

BAB IV

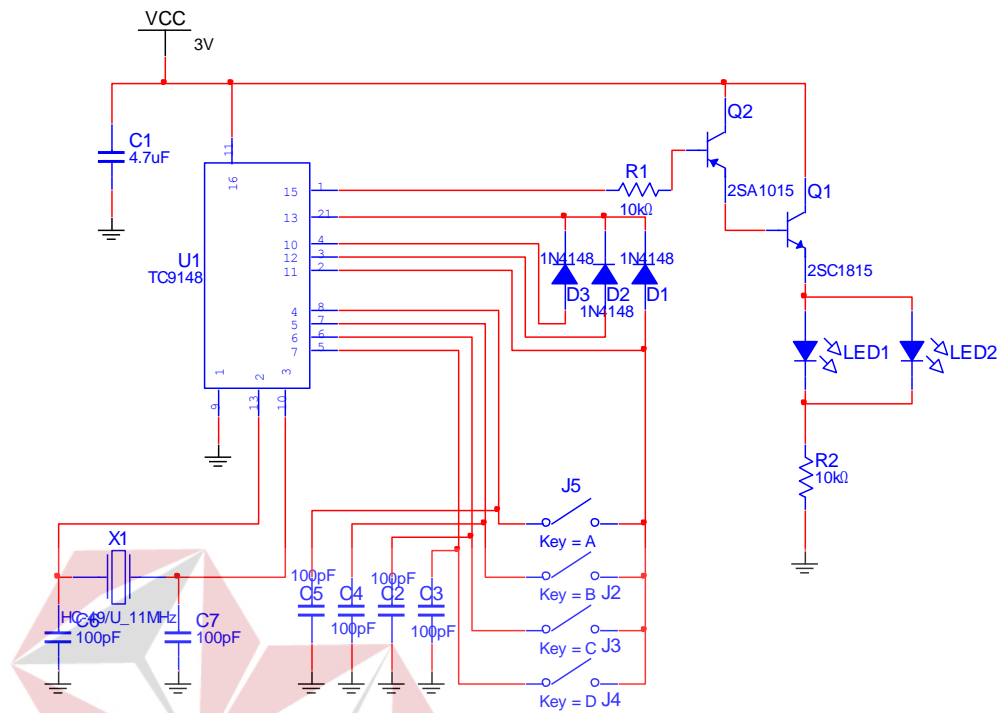
HASIL KERJA PRAKTEK

4.1 Merancang Rangkaian Remote Control

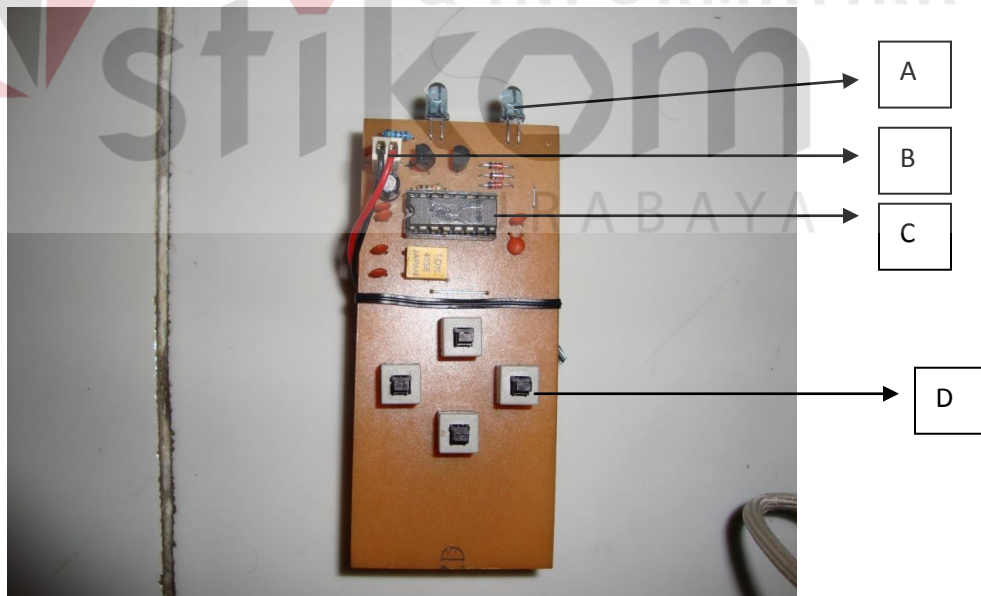
Sesuai namanya remote control adalah alat pengendali jarak jauh yang berfungsi untuk mengendalikan sebuah benda (biasanya memiliki komponen elektronik). Benda yang dikendalikan tersebut kemudian akan memberikan respon sesuai jenis instruksi yang diberikannya. Instruksi diberikan dengan cara menekan tombol yang sesuai pada remote control. Sejarah mencatat bahwa pada masa awal pengembangannya penerapan remote control sempat digunakan oleh pasukan Jerman untuk menggerakkan kapal-kapal lautnya dari jarak jauh untuk ditabrakan ke kapal perang pasukan sekutu pada Perang Dunia I. Saat ini remote control digunakan untuk berbagai keperluan dari untuk mengubah temperatur AC hingga mengatur gerak robot.

