

JOBSHEET VI
MENGGUNAKAN ANALOG TO DIGITAL CONVERTER (ADC)
DALAM MIKROKONTROLLER ATMEGA8535

TUJUAN

- Mengetahui dan memahami cara menggunakan ADC yang ada di dalam mikrokontroler.
- Mengetahui dan memahami bagaimana memrogram mikrokontroler untuk mengonversi data analog menjadi data digital.

ANALOG TO DIGITAL CONVERTER (ADC)

Penggunaan ADC sebagai pengonversi data analog menjadi data digital merupakan sesuatu hal yang diperlukan jika data yang masuk ke dalam mikrokontroler, biasanya data dari sensor berupa sinyal analog.

Fitur ADC dalam ATMEGA8535 adalah sebagai berikut:

- Resolusi 10 bit.
- Waktu konversi 65-260 μ s.
- Input 8 kanal.
- Input ADC 0-5Vcc.
- 3 Mode pemilihan tegangan referensi.

Ada beberapa langkah yang harus dilakukan untuk inialisasi ADC, yaitu penentuan clock, tegangan referensi, format data output dan mode pembacaan. Inialisasi ini dilakukan pada register-register berikut:

ADMUX (ADC Multiplexer Selection Register)

ADMUX merupakan register yang mengatur tegangan referensi yang digunakan ADC, format data output dan saluran ADC.

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	

- ***REFS0-1 (Reference Selection Bits)***

REFS0-1 adalah bit-bit pengatur mode tegangan referensi ADC.

Table 84. Voltage Reference Selections for ADC

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

- ***ADLAR (ADC Left Adjust Result)***

ADLAR adalah bit keluaran ADC. Jika ADC telah selesai konversi, maka data ADC akan diletakkan di 2 register, yaitu ADCH dan ADCL dengan format sesuai ADLAR.

Format data ADC jika ADLAR=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	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

Format data ADC jika ADLAR=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	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

- **MUX0-4 (Analog Channel and Gain Selection Bits)**
MUX0-4 adalah bit-bit pemilih saluran pembacaan ADC.

Table 85. Input Channel and Gain Selections

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		N/A	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	N/A		ADC0	ADC2	1x	
11001			ADC1	ADC2	1x	
11010			ADC2	ADC2	1x	
11011			ADC3	ADC2	1x	
11100			ADC4	ADC2	1x	
11101			ADC5	ADC2	1x	
11110			1.22V (V _{BO})	N/A		
11111			0V (GND)			

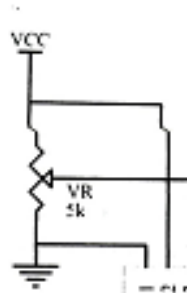
ADCSRA (ADC Control and Status Register A)

Table 87. ADC Auto Trigger Source Selections

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

Rangkaian yang digunakan untuk mempelajari ADC dapat dilihat dalam Gambar dibawah. Rangkaian ini merupakan rangkaian pembagi tegangan dimana tegangan keluaran dapat dihitung dengan rumus:

$$V_{out} = \frac{R_2 \times V_{cc}}{R_1 + R_2}$$

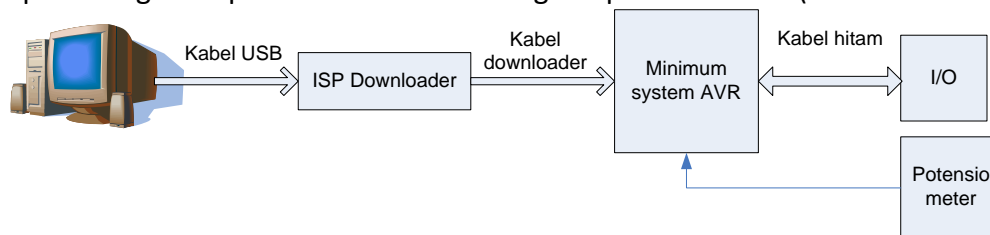


ALAT DAN BAHAN YANG DIGUNAKAN

- 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
- 1 buah kabel pita hitam
- 1 buah potensiometer

PROSEDUR

1. Rangkailah peralatan yang diperlukan seperti dalam Gambar diatas. Hubungkan soket jumper PORTC pada minimum system dengan soket jumper pada OUTPUT LED. Vout pada rangkaian potensiometer dihubungkan pada PORTA.0 (ADC channel 0).



2. Buka program Code Vision AVR
3. Buat program dengan menggunakan aplikasi Code Vision AVR
4. Buatlah file project (.prj) kemudian pilih IC yang digunakan (ATmega8535) dan atur clock 4.000 Mhz. (seperti praktikum sebelumnya)
5. Buatlah file source (.c) kemudian hubungkan file project dengan file source seperti pada praktikum sebelumnya.
6. Tambahkan file header


```
#include <mega8535.h>
#include <delay.h>
```
7. Program berikut merupakan deklarasi variabel hasil konversi ADC. Tuliskan variabel berikut di luar program utama.


```
// Declare your local variables here
unsigned char adcdt;
```
8. Buatlah program utama dan inialisasikan PORT C sebagai output dengan kondisi awal LOW.


```
PORTC=0x00;
DDRC=0xff; //PORTC sebagai output
```
9. Tuliskan inialisasi ADC berikut di dalam program utama.


```
// ADC initialization
// ADC Clock frequency: 31.250 kHz
// ADC Voltage Reference: AVCC pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: Free Running
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0xA7;
SFIOR&=0x0F;
```

10. Perhatikan blok program berikut. Arti dari blok instruksi tersebut adalah setting ADC di PORTA dan inisialisasi ADC. Tuliskan fungsi berikut di luar program utama.

```
#define ADC_VREF_TYPE 0x60
// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
  ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
  // Delay needed for the stabilization of the ADC input voltage
  delay_us(10);
  // Start the AD conversion
  ADCSRA|=0x40;
  // Wait for the AD conversion to complete
  while ((ADCSRA & 0x10)==0);
  ADCSRA|=0x10;
  return ADCH;
}
```

11. Tuliskan program berikut dalam program utama tepatnya di dalam while(1).

```
while (1)
{
  // Place your code here
  adcdt=read_adc(0); //mengambil hasil konversi
  PORTC=adcdt;      //ditampilkan ke LED
}
```

12. Compile dan Build program jika ada yang error perbaiki program. Masukkan file hex menggunakan Khazama AVR Programmer. Klik auto program.
13. Hubungkan output potensiometer pada PORTA.0.
14. Ukur tegangan potensiometer (kaki tengah) sebesar 0-5V (sesuai tabel).
15. Lepas probe AVO Meter lalu perhatikan dan catat nyala LED dan konversi nilai dalam desimal.
16. Hitung perhitungan nilai digital dalam desimal dengan rumus berikut:
 $V_{digital} = V_{ukur} / V_{cc} * 255$
 Contoh: misal tegangan analog yang diukur 1 V, maka tegangan digital adalah:
 $V = 1/5 * 255 = 51$ desimal, atau 33H atau 00110011B

DATA HASIL PERCOBAAN

Tabel ADC

No	Tegangan analog (0-5V)	Tampilan LED (bit7-bit0)	Nilai digital LED (decimal)	Nilai digital perhitungan (desimal)
1	0			
2	0.5			
3	1			
4	1.5			
5	2			
6	2.5			
7	3			
8	3.5			
9	4			
10	4.5			
11	5			

ANALISIS DATA

1. Buatlah grafik nilai digital ADC fungsi tegangan analog.
2. Analisis grafik tersebut.