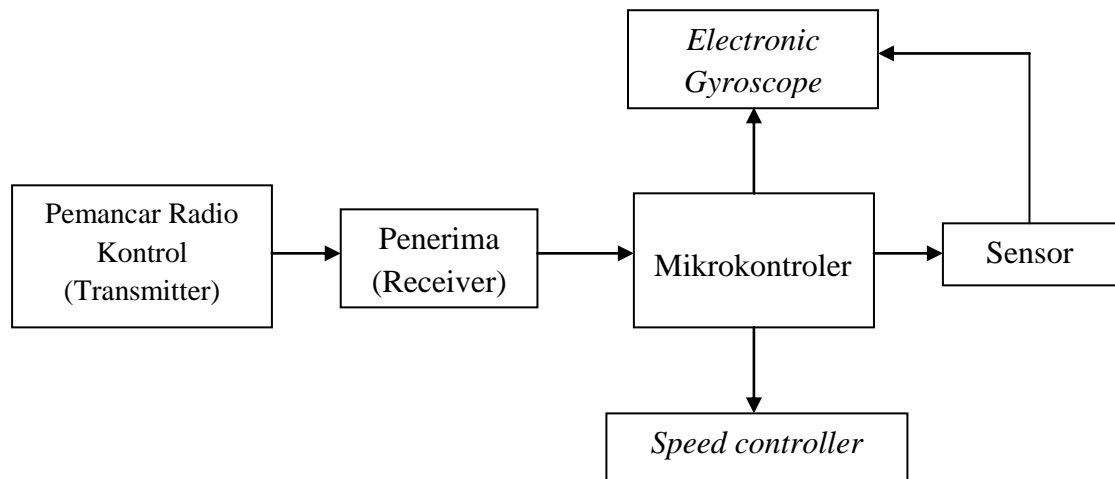


BAB III

PERANCANGAN ALAT DAN PROGRAM

3.1 Diagram Blok

Secara garis besar, diagram blok Simulasi Pendaratan Helikopter Mainan ditunjukkan pada gambar berikut ini :



Gambar 3.1 Diagram Blok Rangkaian

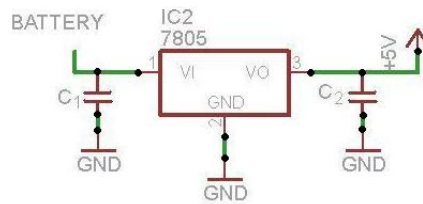
Pada gambar diatas fungsi mikrokontroler hanya melewatkan data dari penerima pada *speed controller* dan *gyroscope* disaat koneksi dari pemancar dan penerima masih tersambung, dan saat koneksi dari pemancar terputus maka mikrokontroler mengambil alih sistem navigasi helikopter.

Cara kerja dan fungsi masing-masing tiap blok adalah:

1. Radio kontrol pada gambar diatas hanyalah navigator arah dan ketinggian helikopter secara manual yang menggunakan gelombang radio sebagai media komunikasinya.
2. Sensor Ping Ultrasonik berfungsi sebagai acuan jarak antara helikopter dan landasan.
3. *Electronic gyroscope* adalah perbandingan putaran baling-baling atas dan baling-baling bawah untuk arah helikopter (kanan atau kiri, sehingga perbedaan putaran yang akan menentukan posisi helikopter, dan saat banyaknya putaran baling-baling utama (atas dan bawah) sama maka posisi helikopter akan cenderung diam secara horizontal (badan helikopter tidak berputar terhadap baling baling), *gyroscope* akan bekerja sesuai dengan input yang diberikan *speed controller* .
4. *Speed controller* adalah nilai output tegangan pwm yang menentukan ketinggian helikopter (putaran baling-baling atas harus sama dengan putaran baling-baling bawah), dan arah badan helikopter (putaran baling-baling atas berbeda dengan putaran baling-baling bawah) , dan input pada baling-baling ekor untuk maju atau mundur dalam tiap menitnya (*RPM-Revolution Per Minutes*) yang menentukan ketinggian dan arah pergerakan helikoter.
5. Mikrokontroler berfungsi untuk melewatkan data dari pemancar ke penerima pada *gyroscope* dan *speed controller*, dan membaca ketinggian helikopter melalui. Saat koneksi pemancar ke penerima terputus maka mikrokontroler akan mengatur putaran baling-baling utama sesuai data terakhir yang dikirim pemancar, dan giroskop akan mengatur posisi dimana helikopter akan mendarat. Saat masing-masing sensor ping ultrasonik membaca jarak yang sama maka kecepatan putaran baling-baling utama akan berkurang dengan perlahan sehingga helikopter akan mendarat ditempat yang datar dengan aman.