4.1.1 Transmitter

Transmitter ini digunakan sebagai remote untuk melakukan panggilan perintah untuk menjalankan motor DC yang berfungsi untuk membuka dan menutup pintu pagar. Rencananya, transmitter akan dibuat dengan menyerupai remote control pada umumnya. Disini pemancar remote control menggunakan pemancar dari LED. LED ini didesain dengan operasi pada panjang gelombang 950nm pada tegangan maximum. Pada perancangan ini menggunakan IC C-MOS TC9148 yang dikembangkan pada *remote control transmitter* infra merah. IC ini memiliki 18 fungsi dan total 75 perintah yang dapat dikirimkan. 63 perintah dengan *continous key* dan 12 perintah *single shot*.



Gambar 6.1 Skematik IC TC148



Gambar 6.2 Remote Control Setelah Jadi

Penjelasan Remot Control Transmitter :

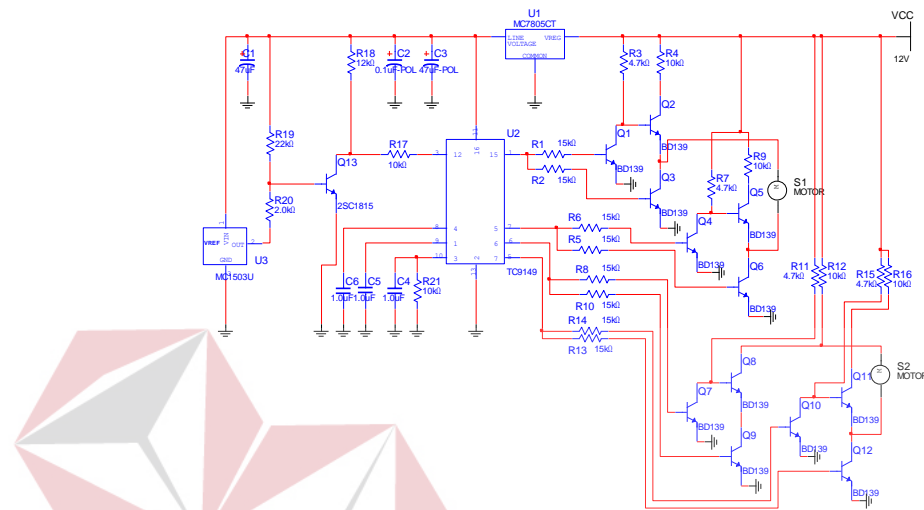
1. A = Infra Merah
2. B = Power Supply 3V
3. C = IC TC9148 (Transmitter)
4. D = Tombol 4 Arah

4.1.2 Receiver

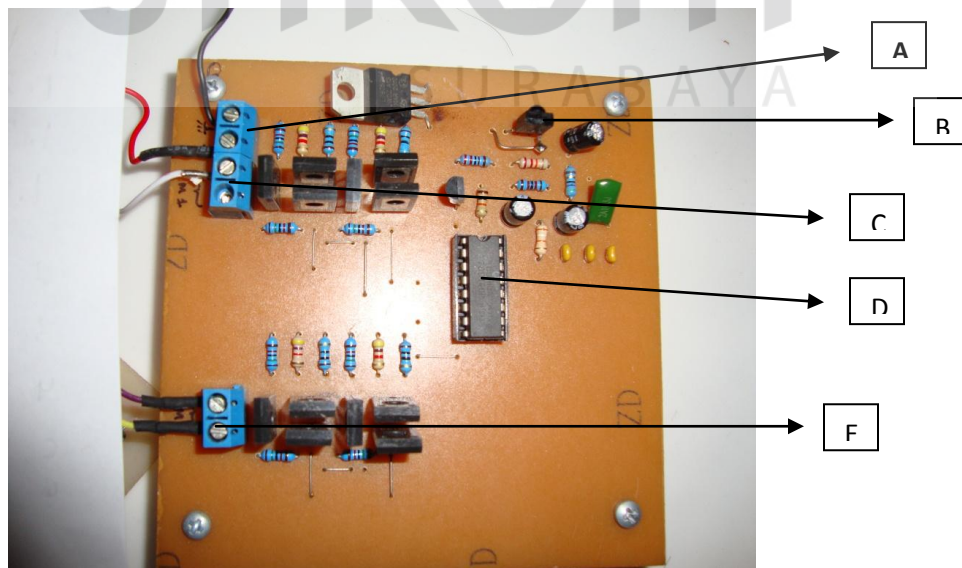
Pada bagian penerima, untuk menerima sinyal inframerah digunakan phototransistor tipe TSOP34838 phototransistor ini dioperasikan pada panjang gelombang terima pada 950nm yang cocok dengan pemancar TLN105. Pada *receiver* menggunakan IC TC9149 yang merupakan pasangan dari TC9148. IC ini memiliki 18 buah output yang berbeda untuk setiap fungsi yang dipancarkan dari *transmitter* yang menggunakan IC TC9148.



