel pita hitam

PROSEDUR

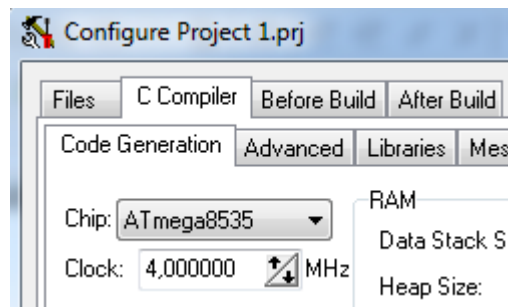
1. Hubungkan soket jumper PORTB pada minimum system dengan soket jumper OUTPUT pada I/O dan soket jumper PORTC pada minimum system dengan soket jumper INPUT pada I/O
2. Buka program Code Vision AVR
3. Buat file baru dengan cara klik File>>New, atau klik icon Create a New File or Project, atau ketik Ctrl + N. Kemudian muncul dialog box, pilih Project pada File Type, kemudian klik OK.



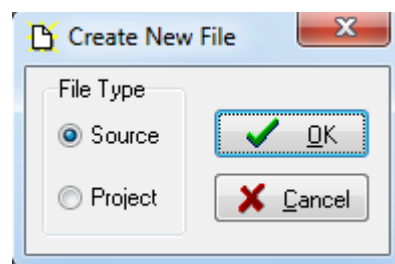
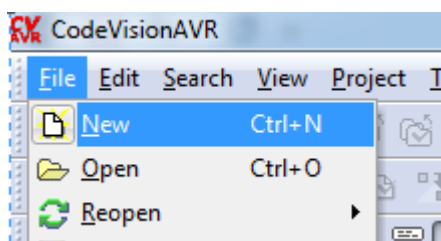
4. Kemudian klik No ketika muncul dialog box untuk menanyakan apakah ingin menggunakan CodeWizardAVR



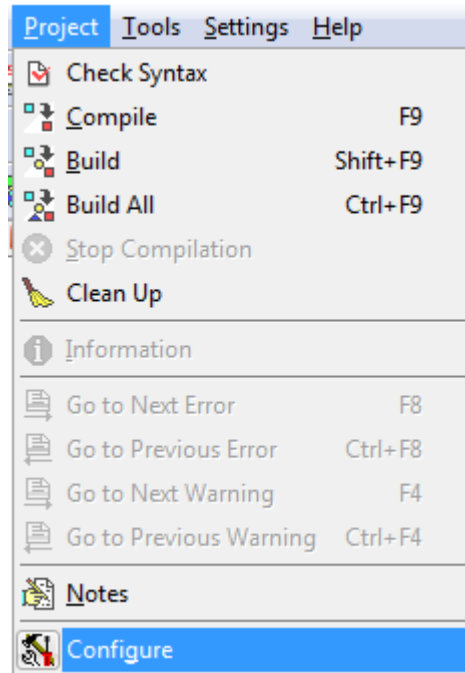
5. Kemudian pilih lokasi penyimpanan file dan buat folder baru dengan nama "praktikum 2" untuk menyimpan file.
6. Kemudian simpan file pada folder tersebut dan berinama file dengan nama "saklar" dan klik Save.
7. Kemudian muncul dialog box Configure Project. Pada subtab Code Generation pada tab C Compiler pilih chip ATmega8535 dan ubah Clock menjadi 4 Mhz. Kemudian klik OK



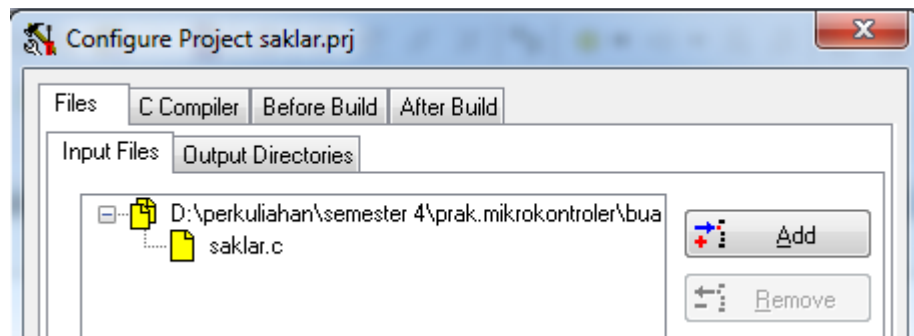
8. Buat file Source baru dengan cara klik File>>New, atau klik icon Create a New File or Project, atau ketik Ctrl + N. Kemudian muncul dialog box, pilih Source pada File Type, kemudian klik OK. Kemudian Save as file Source itu dengan nama "saklar" dan simpan di dalam direktori yang sama yaitu di dalam folder "praktikum 2".



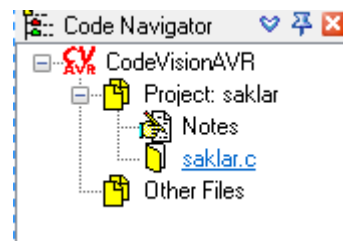
9. Hubungkan file Project dengan File Source dengan Cara klik Project>> Configure



10. Pada dialog box, klik Add pada tab Input Files. Kemudian pilih file yang sudah anda simpan tadi dan klik Open. Kemudian klik OK.



