

DESAIN DAN IMPLEMENTASI KONTROLER PID UNTUK PENGATURAN KECEPATAN MOTOR DC BERBASIS MIKROKONTROLER ATMEGA8535

1. TUJUAN PERCOBAAN

Percobaan ini bertujuan untuk :

- Melakukan identifikasi plant motor DC
- Mendesain dan mengimplementasikan kontroler PID untuk mengatur kecepatan putar motor DC dengan mikrokontroler Atmega8535

2. ALAT-ALAT YANG DIBUTUHKAN

Berikut beberapa alat yang harus dipersiapkan:

- Motor DC CM150F
- Tachometer GT 150X
- Multimeter
- Power supply PS150 E
- Modul Mikrokontroler Atmega 8535 + downloader
- Signal Conditioning
- Modul DAC 8 bit
- Komputer
- Beberapa kabel penghubung

3. DASAR TEORI

3.1. System pengaturan loop terbuka

Sistem pengaturan loop terbuka merupakan suatu sistem pengaturan yang keluarannya tidak mempunyai pengaruh terhadap aksi kontrol. Pada sistem pengaturan loop terbuka tidak terdapat jaringan umpan balik. Oleh karena itu sistem pengaturan loop terbuka hanya dapat digunakan jika hubungan antara masukan dan keluaran sistem diketahui dan tidak terdapat gangguan internal maupun eksternal.

3.2. Sistem pengaturan loop tertutup

Sistem pengaturan loop tertutup merupakan sistem pengaturan dimana sinyal keluaran mempunyai pengaruh langsung terhadap sinyal kontrol (aksi kontrol). Pada sistem pengaturan

loop tertutup terdapat jaringan umpanbalik (feedback) karenanya sistem pengaturan loop tertutup seringkali disebut sebagai sistem pengaturan umpanbalik

3.3. Pengenalan mikro kontroler

Mikrokontroler adalah suatu alat elektronika digital yang mempunyai masukan dan keluaran serta kendali dengan program yang bisa ditulis dan dihapus dengan cara khusus, cara kerja mikrokontroler sebenarnya membaca dan menulis data. Mikrokontroler merupakan komputer didalam chip yang digunakan untuk mengontrol peralatan elektronik, yang menekankan efisiensi.

3.3.1. Subsistem Penyusun Mikrokontroller

3.3.1.1. Bagian Masukan/Keluaran (I/O)

Sistem dengan mikrokontroler umumnya menggunakan piranti input yang jauh lebih kecil seperti saklar atau keypad kecil. Hampir semua input mikrokontroler hanya dapat memproses sinyal input digital dengan tegangan yang sama dengan tegangan logika dari sumber. Level nol disebut dengan VSS dan tegangan positif sumber (VDD) umumnya adalah 5 volt. Beberapa mikrokontroler dilengkapi dengan ADC dalam satu rangkaian terpadu.

Sistem mikrokontroler mempunyai output yang jauh lebih sederhana seperti lampu indikator atau beeper. Frasa kontroler dari kata mikrokontroler memberikan penegasan bahwa alat ini mengontrol sesuatu. Mikrokontroler atau komputer mengolah sinyal secara digital, sehingga untuk dapat memberikan output analog diperlukan proses konversi dari sinyal digital menjadi analog. Piranti yang dapat melakukan konversi ini disebut dengan DAC (Digital to Analog Converter).

3.3.1.2. CPU (Central Processing Unit)

CPU adalah otak dari sistem mikrokontroler. Pekerjaan utama dari CPU adalah mengerjakan program yang terdiri atas instruksi-instruksi yang diprogram oleh programmer. Suatu program mikrokontroler akan menginstruksikan CPU untuk membaca informasi dari piranti input, membaca informasi dari dan menulis informasi ke memori, dan untuk menulis informasi ke output.

