

MIKROKONTROLER ATMEGA BERBASIS CODEVISION AVR (ADC DAN APLIKASI TERMOMETER)

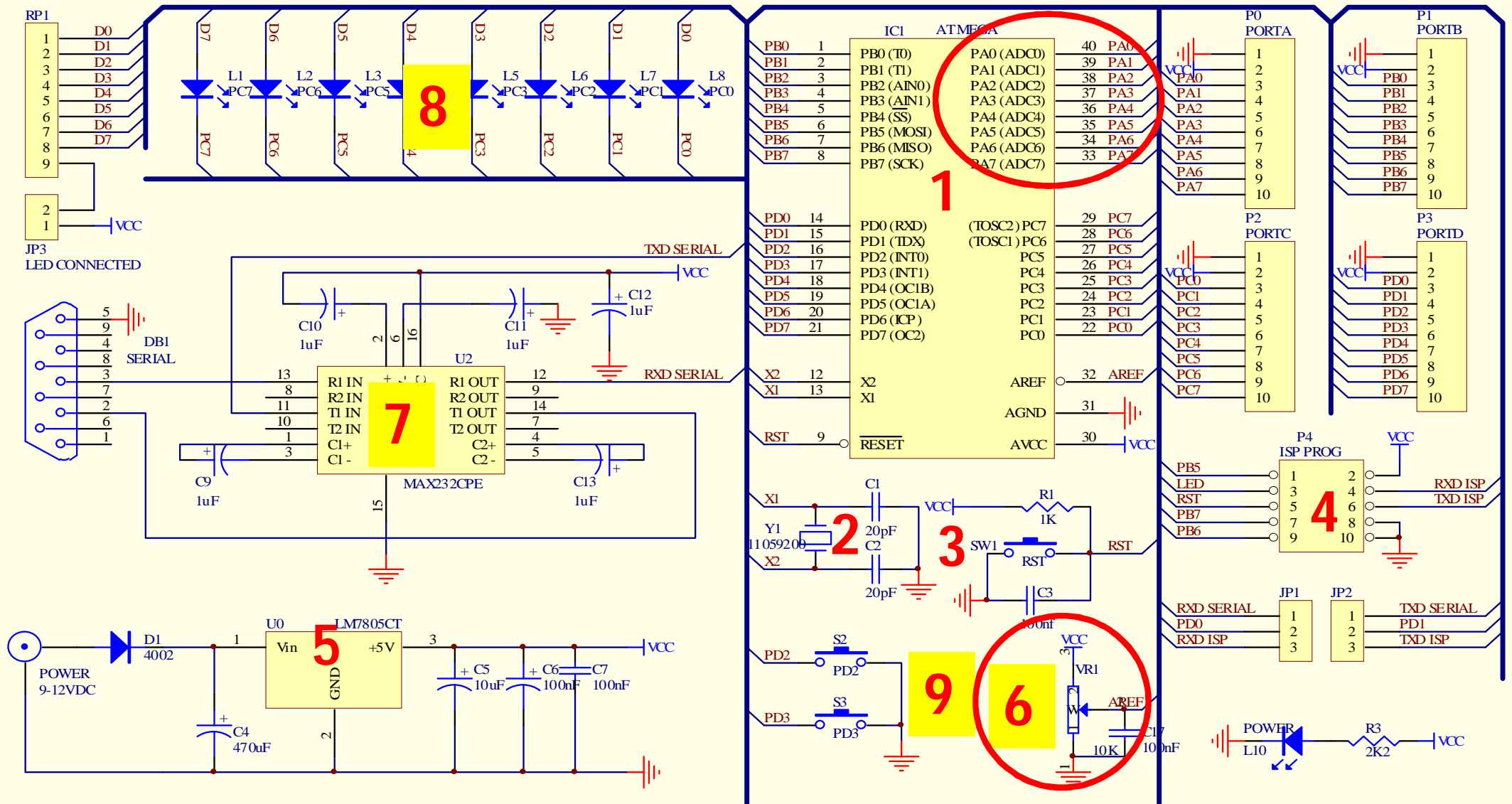
ADC

- Konsep Dasar ADC
 - ▣ ADC = Analog to Digital Converter
 - ▣ Pengubah sinyal analog menjadi sinyal digital (bit)
 - ▣ Rumus dasar ADC:
 - **$\text{Data_ADC} = (\text{Vin}/\text{Vref}) * (\text{Maksimal_Data_Bit})$**
 - Vin = Tegangan analog input yang akan diubah ADC.
 - Vref = Tegangan referensi yang dipakai oleh ADC sebagai acuan pengubah.
 - Maksimal_Data_Bit = Kemampuan maksimal data yang dapat dihasilkan oleh ADC.

