MODUL 6
Meter Cahaya Digital

I. Pendahuluan

Pada praktikum ini, anda akan mencoba memanfaatkan fasilitas masukan analog pada mikrokontroler AVR ATmega8535. ATmega8535 mempunyai ADC (Analog to Digital Converter) internal 10-bit sebanyak 8 kanal. Modul I/O yang digunakan pada praktikum ini mempunyai 3 buah resistor peka cahaya atau LDR (Light Dependent Resistor), 1 buah sensor suhu LM35, dan 1 buah mikrofon yang terhubung pada masukan analog ATmega8535. Cahaya yang diterima oleh sebuah LDR akan dikonversi menjadi tegangan listrik. Kemudian nilai tegangan listrik dari LDR diubah menjadi kode biner oleh ADC untuk diinformasikan pada layar LCD. Inisialisasi ADC dilakukan menggunakan bantuan CodeWizardAVR dan program ditulis dengan bahasa-C.

II. Tujuan

Setelah menyelesaikan praktikum ini, yang anda peroleh adalah :

- dapat mengimplementasikan fungsi masukan analog pada mikrokontroler AVR yang mendapat masukan dari LDR sebagai sensor cahaya.
- dapat mengimplementasikan fungsi kontrol LCD untuk menampilkan nilai tegangan yang diterima LDR.
- dapat mengimplementasikan fungsi input/output untuk memberikan nilai ambang batas (threshold) untuk menyalakan LED sebagai indikator.
- dapat mengimplementasikan sebuah sistem meter cahaya digital pada AVR ATmega8535 dengan pemrograman bahasa-C.

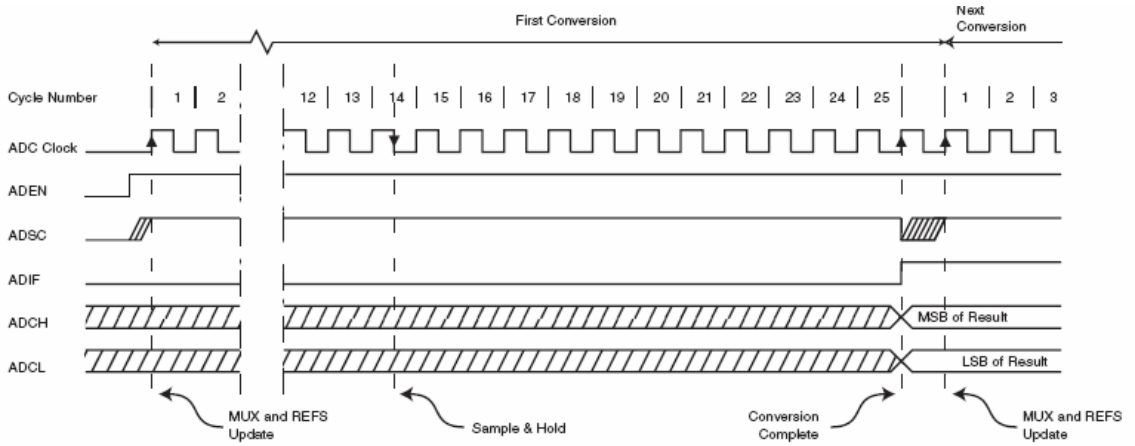
III. Gambaran Disain

Cahaya yang diterima oleh permukaan LDR akan mengubah nilai resistansinya. Dengan konfigurasi pembagi tegangan seperti ditunjukkan oleh gambar 2, maka tegangan jatuh pada resistor R akan berubah nilainya. Tegangan ini dihubungkan dengan masukan analog dari ATmega8535 untuk dikonversi menjadi kode biner dan akan ditampilkan pada layar LCD. Inisialisasi ADC dapat dilakukan dengan bantuan CodeWizardAVR. Lebar data ADC yang digunakan adalah 8-bit. Fasilitas interrupt ADC diaktifkan yang berarti bahwa bila ADC telah selesai melakukan konversi maka program akan meloncat menuju rutin layanan interupsi yang sesuai. Pada tugas yang akan dikerjakan, anda membaca nilai tegangan pada ketiga LDR secara bergantian dan ditampilkan pada layar LCD secara bergantian pula. Kemampuan pemrograman bahasa-C sangat diperlukan pada praktikum ini.