Gambar 6.3 Phototransistor



Gambar 6.4 Skematik Receiver IC TC9149



Gambar 6.5 Receiver Setelah Jadi

Penjelasan Remote Control Receiver :

A = Power Supply 5V – 24V

B = Phototransistor tipe TSOP34838

C = Pengendali Remote Tombol Atas dan Bawah

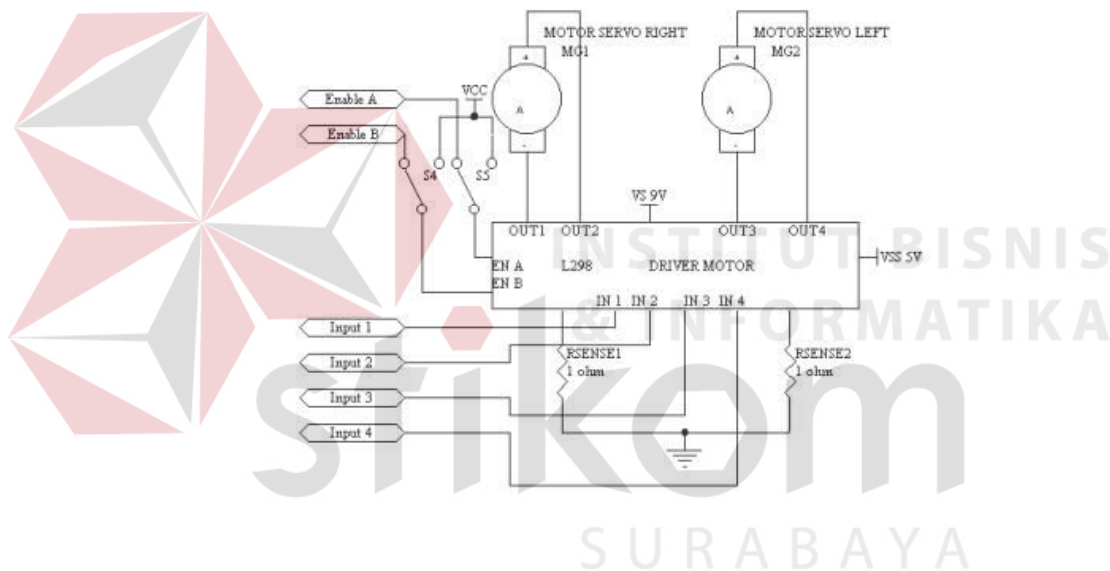
D = IC TC9149

E = Pengendali Remote Kanan dan Kiri

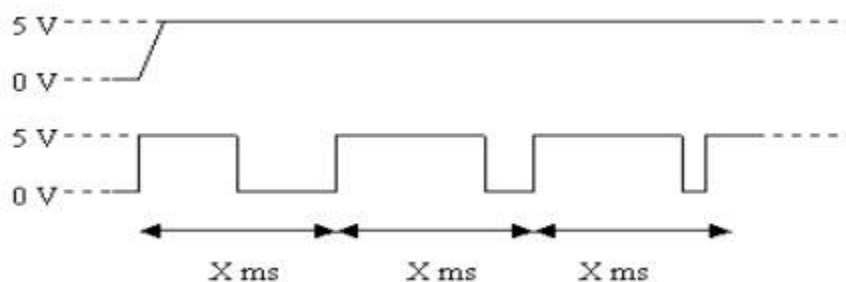
4.2 Merancang Driver Motor DC

Ada beberapa macam driver motor DC yang biasa kita pakai seperti menggunakan relay yang diaktifkan dengan transistor sebagai saklar, namun yang demikian dianggap tidak efisien dan terlalu ribet "repot" dalam pengerjaan hardware-nya. Dengan berkembangnya dunia IC, sekarang sudah ada H Bridge yang dikemas dalam satu IC dimana memudahkan kita dalam pelaksanaan hardware dan kendalinya apalagi jika menggunakan mikrokontroler, saya rasa akan lebih mudah lagi penggunaannya. IC yang familiar seperti IC L298 dan L293, kedua IC ini memiliki kelebihan dan kekurangan masing-masing. Untuk lebih jelasnya Anda bisa lihat datasheet dari masing-masing IC tersebut.

Modul yang menggunakan IC driver L298 yang memiliki kemampuan menggerakkan motor DC sampai arus 4A dan tegangan maksimum 46 VoltDC untuk satu kanalnya. Rangkaian driver motor DC dengan IC L298 diperlihatkan pada gambar 6.6. Pin *Enable A* dan *B* untuk mengendalikan jalan atau kecepatan motor, pin Input 1 sampai 4 untuk mengendalikan arah putaran. Pin *Enable* diberi VCC 5 Volt untuk kecepatan penuh dan PWM (*Pulse Width Modulation*) untuk kecepatan rotasi yang bervariasi tergantung dari level highnya. Ilustrasinya ditunjukkan pada gambar 2.



Gambar 7.1 Rangkaian Driver motor DC dengan L298



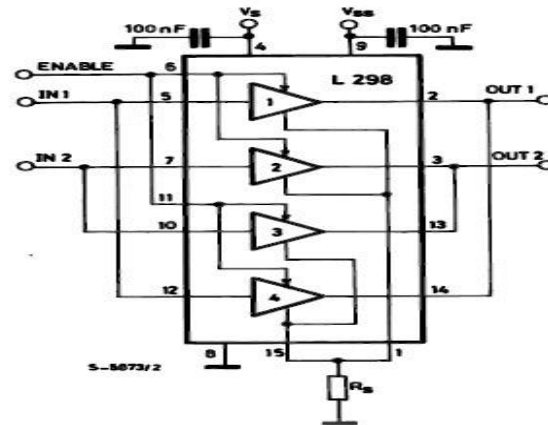
Gambar 7.2 ilustrasi Timing enable pada IC

Rangkaian ini sangat berguna dan efisien sekali jika kita ingin membangun suatu mobile robot seperti yang dipergunakan dalam kontes robot indonesia (KRI) dan kontes robot cerdas indonesia (KRCI), paling tidak rangkaian ini sudah penulis buktikan pada tugas kuliah penulis dan berhasil dengan baik.