3.2 Rangkaian Regulator

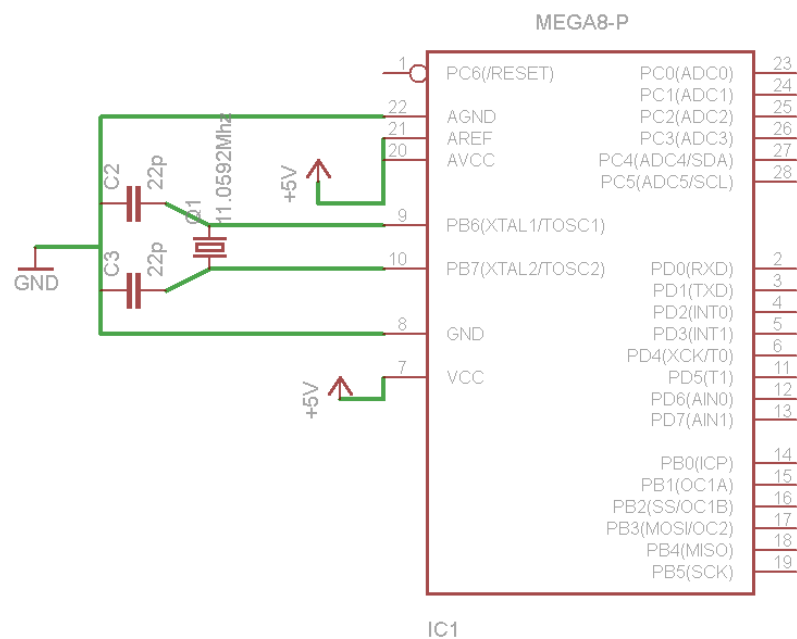
Agar rangkaian dapat bekerja, maka dibutuhkan sumber tegangan. Pada perancangan ini dibutuhkan sumber tegangan yang konstan, yaitu (+) 5 volt DC, yang digunakan untuk mensupply tegangan ke IC ATmega 8 dan sensor ping ultrasonik. Rangkaian tampak seperti gambar di bawah ini:



Gambar 3.2 Rangkaian Regulator

3.3. Perancangan Rangkaian Mikrokontroler ATmega 8

Rangkaian ini berfungsi untuk mengendalikan seluruh sistem. Komponen utama dari rangkaian ini adalah IC mikrokontroler ATmega 8. Pada IC ini program diisi sehingga rangkaian dapat berjalan sesuai dengan yang dikehendaki. Rangkaian mikrokontroler ditunjukkan pada gambar berikut ini:



Gambar 3.3 Rangkaian mikrokontroler ATmega 8

Mikrokontroler ini memiliki 24 port I/O, yaitu port B, port C dan port D. Pin 23 sampai 28 adalah Port C yang merupakan port ADC, dimana port ini dapat menerima data analog. Untuk supply tegangan Pin 7 dihubungkan ke sumber tegangan 5 volt sedangkan pin 8 dan pin 22 dihubungkan ke ground. Rangkaian mikrokontroler ini menggunakan komponen kristal sebagai sumber clocknya pada pin 9 dan pin 10. Nilai kristal ini akan mempengaruhi kecepatan mikrokontroler dalam mengeksekusi suatu perintah tertentu.