3.3.1.3. Clock dan Memori Komputer

Sistem mikrokontroler menggunakan osilator clock untuk memicu CPU mengerjakan satu instruksi ke instruksi berikutnya dalam alur yang berurutan. Setiap langkah kecil dari operasi mikrokontroler memakan waktu satu atau beberapa clock untuk melakukannya. Ada beberapa macam tipe dari memori mikrokontroler yang digunakan untuk beberapa tujuan yang berbeda dalam sistem mikrokontroler. Tipe dasar yang sering ditemui dalam mikrokontroler adalah ROM (Read Only Memory) dan RAM (Random Access Memory). ROM digunakan sebagai media penyimpanan program dan data permanen yang tidak boleh berubah meskipun tidak ada tegangan yang diberikan pada mikrokontroler. RAM digunakan sebagai tempat penyimpanan data sementara dan hasil kalkulasi selama proses operasi. Beberapa mikrokontroler mengikutsertakan tipe lain dari memori seperti EPROM (Erasable Programmable Read Only Memory) dan EEPROM (Electrically Erasable Programmable Read Only Memory).

3.3.1.4. Program Komputer

Komponen utama dari program adalah instruksi-instruksi dari instruksi set CPU. Program disimpan dalam memori dalam sistem mikrokontroler dimana mereka dapat secara berurutan dikerjakan oleh CPU.

3.3.1.5. Perangkat Lunak

Pemrograman mikrokontroler dapat dilakukan dengan menggunakan bahasa C. Dengan bahasa C, pemrograman mikrokontroler menjadi lebih mudah, hal ini karena dengan format bahasa C akan secara otomatis diubah menjadi bahasa assembler dengan format file hexa. Perangkat lunak pada mikrokontroler dapat dibagi menjadi lima kelompok sebagai berikut :

1. Instruksi Transfer Data

Instruksi ini berfungsi memindahkan data, yaitu antar register, dari memori ke memori, dari register ke memori dan lain lain

2. Instruksi Aritmatika

Instruksi ini melaksanakan operasi aritmatika yang meliputi penjumlahan, pengurangan, penambahan satu (*increment*), pengurangan satu (*decrement*), perkalian dan pembagian.

3. Instruksi Logika dan Manipulasi Bit

Berfungsi melaksanakan operasi logika AND, OR, XOR, perbandingan, penggeseran dan komplemen data.

4. Instruksi Percabangan

Berfungsi untuk mengubah urutan normal pelaksanaan suatu program. Dengan instruksi ini, program yang sedang dilaksanakan akan meloncat ke suatu alamat tertentu.

5. Instruksi Stack, I/O, dan Kontrol

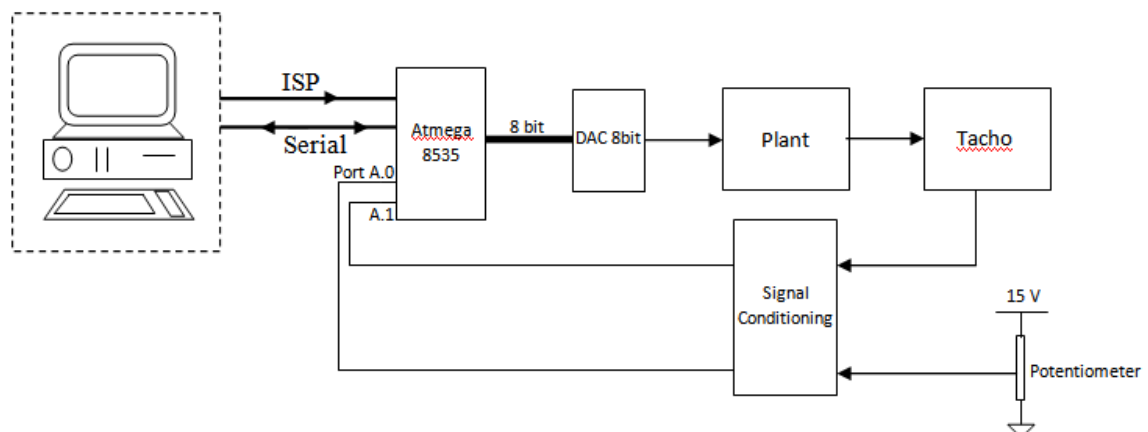
Instruksi ini mengatur penggunaan stack, membaca/menulis port I/O, serta pengontrolan.