IC H Bridge driver motor DC L298 memiliki dua buah rangkaian H-Bridge di dalamnya, sehingga dapat digunakan untuk men-drive dua buah motor DC. H Bridge driver motor DC L298 masing – masing dapat mengantarkan arus 2A. Namun, dalam penggunaannya, H Bridge driver motor DC L298 dapat digunakan secara paralel. Sehingga kemampuan menghantarkan dari H Bridge driver motor DC L298 arusnya menjadi 4A. Konsekuensi dari pemasangan H Bridge driver motor DC L298 dengan mode paralel maka, penulis perlu 2 buah H bridge driver motor DC L298 untuk mengendalikan 2 motor DC menggunakan H Bridge driver motor DC pada mode paralel.

Dengan ini pin dari IC H Bridge motor DC l298 yang terhubung dalam modus operasi paralel adalah sebagai berikut :

1. OUT1 dihubungkan dengan OUT4
2. OUT2 dihubungkan dengan OUT#
3. IN1 dihubungkan dengan IN4



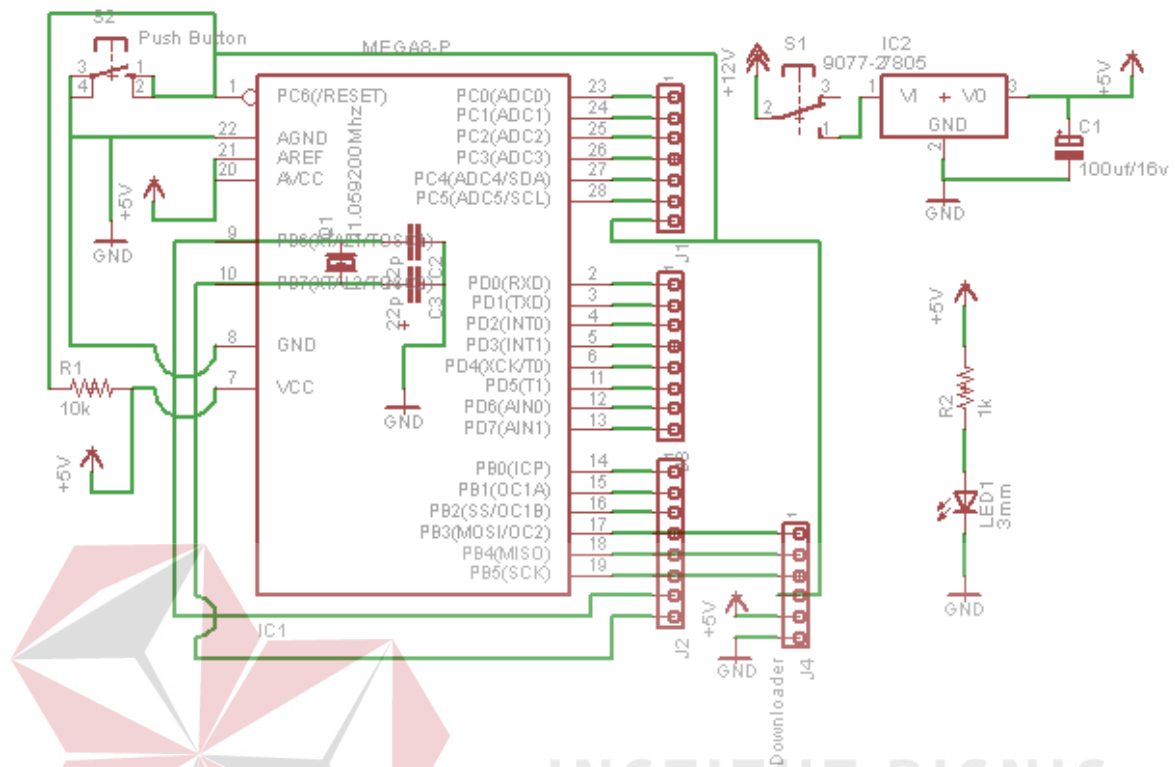
Gambar 7.3 Mode Pararel H Bridge Driver Motor DC L298

4.3 Minimum Sistem ATmega8

Rangkaian minimum sistem adalah rangkaian minimal dimana chip microcontroller dapat berkerja (running). Chip AVR ATmega dilengkapi dengan osilator internal sehingga, untuk menghemat biaya, tidak perlu menggunakan Kristal / resonator eksternal untuk sumber clock CPU.

Sistem minimum AVR sangat sederhana dimana hanya menghubungkan VCC dan AVCC ke +5V dan GND dan AGND ke ground tanpa memakai kristal, dan pin reset diambangkan (tidak dihubungkan apa-apa) chip sudah siap berkerja normal.