3.4 Baterai

Besar yang digunakan adalah baterai Lithium-Ion dengan tegangan 7.4volt dan arus 1500mAh yang dapat kita lihat pada gambar 3.4.



Gambar 3.4 Baterai Lithium-Ion

3.5 Rangkaian Sensor Jarak (PING)

Sensor jarak yang digunakan adalah sensor PING ultrasonic. Sensor ini dapat mengukur jarak hingga 3 meter, sensor ini menggunakan metode pemancaran dan penerimaan sinyal suara ultrasonic. Gambar sensor jarak PING ditunjukkan pada gambar 2.6.

Sensor jarak PING ultrasonic ini didesign untuk mengukur jarak. Namun jarak yang diukur hanya sebatas 3 meter. Sensor ini terdiri dari 3 pin, yaitu satu pin untuk Vcc 5 Volt, satu pin untuk ground dan satu pin untuk sinyal.

3.6 Helikopter

Helikopter yang digunakan untuk proyek ini adalah tipe Double Horse Volitation 9053 Metal Electric RC Helicopter yang memiliki dua baling-baling pada kepala helikopter (untuk mengatur ketinggian dan arah helikopter) dan baling-baling pada ekor helikopter (untuk pergerakan maju atau mundur).

3.6.1 Gyroscope dan Speed controller



Gambar 3.5 Baling-baling utama helikopter

Pada gambar diatas kita dapat melihat bahwa terdapat dua baling-baling utama (Double-Blade) yang digunakan sebagai giroskop elektronik, dan baling-baling ekor pada gambar dibawah yang berfungsi sebagai tenaga pendorong helikopter (untuk maju atau mundur saat diudara).



Gambar 3.6 Baling-baling ekor helicopter

3.6.2 Pemancar (*Transmitter*)

Pemancar yang digunakan merupakan pemancar radio (*radio remote control*) yang beroperasi pada frekuensi 49 MHz terdiri dari beberapa tombol navigasi yang menjadi input arah dan ketinggian helikopter.