4. PROSEDUR PERCOBAAN

4.1 Identifikasi Sistem

Langkah Percobaan

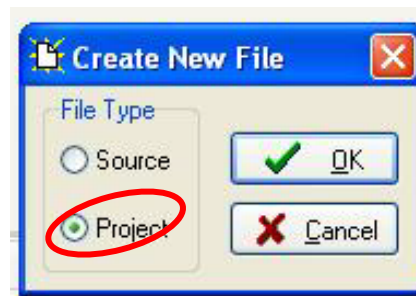
1. Rangkai peralatan sesuai dengan blok diagram seperti di bawah :



Gambar 1. Konfigurasi Sistem

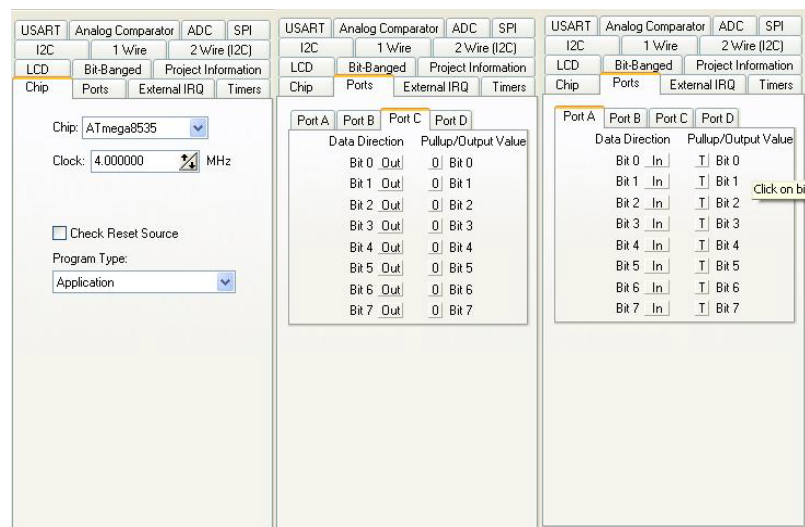
2. Buka Program CodeVisionAVR.

3. Buka **file** → **new** → pilih “**project**”

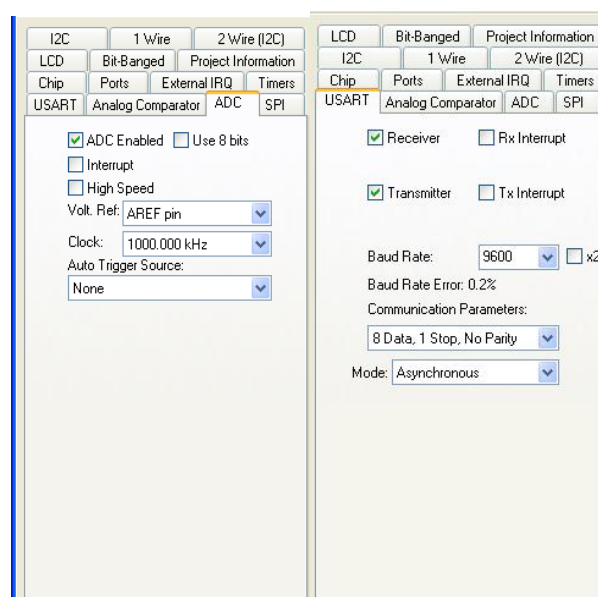


Gambar 2. Jendea **Create New File**

4. Klik **Yes** dan simpan semua tipe *file* dengan **nama** yang sama
5. Pada menu **Codewizard AVR** atur pilihan sesuai dengan gambar, kemudian klik **File** → **Generate, Save and Exit**



(a)



(b)



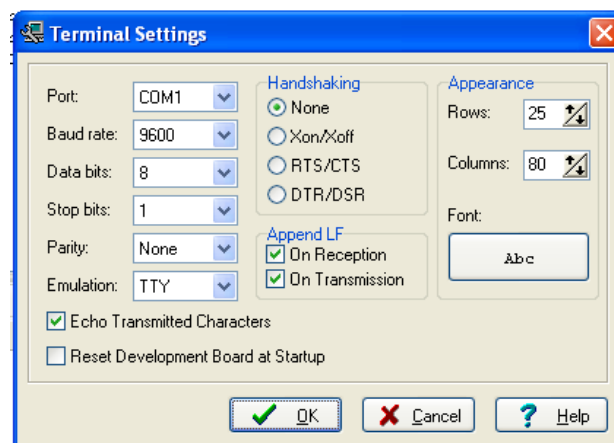
(c)