Chip akan reset jika tegangan catu nol atau pin RST dipaksa 0. Jika membutuhkan tombol reset, dapat ditambahkan dengan rangkaian reset. Tegangan kerja chip tipe L dapat beroperasi 2.7V – 5.5V.



Gambar 8.1 Skematik Minimum Sistem ATmega8

4.4 Pembuatan Program

Program komputer (juga disebut sebagai software atau hanya program) adalah serangkaian instruksi berurutan yang ditulis untuk melakukan serangkaian tugas untuk komputer. Komputer tak memiliki kemampuan untuk menyelesaikan masalahnya sendiri. Komputer hanyalah berupa alat yang digunakan untuk melakukan perhitungan-perhitungan dan serangkaian tugas yang dibebankan kepadanya. Pembuat program disebut sebagai programmer.

secara umum, program memberikan kemampuan komputer untuk melakukan fungsi khusus. Komputer akan mengeksekusi atau menjalankan instruksi yang ada dalam program di dalam central processor. Program dibuat dengan menggunakan **bahasa**

pemrograman sebagai alat untuk mengungkapkan ide sang programmer agar dapat dijalankan oleh komputer. Setelah ditulis dalam bahasa pemrograman, tidak serta merta apa yang kita tulis tersebut dapat dilaksanakan oleh komputer. Komputer hanya mengerti bilangan biner dan instruksi-instruksi menggunakan bahasa mesin. Untuk membuat komputer mengerti maksud dan tujuan kita, diperlukanlah sebuah kompiler yang dapat menerjemahkan bahasa pemrograman ke bahasa mesin. Namun ada juga komputer program yang tidak berupa kode bahasa mesin, namun berupa kode-kode khusus dan terkadang juga berupa kode-kode bahasa pemrograman. Program model ini memerlukan bantuan interpreter untuk menerjemahkan bahasa tersebut ke bahasa mesin sehingga dimengerti oleh perangkat keras.

Source code ditulis oleh seorang programmer dan ditulis dengan salah satu bahasa pemrograman menggunakan dua paradigma utama yaitu pemrograman **imperatif** atau pemrograman **deklaratif**.

Disini Penulis menggunakan CodeVision AVR sebagai aplikasi untuk memprogram microcontroller ATmega8.

4.4.1 Pengenalan CodeVision AVR

CodeVision AVR Penggunaan mikrokontroler sekarang ini telah umum. Mulai dari penggunaan untuk kontrol sederhana sampai kontrol yang cukup kompleks, mikrokontroler dapat berfungsi jika telah diisi sebuah program, pengisian program ini dapat dilakukan menggunakan compiler yang selanjutnya didownload ke dalam mikrokontroler menggunakan *downloader*. Salah satu compiler program yang umum

digunakan sekarang ini adalah CodeVision AVR yang menggunakan bahasa pemrograman C.

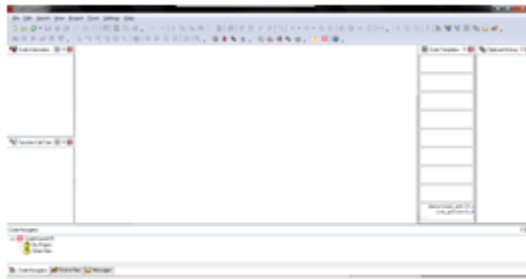
CodeVision AVR mempunyai suatu keunggulan dari compiler lain, yaitu adanya codewizard, fasilitas ini memudahkan kita dalam inisialisasi mikrokontroler yang akan kita gunakan, codevision telah menyediakan konfigurasi yang bisa diatur pada masing-masing chip mikrokontroler yang akan kita gunakan, sehingga kita tidak perlu melihat *datasheet* untuk sekedar mengonfigurasi mikrokontroler. Berikut ini langkah-langkah menggunakan codevision.

4.4.2 Cara Pengoperasian Aplikasi

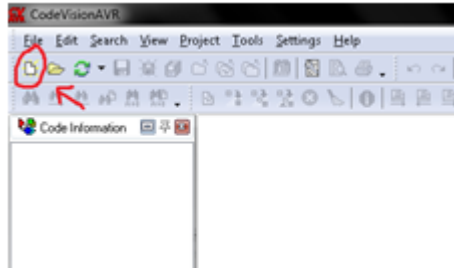
1. Buka aplikasi CodeVision AVR



2. Setelah itu jendela CodeVision akan terbuka, dan menampilkan project kosong.



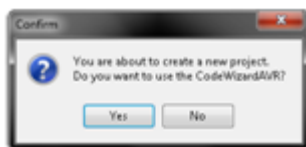
3. Pilih Create a New File or Project



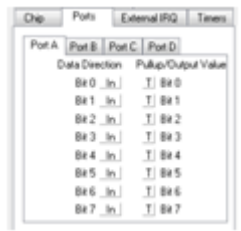
4. Setelah itu muncul dialog box untuk memilih tipe file yang akan dibuat. Pilih file project, kemudian tekan OK.



5. Ketika dikonfirmasi apakah ingin menggunakan CodeWizard? pilih Yes. Ini merupakan fasilitas yang dapat memudahkan pemrogram melakukan konfigurasi mikrokontroler.



