

DT-AVR **DT-AVR** *Application Note* AN144 – Konversi Data Digital ke Analog

Oleh: Tim IE

Pengolahan data dalam sebuah mikroprosesor / mikrokontroler dilakukan dalam bentuk digital, namun dalam beberapa kasus kita membutuhkan keluaran berupa data analog. Aplikasi kali ini akan memberikan sebuah contoh penggunaan modul DT-I/O DAC-08 untuk melakukan konversi data digital menjadi data analog. Modul kontroler yang digunakan yaitu DT-AVR Low Cost Micro System yang berbasis pada mikrokontroler keluarga AVR® ATMEGA8535. Konversi data digital ke analog seringkali digunakan dalam aplikasi-aplikasi praktis, seperti mengatur intensitas cahaya lampu, mengontrol kecepatan motor, MP3 *player*, dsb.

Komponen yang diperlukan dalam aplikasi ini adalah:

- 1x DT-AVR Low Cost Micro System
- 1x DT-I/O DAC-08

Adapun blok diagram sistem secara keseluruhan adalah sebagai berikut:



Gambar 1
Blok Diagram AN144

Hubungan antara modul-modul tersebut adalah sebagai berikut:

DT-AVR Low Cost Micro System		DT-I/O DAC-08	
PORT A.0 (J10 - pin3)		AOUT (J3)	
PORT B (J11)		DAC DATA (J1)	
Pin	Nama	Pin	Nama
1	GND	1	GND
2	VCC	2	VCC
3	PORTB.0	3	D0
4	PORTB.1	4	D1
5	PORTB.2	5	D2
6	PORTB.3	6	D3
7	PORTB.4	7	D4
8	PORTB.5	8	D5
9	PORTB.6	9	D6
10	PORTB.7	10	D7

Tabel 1
Hubungan DT-AVR Low Cost Micro System Dengan DT-I/O DAC-08

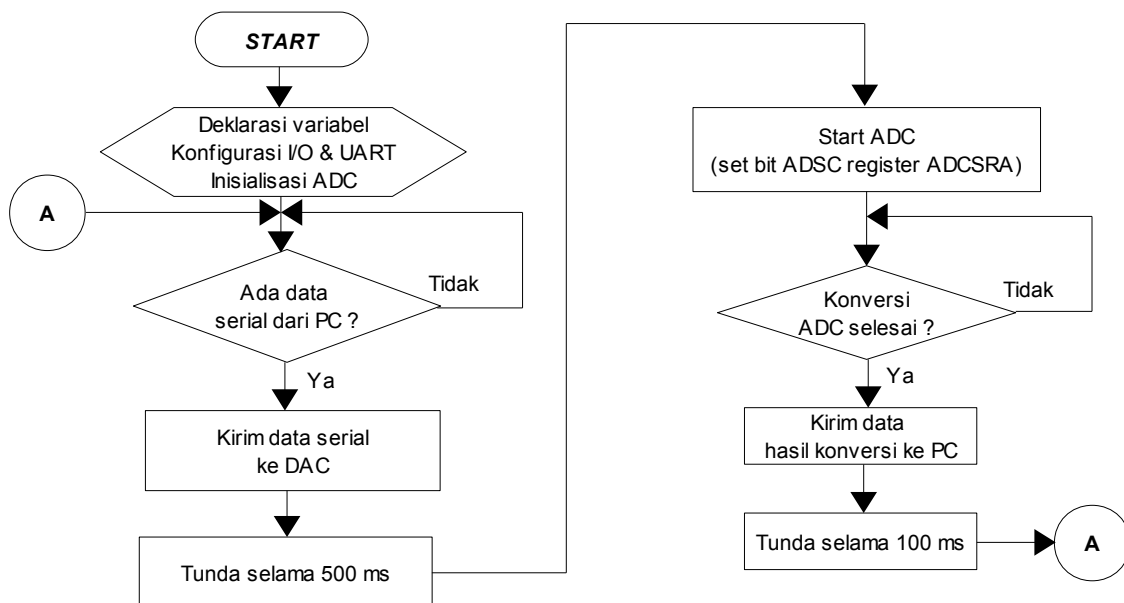
DT-AVR Low Cost Micro System		DT-I/O DAC-08	
PORT C (J12)		DAC CONTROL (J2)	
Pin	Nama	Pin	Nama
1	GND	1	GND
2	VCC	2	VCC
5	PORTC.2	5	A/B
6	PORTC.3	6	LDACpin
7	PORTC.4	7	WR
8	PORTC.5	8	PDpin
9	PORTC.6	9	CLRpin
10	PORTC.7	10	CSpin