Gambar 3. (a) *tab* chip, port A dan port C (b) *tab* ADC, USART (c) menu file

6. Pada layar notes masukkan program dibawah ini.

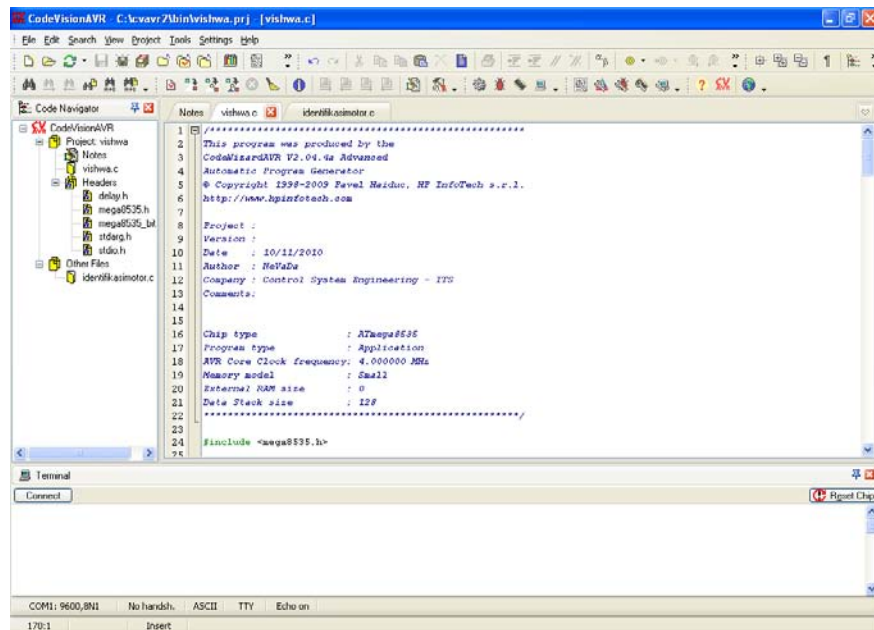
```
// Declare your global variables here
unsigned int data_tacho,data_sp;
.
.
.
.
.
.
printf("Hasil identifikasi open loop:");
printf("\n\r");
printf("Input \t Output");
printf("\r");
while (1)
{
// Place your code here
data_sp=read_adc(0);
data_tacho=1.5*read_adc(1);
printf("%i \t %i",data_sp,data_tacho);
printf("\r");
PORTC=data_sp;
};
}
```

7. Pilih **Settings** → **Terminal** pastikan pilihannya sesuai dengan port computer yang digunakan



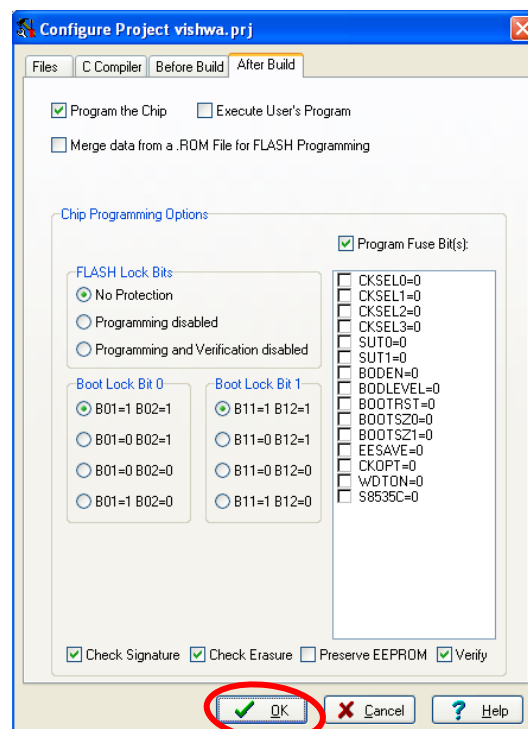
Gambar 4. Jendela **Terminal Settings**

8. Pilih Tools → Terminal → klik Connect



Gambar 5. Jendela CodeVision AVR dengan tampilan Terminal

9. Pilih Project → Configure → After Build → klik OK



Gambar 6. Jendela CodeVision AVR dengan tampilan Terminal

10. Pilih Project → Build All → klik “Program the Chip” pastikan tidak ada *error*