6. Jendela baru CodeWizard akan muncul seperti gambar berikut :



10. Pada gambar diatas ditunjukkan bahwa Port A seluruhnya digunakan sebagai input.

Jika kita ingin mengganti beberapa pin sebagai output, maka klik tombol disebelah tulisan Bit 'x'. Seperti dicontohkan pada gambar berikut :



11. Perlakuan yang sama dapat diterapkan untuk Port B, Port C, dan Port D.

12. Kemudian Pilih File -> Generate, Save, and Exit, setelah itu tampil dialog box untuk menyimpan File source (*.c), file project (*.prj), dan file CodeVisionAVR project (*.cwp). Simpan dengan nama yang sama untuk memudahkan pengelompokan file.

13. Setelah itu muncul source code dengan konfigurasi mikro yang sesuai dengan pemilihan pada wizard.

```

Chip type           : ATmega8535
Program type        : Application
AVR Core Clock frequency: 12.000000 MHz
Memory model        : Small
External RAM size    : 0
Data Stack size     : 128
*****
#include <mega8535.h>
// Declare your global variables here

```

Setelah pembuatan program selesai, maka wajib untuk meng-load ke dalam microcontroller yang sudah terhubung dengan minimum sistem yang dapat membantu untuk memasukkan program dalam chip microcontroller tersebut.

4.5 Tahap Penggabungan Antar Rangkaian Hasil Kerja

Setelah kita membuat seluruh rangkaian yang diperlukan seperti pada tahapan perancangan rangkaian elektronika di atas, maka sekarang adalah tahapan untuk menggabungkan seluruh elemen yang penulis buat untuk di gabungkan menjadi satu rangkaian.

Tahapan yang perlu untuk digabungkan adalah sebagai berikut :

1. Rangkaian remote control transmitter dan receiver
2. Rangkaian microcontroller ATmega8 dan driver motor
3. Peletakkan limit switch
4. Peletakan motor DC dengan pintu pagar

4.5.1 Rangkaian Remote Control Transmitter dan Receiver

Disini penulis akan menjelaskan bagaimana tahapan menghubungkan cara kerja transmitter dan receiver dari suatu remote control. Dalam hal ini penulis membuat tombol sebanyak 4 (empat) buah. Fungsi dari seluruh tombol adalah sebagai berikut :

1. Tombol atas : Berfungsi sebagai membuka pintu pagar secara otomatis.
2. Tombol bawah : Berfungsi sebagai menutup pintu pagar secara otomatis.
3. Tombol kanan : Berfungsi sebagai menutup pintu pagar secara perlahan.
4. Tombol kiri : berfungsi sebagai membuka pintu pagar secara perlahan.

4.5.2 Rangkaian Microcontroller ATmega8 dan Driver motor

Dalam penggabungan antara microcontroller dengan driver motor, diperlukan sebuah minimum system dalam microcontroller yang berfungsi untuk memasukkan program perintah untuk menjalankan fungsi dari driver motor. Seperti di ulaskan dalam bab 4.3 terdapat beberapa port yang akan memiliki fungsi output untuk menjalankan fungsi dari motor driver. Minimum system harus memiliki daya tidak lebih dari 5V. Tetapi jika memiliki rangkaian regulator maka kita dapat memberikan input tegangan sebesar lebih dari 5V.

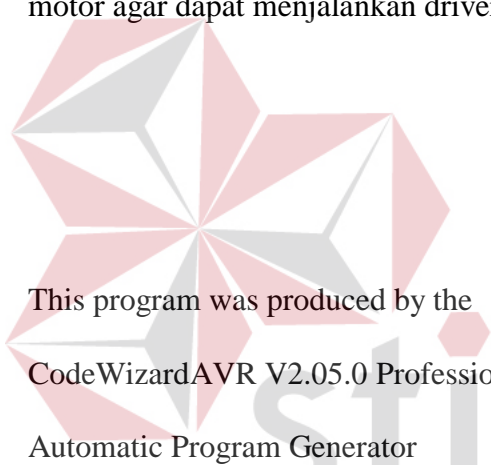
Setelah itu driver motor memiliki 4 buah pin yang semua memiliki fungsi berbeda, yaitu :

1. Pin 1 : Berfungsi untuk menjalankan enable A yang berfungsi untuk menjalankan motor ke arah maju.

2. Pin 2 : Berfungsi untuk menjalankan enable B yang berfungsi untuk menjalankan motor kea rah mundur .
3. Pin 3 : Berfungsi sebagai input tegangan 5V untuk menjalankan IC L298
4. Pin 4 : Berfungsi sebagai ground tegangan dari driver motor.

4.5.3 Program Untuk Menjalankan Driver Motor

Dinisi penulis akan mencantumkan sebuah program untuk menjalankan driver motor agar dapat menjalankan driver motor sesuai dengan perintah.