11. Jika sudah ditambahkan, maka pada Code Navigator akan terlihat saklar.c.



12. Tambahkan file header

```
#include <mega8535.h>
```

13. Buat program utama dan inialisasikan PORTB sebagai output (DDRB = FFH) dan output value = 0 (PORTB=00H), dan PORTC sebagai input (DDRC = 00H) dan resistor pullup dihubungkan (PORTC=FFH) sehingga pada program bagian inialisasi PORTB dan PORTC terlihat sebagai berikut:

```
void main(void)
{

    PORTB=0x00;
    DDRB=0xFF;
    PORTC=0xFF;
    DDRC=0x00;

    while (1)
    {

    }

}
```

14. Tambahkan deklarasi variabel lokal di bawah ini dalam program utama.

```
unsigned char data_in;
```

15. Tuliskan dalam program utama tepatnya tuliskan program didalam while (1)

```
while (1)
{
    data_in=PINC;
    PORTB=data_in;
}

}
```

16. Compile program tersebut dengan cara klik Project>> Compile atau ketik F9.
 17. Jika ada *error*, koreksi program yang *error* dan perbaiki. Jika sudah tidak ada yang *error* build program tersebut dengan cara klik Project>> Build atau klik Shift + F9.
 18. Hubungkan kabel usb printer antara PC/ Laptop dengan trainer.
 19. Buka aplikasi Khazama AVR Programmer. Pilih ATmega8535 pada pilihan chip AVR. Kemudian buka program kompilasi saklar.hex biasanya terdapat pada folder atau sub direktori.. \Exe yang terdapat pada folder "praktikum 2" dengan cara klik file>> Load Flash File to Buffer atau ketik Ctrl + L kemudian klik Open.
 20. Klik Auto Program kemudian pindahkan saklar sesuai tabel 2.1 dibawah, amati dan catat nyala LED.

Tabel 2.1 Program Saklar1

No.	Saklar								LED							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	1								
2	0	0	0	0	0	0	1	1								
3	0	0	0	0	0	1	0	1								
4	0	0	0	0	1	0	0	1								
5	0	0	0	1	0	0	0	1								
6	0	0	1	0	0	0	0	1								
7	0	1	0	0	0	0	0	1								
8	1	0	0	0	0	0	0	1								
9	1	1	1	1	1	1	1	1								

21. Hapus program yang ada pada program utama tepatnya program di dalam while (1) kemudian tuliskan program di dalam while(1).

```
while (1)
{
    data_in = PINC.0;
    PORTB.7 = data_in;
}
}
```

22. Compile dan Build program, jika ada yang error perbaiki program. Masukkan file hex menggunakan Khazama AVR Programmer. Klik auto program.
23. Pindahkan saklar sesuai tabel 3.1 dibawah, amati dan catat nyala LED.

Tabel 3.1 Program Saklar2

No.	Saklar								LED							
	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
1	0	0	0	0	0	0	0	1								
2	0	0	0	0	0	0	1	1								
3	0	0	0	0	0	1	0	0								
4	0	0	0	0	1	0	0	1								
5	0	0	0	1	0	0	0	0								
6	0	0	1	0	0	0	0	1								
7	0	1	0	0	0	0	0	0								

24. Buat program untuk masing–masing algoritma berikut dengan menggunakan if :
- Jika switch bit 0 berlogika 1 maka LED bit 4 menyala.
 - Jika switch bit 1 berlogika 1 maka LED bit 7 menyala.
25. Buat satu program untuk masing–masing algoritma berikut dengan menggunakan if-else.
- Jika switch bit 0 berlogika 1 maka LED bit 4 menyala.
 - Jika switch bit 1 berlogika 1 maka LED bit 7 menyala.
 - Jika switch bit 2 berlogika 1 maka LED menyala dengan konfigurasi ON-OFF-ON-OFF-ON-OFF-ON-OFF (bit7-bit0).
 - Jika switch bit 3 berlogika 1 maka LED menyala dengan konfigurasi OFF-ON-OFF-ON-OFF-ON-OFF-ON (bit7-bit0).
 - Jika switch bit 4 berlogika 1 maka LED menyala semua.
 - Jika switch bit 5 berlogika 1 maka LED mati semua.
 - Jika switch bit 6 berlogika 1 maka LED menyala berurutan mulai dari bit0 ke bit7 lalu kembali lagi ke bit0 dengan delay 1 detik
 - Jika switch bit 7 berlogika 1 maka LED akan menyala dengan nilai sama dengan dua digit terakhir NIM anda.

DATA HASIL PERCOBAAN

- Tabel 2.1
- Tabel 3.1
- Source code program pada percobaan langkah 9
- Source code program pada percobaan langkah 10

ANALISIS DATA

- Analisis Program Saklar1 dan Saklar2.
- Instruksi apa yang digunakan untuk membaca data input?
- Apakah perbedaan program yang menggunakan if dan if-else?