Gambar 3.7 Pemancar radio kontrol

3.6.3 Penerima (*Receiver*)

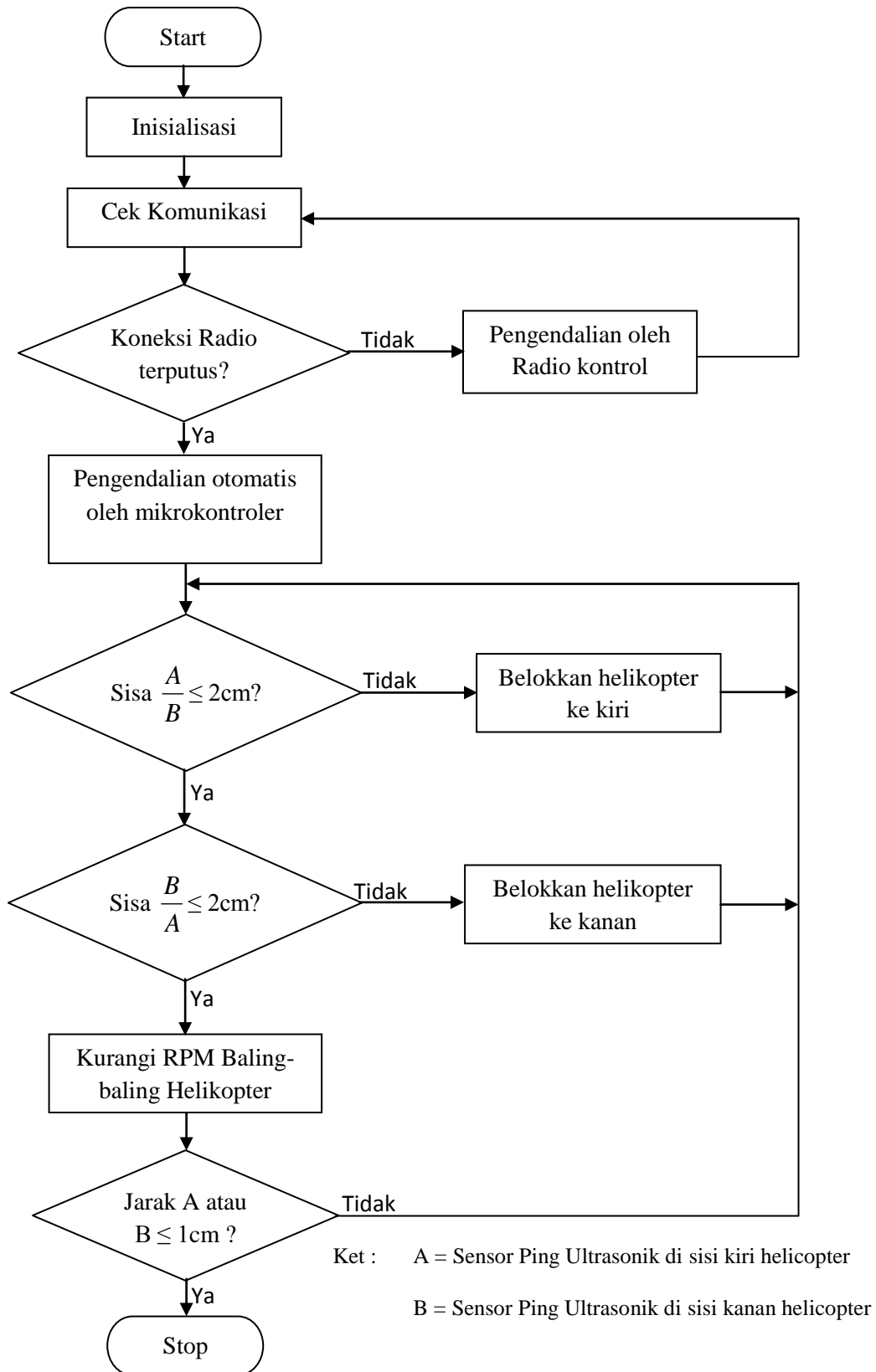
Pada modul penerima telah terdapat juga driver motor untuk baling-baling utama dan baling-baling ekor. Modul penerima beroperasi pada frekuensi 49 MHz yang dapat kita lihat pada gambar 3.8.



Gambar 3.8 Modul penerima pada helikopter

3.7 Perancangan Program

Diagram alir pemrogramannya adalah sebagai berikut :



Gambar. 3.9 Diagram Alir Rangkaian

Keterangan:

1. **Start** : Memulai Program.
2. **Inisialisasi** : Menentukan nilai-nilai input yang digunakan.
3. **Cek komunikasi** : Sistem mengecek komunikasi pada penerima.
4. **Koneksi terputus ?** : Ya, hanya berfungsi saat helikopter berada di udara (Belum/Tidak mendarat), karena pengendalian dapat dilakukan radio kontrol bila koneksi antara pemancar dan penerima tidak terputus (baik udara maupun darat).
5. **Pengendalian otomatis oleh mikrokontroler** : Mikrokontroler mengambil alih sistem navigasi untuk mendaratkan helikopter.
6. **Sisa $\frac{A}{B} \leq 2\text{cm}$?** : Untuk Mengetahui sisa hasil pembagian jarak antara sensor A terhadap sensor B. Jika ya, maka lanjutkan ke proses berikutnya, namun bila tidak maka mikrokontroller akan memberi perintah untuk membelokkan helikopter ke kiri.
7. **Sisa $\frac{B}{A} \leq 2\text{cm}$?** : Untuk Mengetahui sisa hasil pembagian jarak antara sensor B terhadap sensor A. Jika ya, maka lanjutkan ke proses berikutnya, namun bila tidak maka mikrokontroller akan memberi perintah untuk membelokkan helikopter ke kanan.
8. **Kurangi RPM Baling-baling Helikopter** : Pengurangan cepatnya putaran baling-baling (RPM) untuk mendaratkan helikopter.
9. **Jarak A atau B $\leq 1\text{cm}$?** : Merupakan jarak antara masing-masing sensor terhadap landasan. Jika ya, maka mikrokontroler akan menghentikan semua proses, jika tidak maka akan kembali pada proses pengukuran jarak sensor terhadap landasan.
10. **Stop** : Menghentikan aktivitas semua program.