This program was produced by the
CodeWizardAVR V2.05.0 Professional
Automatic Program Generator

© Copyright 1998-2010 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 9/24/2011

Author : NeVaDa

Company :

Comments:

Chip type : ATmega8

Program type : Application

AVR Core Clock frequency: 11.059200 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

*****/

#include <mega8.h>

#include <delay.h>

#define kanan_manual PIND.0

#define kiri_manual PIND.1

#define kanan_otomatis PIND.2

#define kiri_otomatis PIND.3

#define batas_min PIND.4

#define batas_max PIND.5

#define buka PORTC.3

#define tutup PORTC.2

#define oto PORTC.4

// Declare your global variables here

```
unsigned char flag_buka, flag_tutup, darurat;
```

```
void buka_otomatis();
```

```
void main(void)
```

```
{
```

```
// Declare your local variables here
```

```
// Input/Output Ports initialization
```

```
// Port B initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
```

```
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
```

```
PORTB=0x00;
```

```
DDRB=0x00;
```

```
// Port C initialization
```

```
// Func6=In Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
```

```
// State6=T State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
```

```
PORTC=0x00;
```

```
DDRC=0x3F;
```

```
// Port D initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In

// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T

PORTD=0x00;

DDRD=0x00;


// Timer/Counter 0 initialization

// Clock source: System Clock

// Clock value: Timer 0 Stopped
TCCR0=0x00;
TCNT0=0x00;


// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer1 Stopped
// Mode: Normal top=0xFFFF

// OC1A output: Discon.

// OC1B output: Discon.

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off
```



```
TCCR1A=0x00;

TCCR1B=0x00;

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x00;

ICR1L=0x00;

OCR1AH=0x00;

OCR1AL=0x00;

OCR1BH=0x00;

OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected

ASSR=0x00;

TCCR2=0x00;

TCNT2=0x00;

OCR2=0x00;

// External Interrupt(s) initialization

// INT0: Off
```



```
// INT1: Off
```

```
MCUCR=0x00;
```

```
// Timer(s)/Counter(s) Interrupt(s) initialization
```

```
TIMSK=0x00;
```

```
// USART initialization
```

```
// USART disabled
```

```
UCSRB=0x00;
```

```
// Analog Comparator initialization
```

```
// Analog Comparator: Off
```

```
// Analog Comparator Input Capture by Timer/Counter 1: Off
```

```
ACSR=0x80;
```

```
SFIOR=0x00;
```

```
// ADC initialization
```

```
// ADC disabled
```

```
ADCSRA=0x00;
```

```
// SPI initialization
```

```
// SPI disabled
```

```
SPCR=0x00;
```

```
// TWI initialization

// TWI disabled

TWCR=0x00;

//kanan_manual = kiri_manual = 0;

tutup = buka = oto = 0;

flag_buka = flag_tutup = darurat = 0;

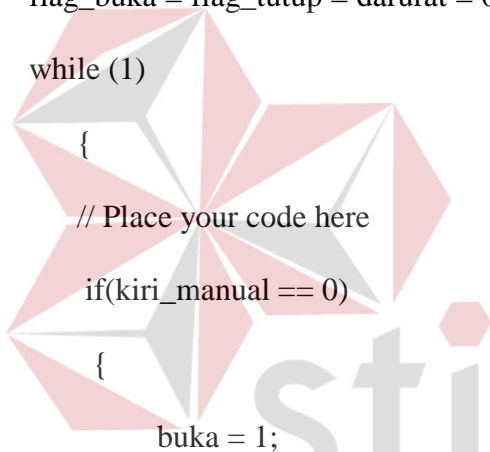
while (1)
{
    // Place your code here
    if(kiri_manual == 0)
    {
        buka = 1;

    }

    else if(kanan_manual == 0)
    {
        tutup = 1;

    }

    else if(kanan_otomatis == 0)
```



INSTITUT BISNIS
& INFORMATIKA
stikom
SURABAYA

```
{  
    flag_buka = 1;  
    //darurat = 0;  
    while(flag_buka == 1)  
    {  
        buka = 1;  
        delay_ms(500);  
        buka = 0;  
        delay_ms(500);  
        ///flag_buka = 1;  
        if(batas_max == 1 && batas_min == 0)  
        {  
            flag_buka = 0;  
            delay_ms(20);  
            if(batas_max == 1 && batas_min == 0)  
            {flag_buka = 0; buka = 0;}  
        }  
    }  
    }  
    buka = 0;  
}  
else if(kiri_otomatis == 0 )
```



```
{  
    flag_tutup = 1;  
    //darurat = 0;  
    while(flag_tutup == 1)  
    {  
        tutup = 1;  
        delay_ms(500);  
        tutup = 0;  
        delay_ms(500);  
        //flag_tutup = 1;  
        if(batas_min == 0)  
        {  
            flag_tutup = 0;  
            delay_ms(20);  
            if(batas_min == 0)  
            { flag_tutup = 0; tutup = 0; }  
        }  
    }  
    }  
    tutup = 0;  
}  
else  
{
```

```

    oto = tutup = buka = 0;

}

};

}

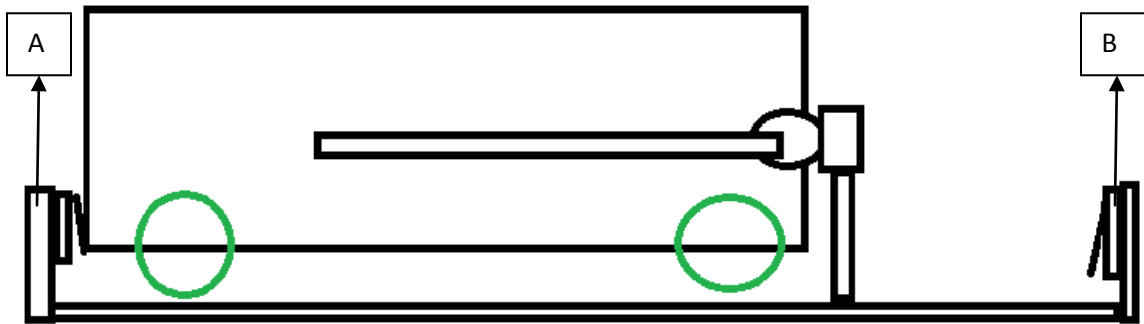
```