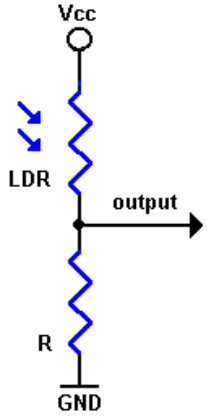
IV. Dasar Teori

Board AVR yang digunakan menggunakan chip ATmega8535. IC ini mempunyai ADC (Analog to Digital Converter) internal dengan fitur sebagai berikut (untuk lebih detil dapat mengacu pada datasheet)

- 10-bit Resolution
- 65 - 260 µs Conversion Time
- Up to 15 kSPS at Maximum Resolution
- 8 Multiplexed Single Ended Input Channels
- Optional Left Adjustment for ADC Result Readout
- 0 - VCC ADC Input Voltage Range
- Selectable 2.56V ADC Reference Voltage
- Free Running or Single Conversion Mode
- ADC Start Conversion by Auto Triggering on Interrupt Sources
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler



Gambar 1. Timing diagram untuk Mode single-conversion



Gambar 2. Konfigurasi LDR untuk masukan analog ke ATmega8535

Register-register yang dipakai untuk mengakses ADC adalah:

- ADMUX – ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
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 – REFS1:0 : Bit Pemilih tegangan referensi

REFS1	REFS0	Voltage Reference Selection
0	0	AREF, Internal Vref turned off
0	1	AVCC with external capacitor at AREF pin
1	0	Reserved
1	1	Internal 2.56V Voltage Reference with external capacitor at AREF pin

Bit 5 – ADLAR : ADC Left Adjust Result

Bit 4:0 – MUX4:0 : Bit pemilih Analog Channel dan Gain

MUX4..0	Single Ended Input	Pos Differential Input	Neg Differential Input	Gain
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000	N/A	ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010		ADC0	ADC0	200x
01011		ADC1	ADC0	200x
01100		ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110		ADC2	ADC2	200x
01111		ADC3	ADC2	200x
10000		ADC0	ADC1	1x
10001		ADC1	ADC1	1x
10010		ADC2	ADC1	1x
10011		ADC3	ADC1	1x
10100		ADC4	ADC1	1x
10101		ADC5	ADC1	1x
10110		ADC6	ADC1	1x
10111		ADC7	ADC1	1x
11000		ADC0	ADC2	1x
11001		ADC1	ADC2	1x
11010		ADC2	ADC2	1x
11011		ADC3	ADC2	1x
11100		ADC4	ADC2	1x

MUX4..0	Single Ended Input	Pos Differential Input	Neg Differential Input	Gain
11101		ADC5	ADC2	1x
11110	1.22V (V_{BG})	N/A		
11111	0V (GND)			

- ADCSRA – ADC Control and Status Register

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
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 – ADEN : ADC Enable
Diisi 1 untuk mengaktifkan ADC, diisi 0 untuk mematikan ADC sekaligus memberhentikan konversi yang sedang berlangsung.

Bit 6 – ADSC : ADC Start Conversion
Pada mode single-conversion, set bit ini untuk memulai tiap konversi.
Pada mode free-running, set bit ini untuk konversi pertama kalinya.
Bit ADSC bila dibaca akan bernilai 1 selama proses konversi, dan bernilai 0 bila konversi selesai. Mengisi bit ini dengan nilai 0 tidak akan mempunyai dampak.

Bit 5 – ADATE : ADC Auto Trigger Enable
Bila bit ini diisi 1 maka auto trigger ADC akan diaktifkan. ADC akan memulai konversi pada saat tepi positif dari sumber sinyal trigger yang dipilih. Sumber sinyal trigger ditentukan dengan menseting bit ADTS pada register SFIOR.