Fitur ADC μ C. ATmega16A

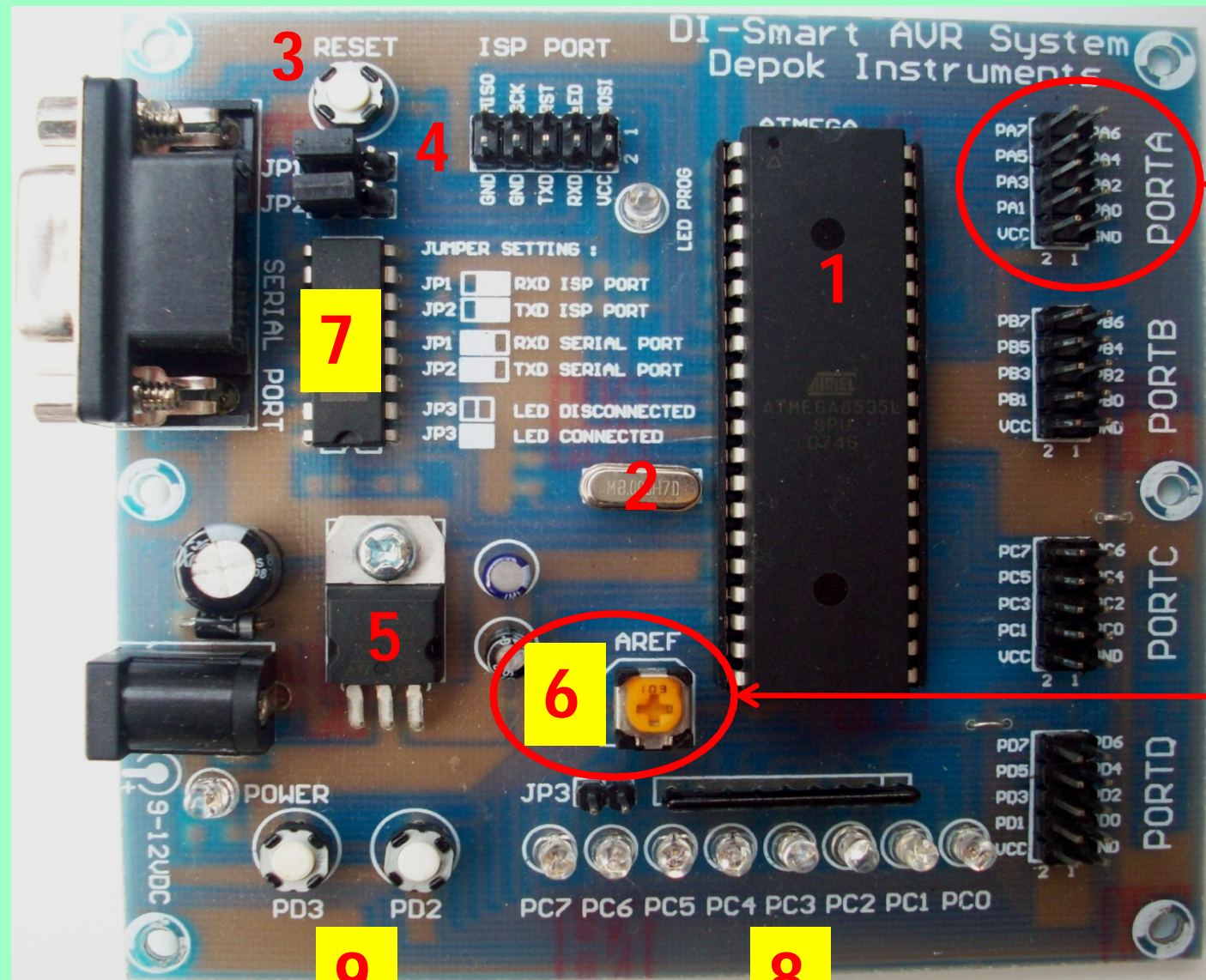
- 8 Kanal ADC:
 - ▣ PORTA.0 = ADC(0); PORTA.1 = ADC(1);
 - ▣ PORTA.2 = ADC(2); PORTA.3 = ADC(3);
 - ▣ PORTA.4 = ADC(4); PORTA.5 = ADC(5);
 - ▣ PORTA.6 = ADC(6); PORTA.7 = ADC(7);
- Maksimal_Data_Bit = 10-bit = 0 – 1023
- Vref ADC:
 - ▣ VCC
 - ▣ $\frac{1}{2}$ VCC
 - ▣ Aref
- Vin ADC = 0V – 5V

DI-Smart AVR.16 System (Sistem Plus μ C. ATmega16A)