BAB IV

PENGUJIAN ALAT DAN PROGRAM

Pada bab ini, akan dibahas pengujian alat mulai dari pengujian alat permodul sampai pengujian alat secara keseluruhan. Pengujian tersebut akan dilakukan secara bertahap dengan urutan sebagai berikut:

- a. Pengujian masing-masing blok
- b. Pengujian sistem secara keseluruhan

4.1 Pengujian Mikrokontroler ATMEGA8

Untuk mengetahui apakah rangkaian mikrokontroler bekerja dengan baik, maka dilakukan pengujian. Pengujian pada bagian ini dilakukan dengan memberikan program pada mikrokontroler. Programnya adalah sebagai berikut :

```
/******
```

```
Project : Test ATmega8
```

```
Version : Bab IV
```

```
Date   : 04/09/2011
```

```
Author : Freeware, for evaluation and non-commercial use only
```

```
Company : FMIPA USU
```

```
Comments:
```

```
by: Jony smith sinambela
```

```
Chip type      : ATmega8
```

```
Program type   : Application
```

```
Clock frequency : 12,00000 MHz
```

```
Memory model   : Small
```

```
External RAM size : 0
```

```
Data Stack size : 256
```

```
*****/
```

```
#include <mega8.h>
```

```
#include <stdio.h>
```

```
#include <delay.h>
```

```

// Declare your global variables here
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=0xFF;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0xFF;

// 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=0xFF;

// 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: Timer 1 Stopped

```

```

// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 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: Timer 2 Stopped
// Mode: Normal top=FFh
// 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;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
while (1)
{
    // Place your code here
    PORTB=0x00;
    delay_ms(200);
    PORTB=0xFF;
    delay_ms(200);

    PORTC=0x00;
    delay_ms(200);
    PORTC=0xFF;
    delay_ms(200);

    PORTD=0x00;
    delay_ms(200);
    PORTD=0xFF;
    delay_ms(200);
};
}

```

Jika program tersebut dijalankan, maka led yang terhubung pada port b, port c, dan port d akan berkedip, dengan demikian maka system minimum mikrokontroller ATMEGA8 telah berjalan dengan baik.

4.2. Pengujian *Speed Controller* dan *Electronic Gyroscope*

Untuk mengetahui kinerja *speed controller* (PWM) dan input *gyroscope* bekerja dengan baik, maka dilakukan pengujian. Pengujian pada bagian ini dilakukan dengan memberikan program pada mikrontroller. Programnya adalah sebagai berikut :

```
/******
```

```
Project : Speed controller (PWM) & Gyroscope
```

```
Version : Bab IV
```

```
Date : 04/09/2011
```

```
Author : Freeware, for evaluation and non-commercial use only
```

```
Company : FMIPA USU
```

```
Comments:
```

```
by: Jony smith sinambela
```

```
Chip type : ATmega8
```

```
Program type : Application
```

```
Clock frequency : 12,00000 MHz
```

```
Memory model : Small
```

```
External RAM size : 0
```

```
Data Stack size : 256
```

```
*****/
```

```
#include <mega8.h>
```

```
#include <stdio.h>
```

```
// Alphanumeric LCD Module functions
```

```
#asm
```

```
.equ __lcd_port=0x12 ;PORTD
```

```
#endasm
```

```
#include <lcd.h>
```

```
#include <delay.h>
```

```
// Declare your global variables here
```

```
void main(void)
```

```
{
```

```

// Declare your local variables here
Unsigned char i;
// Input/Output Ports initialization

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=Out Func1=Out
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=0 State1=0 State0=T
PORTB=0x00;
DDRB=0x06;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// 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: 7372,800 kHz
// Mode: Fast PWM top=00FFh