Bit 4 – ADIF : ADC Interrupt Flag
Bit ini akan bernilai 1 pada saat ADC selesai mengkonversi dan Data register telah diupdate. ADC Conversion Complete Interrupt akan dijalankan bila bit ADIE dan bit-I pada register SREG diset 1. ADIF akan di-clear secara hardware bila mengerjakan penanganan vektor interrupt yang bersesuaian. Alternatifnya, ADIF dapat di-clear dengan menuliskan 1. Hati-hati bila bekerja dengan Read-Modify-Write pada ADCSRA, interrupt yang tertunda dapat dinonaktifkan/batal. Hal ini juga berakibat sama bila instruksi SBI dan CBI digunakan.

Bit 3 – ADIE : ADC Interrupt Enable
Mengisi bit ini dan bit-I pada register SREG menjadi 1 akan mengaktifkan ADC Conversion Complete Interrupt.

Bit 2:0 – ADPS2:0 – Bit pemilih ADC Prescaler
Menentukan bilangan pembagi antara sumber clock XTAL ke clock ADC.

ADPS2	ADPS1	ADPS0	Division Factor
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

- ADCL, ADCH – ADC data register

Bila ADCLAR = 0

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	–	–	ADC9	ADC8	ADCH
	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Bila ADCLAR = 1

Bit	15	14	13	12	11	10	9	8	
	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
	ADC1	ADC0	–	–	–	–	–	–	ADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Setelah ADC selesai melakukan konversi kedua register ini berisi hasil konversi. Bila channel differensial dipilih maka hasilnya dalam format two's complement. Saat ADCL dibaca, data register tidak akan meng-update data sampai ADCH dibaca. Jika hasilnya dirata kiri (left adjust) dan hanya butuh 8-bit maka cukuplah dengan membaca ADCH. Jika butuh 10-bit, baca ADCL dahulu kemudian ADCH.

- SFIOR – Special Function I/O Register untuk sumber auto triger

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	

Bit 7:5 – ADTS2:0 : ADC Auto Trigger Source
Bila ADATE dalam register ADCSRA diset 1, maka nilai dalam bit-bit ini akan menentukan sumber mana yang akan mentrigger konversi ADC. Bila bit ADATE bernilai 0, maka bit-bit ini tidak akan mempunyai efek. Sebuah konversi ditrigger oleh sinyal rising-edge dari interrupt flag yang dipilih. Perlu diingat bahwa memindah sumber trigger yang di-clear ke sumber trigger lain yang di-set akan menyebabkan positive-edge pada sinyal trigger. Bila ADEN dalam register ADCSRA diset, juga akan memulai konversi. Memindah mode ke mode free-running tidak akan menyebabkan pulsa trigger, meskipun bila flag interrupt ADC diset.

ADTS2	ADTS1	ADTS0	Trigger Source
0	0	0	Free Running mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event

Bit 4 – RES : Reserved bit
Bit cadangan, bila dibaca hasilnya nol.

Berikut adalah langkah-langkah yang harus dilakukan untuk melakukan konversi mode single-conversion pada ADC internal dari ATmega8535:

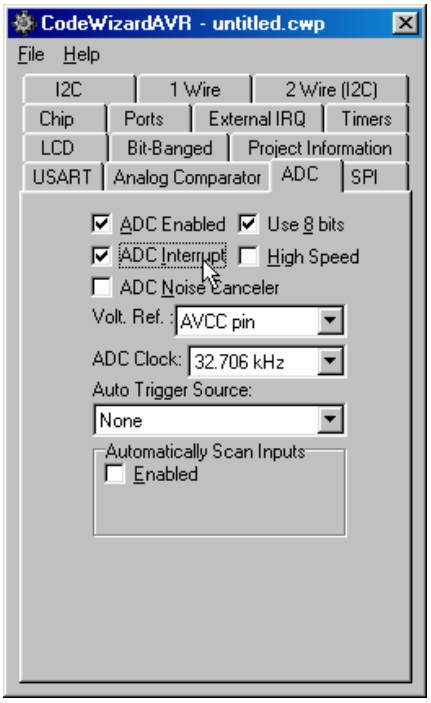
1. Seting register ADMUX.
2. Seting register ADCSRA.
3. Seting register SFIOR (untuk nomor 1, 2, dan 3 dilakukan oleh CodeWizardAVR).
4. Pilih channel ADC pada register ADMUX.
5. Start konversi ADC dengan cara mengaktifkan bit ADSC pada register ADCSRA.
6. Bila ADC selesai melakukan konversi, program akan meloncat ke layanan interupsi ADC.
7. Ambil data pada register ADCH.
8. Tampilkan nilai pada LCD.
9. Delay sekitar 500 milidetik, agar tampilan pada LCD dapat dilihat.
10. Jika diinginkan, ulangi langkah 4.

V. Peralatan

- 1 set PC yang dilengkapi dengan software CodeVision AVR.
- 1 set development board AVR ATMega8535
- 1 power-supply +9VDC

VI. Prosedur Praktikum

1. Jalankan software CodeVisionAVR.
2. Buatlah project baru, dengan menggunakan CodeWizardAVR:
    a. menseting chip menggunakan ATmega8535L pada clock 4 MHz
    b. menseting LCD pada portB
    c. menseting ADC seperti pada gambar 2.



Gambar 2. Seting LCD dan ADC pada CodeWizardAVR

3. Pada CodeWizardAVR tersebut pilih **File → Generate, Save and Exit**. Beri nama dan simpan pada direktori baru didalam direktori e:\embedded.

4. Pada source program-C yang dihasilkan, tambahkan header `delay.h` dan `stdio.h`.

5. Pada bagian bawah didalam main-program, tambahkan perintah (yang berwarna merah) berikut

```
// LCD module initialization
lcd_init(16);

// Tambahkan baris berikut
lcd_gotoxy(0,0);
lcd_putsf("Cahaya pada LDR");
lcd_gotoxy(0,1);
lcd_putsf("adalah =");

// Pilih ADC channel ke-7
ADMUX = ADMUX | 0x07;

// Start ADC !
ADCSRA = ADCSRA | 0x40;

// Global enable interrupts
#asm("sei")
```

6. Pada rutin layanan interupsi ADC, tambahkan perintah berikut

```
// ADC interrupt service routine
interrupt [ADC_INT] void adc_isr(void)
{
unsigned char adc_data;
char kalimat[10];
// Read the 8 most significant bits
```

```
// of the AD conversion result
adc_data=ADCH;
// Place your code here

sprintf(kalimat,"%d ",adc_data);
lcd_gotoxy(9,1);
lcd_puts(kalimat);

delay_ms(500);

// Start ADC again !
ADCSRA = ADCSRA | 0x40;

}
```

7. Lakukan check syntax, compile dan make. Bila tidak ada error, programkan kedalam flash. Cobalah untuk membuka dan menutup LDR, perhatikan tampilan pada layar LCD.

VII. Laporan sementara

Tugas berikut dikerjakan sebagai laporan sementara.

Bila digunakan sebagai konverter analog maka 8-bit portA dari ATmega8535 berfungsi sebagai 8-input untuk ADC. Pada development board, portA terhubung dengan beberapa perangkat yaitu:

PortA	7	6	5	4	3	2	1	0
Perangkat	LDR3	LDR2	-	LDR1	-	LM35	-	MIC

Rancanglah disain software untuk menampilkan nilai hasil konversi tegangan pada ketiga LDR !

VIII. Laporan resmi

Sebagai analisa pada laporan resmi, gambarkan dan jelaskan diagram alir dari tugas yang anda kerjakan. Berilah keterangan pada program yang telah dibuat.

IX. Tambahan

Berikan saran atau komentar guna pengembangan lebih lanjut praktikum ini.