DI-Smart AVR.16 System

1. ATmega16A
2. Crystal
3. Reset
4. ISP
5. Power
6. Vref ADC
7. RS232
8. LED Array
9. Push Button



Pemrograman ADC pada CVAVR

- Inisialisasi "*Wizard*"
 - ▣ Pemilihan Vref
 - VCC
 - $\frac{1}{2}$ VCC
 - V@Aref
 - ▣ Pemilihan *Clock*
 - ▣ Aktivasi ADC

Pemrograman ADC Ditampilkan LCD (Setup "Wizard")

1

Chip: ATmega16

Clock: 11.059200 MHz

Check Reset Source

Program Type: Application

2

ADC Enabled Use 8 bits

Interrupt

Volt. Ref: AREF pin

Clock: 172.800 kHz

Auto Trigger Source: Free Running

3

Enable Alphanumeric LCD Support

Characters/Line: 16

Connections

LCD Module AVR

RS	PORTB	Bit: 0
RD	PORTB	Bit: 1
EN	PORTB	Bit: 2
D4	PORTB	Bit: 4
D5	PORTB	Bit: 5
D6	PORTB	Bit: 6
D7	PORTB	Bit: 7

4

File Program Edit Help

Generate program, save and exit

Pemrograman ADC Ditampilkan LCD (Subprogram Hasil "Wizard")

```
D:\PROGRAM C\AVR\PO3_ADC2LCD.c
Notes P03_ADC2LCD.c
24 #include <mega16.h>
25
26 #include <delay.h>
27
28 // Alphanumeric LCD Module functions
29 #include <alcd.h>
30
31 #define ADC_VREF_TYPE 0x00
32
33 // Read the AD conversion result
34 unsigned int read_adc(unsigned char adc_input)
35 {
36     ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
37     // Delay needed for the stabilization of the ADC input voltage
38     delay_us(10);
39     // Start the AD conversion
40     ADCSRA|=0x40;
41     // Wait for the AD conversion to complete
42     while ((ADCSRA & 0x10)==0);
43     ADCSRA|=0x10;
44     return ADCW;
45 }
46
47 // Declare your global variables here
48
49 void main(void)
```

read_adc(channel)
adalah subprogram
untuk membaca
data dari kanal adc.
Dengan channel
adalah 0 – 7.

Pemrograman ADC Ditampilkan LCD (Inisialisasi ADC Hasil "Wizard")

```
D:\PROGRAM C\AVR\P03_ADC2LCD.c
Notes P03_ADC2LCD.c
138 SFIOR=0x00;
139
140 // ADC initialization
141 // ADC Clock frequency: 172.800 kHz
142 // ADC Voltage Reference: AREF pin
143 // ADC Auto Trigger Source: Free Running
144 ADMUX=ADC_VREF_TYPE & 0xff;
145 ADCSRA=0xA6;
146 SFIOR&=0x1F;
147
148 // SPI initialization
149 // SPI disabled
150 SPCR=0x00;
151
152 // TWI initialization
153 // TWI disabled
154 TWCR=0x00;
155
156 // Alphanumeric LCD initialization
157 // Connections specified in the
158 // Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
159 // RS - PORTB Bit 0
160 // RD - PORTB Bit 1
161 // EN - PORTB Bit 2
162 // D4 - PORTB Bit 4
163 // D5 - PORTB Bit 5
```

Inisialisasi ADC sesuai dengan yang diinputkan saat *Wizard*.

Pemrograman ADC Ditampilkan LCD (Membaca dan Menampilkan ADC) [1/2]

```
□ #include <mega16.h>
□ #include <delay.h>
□ #include <stdio.h>
□ #include <alcd.h>
□ #define ADC_VREF_TYPE 0x00
□ unsigned int read_adc(unsigned char adc_input)
□ {
□ ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
□ delay_us(10);
□ ADCSRA|=0x40;
□ while ((ADCSRA & 0x10)==0);
□ ADCSRA|=0x10;
□ return ADCW;
□ }
```