Tabel 2
Hubungan DT-AVR Low Cost Micro System Dengan DT-I/O DAC-08 (Lanjutan Tabel 1)

Input ADC0 (Port A.0) pada DT-AVR Low Cost Micro System digunakan untuk membaca *output* analog dari DT-I/O DAC-08 (AOUT). Port A.0 ini dapat diganti dengan jalur port A yang lain, namun kode program juga harus disesuaikan. Port B berisi data digital yang akan dikonversi ke bentuk analog oleh DT-I/O DAC-08. Port C digunakan untuk mengontrol DT-I/O DAC-08 seperti *read/write, clear*, dll.

Aturlah *jumper* pada DT-I/O DAC-08 agar semua fungsi dapat digunakan, yaitu: LDAC (J4) di posisi 2-3 (semua mode dapat digunakan), CLR (J5) di posisi 1-2 (dapat menggunakan rutin CLEAR), CS (J7) di posisi 2-3 (modul DAC dapat dipilih oleh program), PD (J8) di posisi 1-2 (dapat menggunakan rutin POWER DOWN/UP). Selain itu, aturlah *jumper* J4 & J5 DT-AVR Low Cost Micro System pada posisi 1-2 agar komunikasi serial UART RS-232 dapat digunakan. Hubungkan DT-AVR Low Cost Micro System ke *COM port* komputer menggunakan kabel serial. Setelah semua rangkaian dan sumber tegangan terhubung dengan tepat, maka programlah **Ending.HEX** ke DT-AVR Low Cost Micro System menggunakan DT-HiQ AVR USB ISP atau *in-system programmer* lain yang kompatibel.

Flowchart program utama untuk sistem ini adalah sebagai berikut:



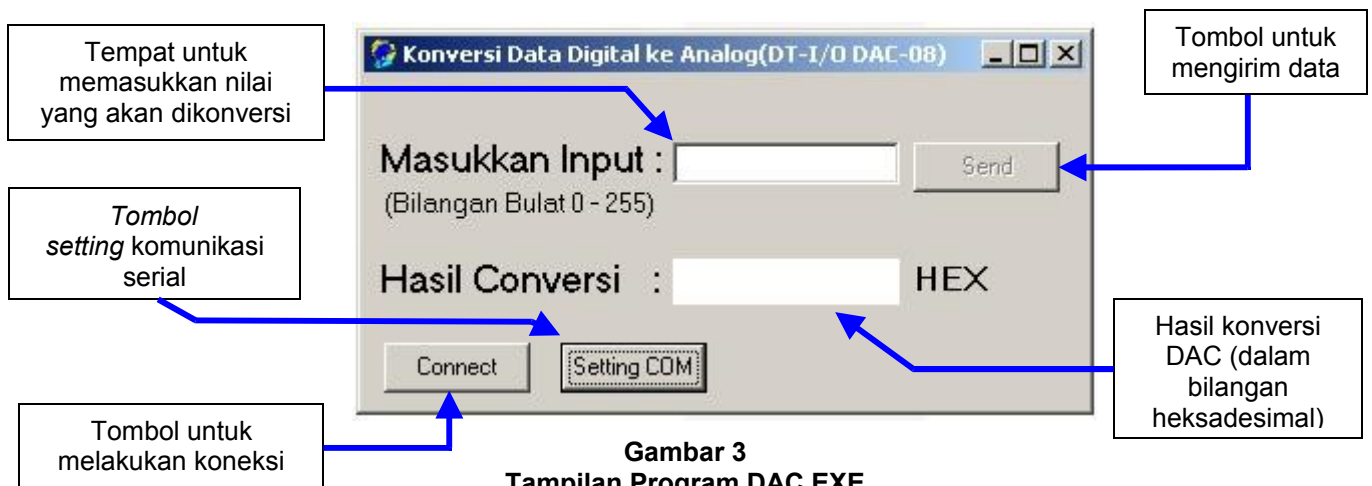
Gambar 2
Flowchart Program Utama

Alur data dalam sistem ini dimulai dari komputer yang dikirimkan secara serial ke DT-AVR Low Cost Micro System. DT-AVR Low Cost Micro System meneruskan data digital tersebut ke modul DT-I/O DAC-08 untuk dikonversi menjadi data analog (tegangan). Data analog keluaran modul DT-I/O DAC-08 diumpankan kembali ke ADC internal

ATMEGA8535 (DT-AVR Low Cost Micro System) untuk dikonversi lagi menjadi data digital dan dikirim ke komputer secara serial melalui COM port.

Program utama dalam sistem ini akan diproses sebagai berikut:

1. Proses yang pertama dilakukan adalah deklarasi variabel *DataSerialTemp* dan *adc_data*.
 - *DataSerialTemp* digunakan untuk menampung data digital yang diterima dari PC.
 - *adc_data* digunakan untuk menampung data digital hasil konversi ADC.
2. Program akan menentukan fungsi masing-masing jalur I/O ATMEGA8535 (sebagai *input/output*). Program juga melakukan konfigurasi port UART agar DT-AVR Low Cost Micro System dapat berkomunikasi secara serial dengan PC. Beberapa *register* yang berkaitan dengan ADC internal ATMEGA8535 juga diatur agar ADC menggunakan referensi AVCC, memilih jalur *input* ADC0 (PORTA.0). ADC ATMEGA8535 akan memulai proses konversi data setelah bit ADSC pada register ADCSRA diisi dengan nilai 1.
3. Setelah proses konfigurasi selesai, berikutnya program akan menunggu datangnya data serial pada UDR. Pengiriman data secara serial dari komputer dapat dilakukan dengan menggunakan program semacam Terminal[®] yang dapat menampilkan data biner (bukan kode ASCII). HyperTerminal[®] juga dapat digunakan tetapi akan sedikit lebih sulit. Pengaturan komunikasi serial sebagai berikut: *baudrate* 9600 bps, 8 bit data, tanpa bit *parity*, 1 bit stop, dan tanpa *flow control*. Dalam aplikasi ini juga disediakan program yang dibuat dengan Borland[®] Delphi 7.0[®] untuk mengirim / menerima data dari/ke DT-AVR Low Cost Micro System.



4. Data dari PC yang diterima oleh mikrokontroler akan disimpan dalam variabel *DataSerialTemp*. Data ini didapatkan dengan membaca *register* UDR pada mikrokontroler ATMEGA8535. Setelah itu, data ini dikirim ke DT-I/O DAC-08 untuk kemudian diubah menjadi besaran analog.
5. Setelah proses konversi DAC selesai, *output* tegangan analog dari DAC dibaca kembali oleh ATMEGA8535 melalui Port A.0 (ADC0). Program akan memerintah ADC internal ATMEGA8535 agar memulai proses konversi.
6. Setelah proses konversi ADC selesai, maka data hasil konversi tersebut dikirimkan kembali ke PC untuk kemudian ditampilkan pada program DAC.EXE seperti gambar 3.

Listing program terdapat pada **AN144.zip**.

Selamat berinovasi !

CodeVisionAVR is copyright by Pavel Haiduc, HP InfoTech s.r.l.
 Borland is registered trademark of Borland Software Corporation.
 Borland Delphi 7.0 is copyright by Borland Software Corporation.
 Cport Componen is created by Dejan Crnila.
 HyperTerminal is a copyright by Hilgraev Inc.
 Terminal is a copyright by Bray.