Dari program diatas kita sudah dapat melakukan perintah untuk menjalankan fungsi dari driver motor untuk menjalankan motor DC yang memiliki fungsi untuk membuka dan menutup pagar. Secara rinci program dibuat sedemikian rupa sehingga memiliki peranan penting untuk mengakses pintu pagar secara otomatis maupun manual, sehingga pintu pagar dapat menutup dan membuka sesuai dengan kebutuhan kita.

4.5.4 Peletakan Limit Switch

Disini penulis memakai 2 buah limit switch. Fungsi dari limit switch tersebut adalah sebagai sebuah sensor untuk mengetahui bahwa pintu pagar sudah menutup dengan kondisi penuh dan membuka secara penuh.

Letak dari limit switch terdapat dalam ujung lintasan pagar. Sehingga dapat tersentuh dengan mudah saat posisi pintu pagar tertutup dan terbuka secara maksimal.



Gambar 9.1 Letak dari Limit Switch

Keterangan :

A = Limit Switch batas maksimum

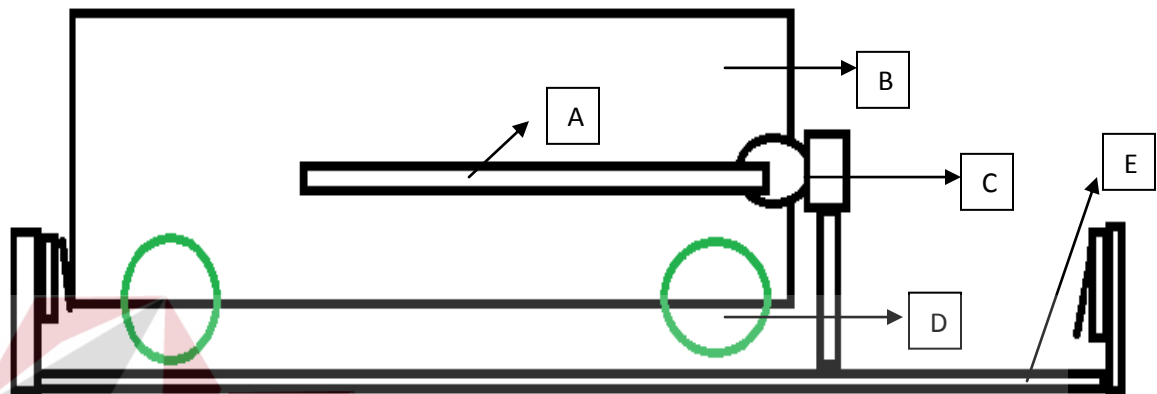
B = Limit Switch batas minimum

4.5.5 Peletakan Motor DC dengan Pintu Pagar

Disini penulis akan mengulas secara jelas bagaimana bentuk pagar dan bagaimana peletakan motor DC untuk menjalankan membuka dan menutup pintu pagar. Pintu pagar yang penulis buat adalah dengan bahan akrilik / mika. Kenapa penulis pakai, dikarenakan pintu pagar ini bersifat simulasi dan mudah untuk di bongkar ataupun di pasang kembali. Karena tugas dari kerja praktek ini adalah mensimulasikan pintu pagar otomatis. Dapat juga di implementasikan di pintu pagar secara nyata, tetapi perlu merakit secara permanen agar mendapatkan hasil yang baik.

Untuk memiliki fungsi sebagai membuka dan menutup pagar penulis menggunakan ulir yang tersambung dengan mur / baut berbungsi untuk menjalankan

pintu pagar yang disambungkan dengan motor DC agar dapat menjalankan membuka dan menutup pintu pagar. Mur tersebut di tanam dengan sebuah plat yang di tempelkan dengan pintu pagar.



Gambar 9.2 Bentuk dari pintu pagar

Keterangan :

- A = Adalah ulir yang terhubung dengan motor DC .
- B = Bentuk dari simulasi pintu pagar berbahan akrilik / mika.
- C = Motor DC
- D = Roda yang berfungsi sebagai penggerak dari pintu pagar.
- E = Lintasan / jalur dari pintu pagar

4.6 Cara Kerja

Cara kerja dari hasil kerja praktek ini adalah jika tombol atas yang di tekan, maka pintu pagar secara otomatis terbuka sampai dengan titik maksimum atau mengenai sensor limit switch yang mengindikasikan maksimum. Dan jika tombol bawah di tekan, maka pintu pagar tertutup secara otomatis sampai dengan titik maksimum atau dengan menyentuh sensor limit switch yang mengindikasikan minimum.

Tombol kanan dan kiri memiliki fungsi yang berbeda, tetapi dengan prinsip yang sama yaitu dengan menyentuh tombol kanan maka pintu terbuka tetapi dengan step bahwa jika tombol di lepas maka pintu berhenti.