```

```

// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xA1;
TCCR1B=0x09;
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: Timer 2 Stopped
// Mode: Normal top=FFh
// 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;

```

```

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOA=0x00;
// LCD module initialization
lcd_init(16);
while (1)
{
    // Place your code here
    { if (OCR1AL==0xFF & OCR1BL==0xFF)
        {
            OCR1AL=0xFF;
            OCR1BL=0xFF;
            delay_ms(500);

            for (i=0; i<255; i++)
            {
                OCR1AL--;
                OCR1BL--;
                delay_ms(10);
            };
            OCR1AL=0x00;
            OCR1BL=0x00;
            delay_ms(500);
        }
        else if (OCR1AL<0xFF & OCR1BL<0xFF)
        {
            OCR1AL++;
            OCR1BL++;
            delay_ms(10);
        }
    }
}

```


Jika program tersebut dijalankan, maka tegangan pada pin 16 atau pin 17 akan naik dari 0 Volt s/d 5 Volt dan turun dari 5 Volt s/d 0 Volt terhadap ground secara bergantian, dengan demikian maka *speed controller* dan *gyroscope* telah bekerja dengan baik.

4.3. Pengujian Rangkaian Sensor Ping Ultrasonik

Untuk mengetahui apakah sensor ini berjalan dengan baik, maka dilakukan pengujian dengan menghubungkan kedua sensor ping ultrasonik pada rangkaian ATmega8 dan memberikan program pada IC Mikrokontroler. Programnya adalah sebagai berikut:

```
/******
```

```
Project : Ping Ultrasonic Test
```

```
Version : Bab IV
```

```
Date : 12/10/2011
```

```
Author : Freeware, for evaluation and non-commercial use only
```

```
Company : FMIPA USU
```

```
Comments:
```

```
by: Jony smith sinambela
```

```
Chip type : ATmega8
```

```
Program type : Application
```

```
Clock frequency : 12,00000 MHz
```

```
Memory model : Small
```

```
External RAM size : 0
```

```
Data Stack size : 256
```

```
*****/
```

```
#include <mega8.h>
```

```
#include <stdio.h>
```

```
// Alphanumeric LCD Module functions
```

```
#asm
```

```
.equ __lcd_port=0x12 ;PORTD
```

```

#endasm
#include <lcd.h>
#include <delay.h>

#define pulse1 PORTC.4
#define echo1 PINC.4
#define arah1 DDRC.4
#define out1 1
#define inp1 0
unsigned int count1=0;
unsigned int jarak1;
unsigned char baris1[16];

// Declare your global variables here
void pengukuran(void)
{
    arah1=out1;
    pulse1=1;
    delay_us(5);
    pulse1=0;
    arah1=inp1;
    pulse1=1;
    while (echo1==1 )
    {
        count1++;
    };
    jarak1=(count1*0.034446);
    sprintf(baris1,"Jarak=%2d Cm",jarak1);
    lcd_gotoxy(0,0);
    lcd_puts(baris1);
    count1=0;
    delay_ms(10);
}

```

```

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=Out Func1=Out
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=0 State1=0 State0=T
PORTB=0x00;
DDRB=0x06;
// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;
// 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: 1500,000 kHz
// Mode: Fast PWM top=00FFh
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge

```

```

// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0xA1;
TCCR1B=0x0A;
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: Timer 2 Stopped
// Mode: Normal top=FFh
// 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;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;

```

```

SFIOR=0x00;
// LCD module initialization
lcd_init(16);
while (1)
{
// Place your code here
jony();
smith();
};
}