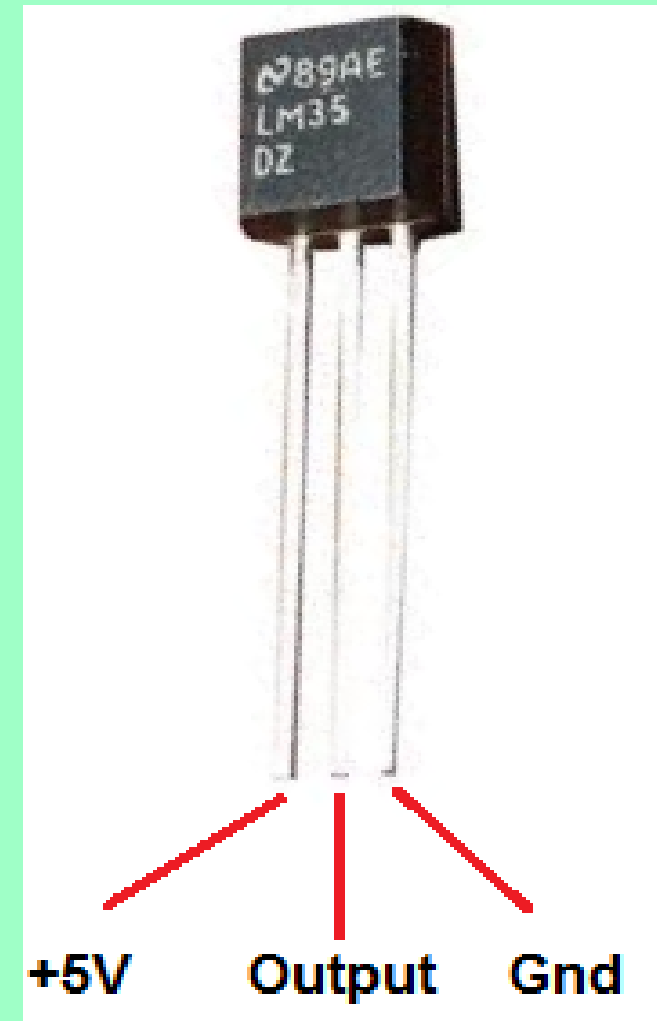
Pemrograman ADC Ditampilkan LCD (Membaca dan Menampilkan ADC) [2/2]

```
□ unsigned char cstr[16];  
□ unsigned int idata;  
□ void main(void)  
    □ {  
        □ ADMUX=ADC_VREF_TYPE & 0xff; ADCSRA=0xA6; SFIOR&=0x1F;  
        □ lcd_init(16);  
        □ while (1)  
            ■ {  
                ■ lcd_clear();  
                ■ idata = read_adc(0);  
                ■ sprintf(cstr,"Data ADC: %04d",idata);  
                ■ lcd_puts(cstr);  
                ■ delay_ms(250);  
            ■ }  
    □ }
```

Aplikasi ADC

“Termometer berbasis Sensor LM35DZ”

- LM35DZ adalah sensor temperatur analog ekonomis namun cukup baik dalam mengukur temperatur.
 - Jangkauan: 0 – 100 °C
 - Sensitivitas: 10 mV/ °C
 - Operasional: 4 – 30 V



Fungsi Transfer Sensor LM35DZ

- Tinjau-ulang dasar ADC:
 - ▣ **$\text{Data_ADC} = (\text{Vin}/\text{Vref}) * (\text{Maksimal_Data_Bit})$**
 - Data_ADC = Hasil konversi ADC internal ATmega16A
 - Vin = Tegangan LM35
 - Vref = Sesuai pengaturan di "Wizard" CVAVR = 5V
 - Maksimal_Data_Bit = 10 Bit = 1023
- Sensitivitas LM35 = 10 mV/ °C
- Temperatur terukur (T):
 - ▣ $T = (\text{Vin}/10 \text{ mV}) \text{ } ^\circ\text{C}$
 - ▣ **$T = ((\text{Data_ADC} * 5\text{V}/1023)/10 \text{ mV}) \text{ } ^\circ\text{C}$**

“Termometer berbasis Sensor LM35DZ” (Skrip Program CVAVR) [1/3]

```
□ #include <mega16.h>
□ #include <delay.h>
□ #include <stdio.h>
□ #include <alcd.h>
□ #define ADC_VREF_TYPE 0x00
□ unsigned int read_adc(unsigned char adc_input)
□ {
□ ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
□ delay_us(10);
□ ADCSRA|=0x40;
□ while ((ADCSRA & 0x10)==0);
□ ADCSRA|=0x10;
□ return ADCW;
□ }
```

“Termometer berbasis Sensor LM35DZ” (Skrip Program CVAVR) [2/3]

- `unsigned char cstr[16], T; //T = Temperatur`
- `unsigned int idata;`
- `float fdata; //Nilai pecahan agar perhitungan baik.`
- `void main(void)`
 - `{`
 - `ADMUX=ADC_VREF_TYPE & 0xff; ADCSRA=0xA6; SFIOR&=0x1F;`
 - `lcd_init(16);`
 - `while (1)`
 - `{`
 - `lcd_clear();`
 - `idata = read_adc(0);`
 - `sprintf(cstr,"Data ADC: %04d",idata);`
 - `lcd_puts(cstr);`

“Termometer berbasis Sensor LM35DZ” (Skrip Program CVAVR) [3/3]

```
■ fdata = (((float)idata*5)/1023)/0.01;  
■ T = (char)fdata;  
■ sprintf(cstr,"Temperatur: %02d C",T);  
■ lcd_gotoxy(0,1);  
■ lcd_puts(cstr);  
■ delay_ms(250);  
■ }  
■ }
```


E-BOOK DINS

- <http://depokinstruments.com/category/3-e-book/>