```

Setelah program tersebut dijalankan, maka dilakukan pengukuran terhadap jarak yang nyata sehingga didapatkan data sebagai berikut :

Tabel 4.1 Pengukuran Jarak Sensor Ultrasonik

No	Pengukuran Sensor Ultrasonik (Cm)	Jarak Sebenarnya (Cm)
1	15 cm	14.8 cm
2	14 cm	13.8 cm
3	13 cm	12.8 cm
4	12 cm	11.8 cm
5	11 cm	10.8 cm
6	10 cm	9.8 cm
7	9 cm	8.8 cm
8	8 cm	7.8 cm
9	7 cm	6.8 cm
10	$\Sigma = 99cm$	$\Sigma = 97.2cm$

Terdapat perbedaan nilai antara pengukuran jarak melalui sensor dan jarak sebenarnya, dimana terdapat penambahan jarak pada pengukuran sensor 0.2 Cm jika dibandingkan pengukuran aslinya. Untuk menghitung tingkat kesalahan (*error rate*) pada pengukuran tersebut digunakan formula :

$$\begin{aligned} \text{\% tingkat kesalahan rata-rata} &= \left| \frac{99 - 97.2}{97.2} \right| \times 100\% \\ &= 1.851\% \end{aligned}$$

Maka besar tingkat kesalahannya adalah 1.851 %, namun dengan demikian kita dapat mengetahui bahwa modul sensor telah bekerja dengan baik.

4.4 Pengujian Pemancar dan Pembacaan Tegangan Pada Modul Penerima

Langkah pengujian sebagai berikut :

- a. Menghubungkan modul penerima dengan modul ATmega8 dan catu daya
- b. Mencatat hasil pengukuran tegangan.

Tabel 4.2 Pengukuran Tegangan pada Modul Penerima

No	Kondisi Radio Kontrol	V pada Modul Penerima (Volt)	Tampilan Pada LCD
1	ON	0.33	143
2	OFF	0.24	92

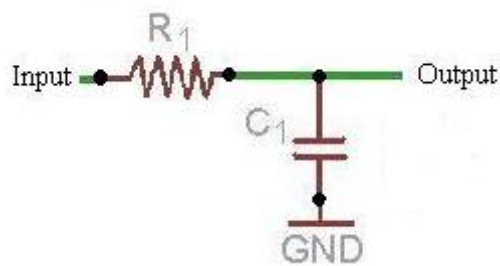
Tegangan pada modul penerima “0.33 volt” merupakan nilai tegangan saat kondisi pemancar aktif (on) dan “0.24 volt” merupakan nilai tegangan saat kondisi pemancar dinonaktifkan (off). Angka “143” dan “92” merupakan hasil konversi nilai tegangan menjadi biner pada ADC (Analog to Digital Converter) oleh mikrokontroler yang ditampilkan oleh LCD.

Tabel 4.3 Pengukuran pengaruh Tegangan terhadap jarak pemancar

No.	Jarak Pemancar dari penerima (meter)	V pada Modul Penerima (volt)
1	5	0.33
2	15	0.29
3	25	0.25
4	40	0.22
5	65	0.16
6	>70	<0.15

Besarnya nilai tegangan yang terukur dipengaruhi jarak antara pemancar dan modul penerima, dari tabel 4.2 kita dapat melihat bahwa besar tegangan saat kondisi “ON” adalah 0.33 volt dan saat kondisi “OFF” adalah 0.24 volt, namun saat pengukuran dengan pengaruh jarak antara pemancar dan penerima terdapat variabel tegangan yang berada diantara posisi ON, OFF dan dibawah tegangan OFF, pada tabel 4.3 kita dapat melihat bahwa tegangan pemancar dikatakan OFF bila jarak lebih dari 25 meter, karena besar tegangan saat kondisi OFF adalah 0.24 volt. Dari pengukuran tersebut dapat disimpulkan bahwa pada saat jarak antara pemancar dan penerima lebih dari 25 meter, helikopter akan mendarat secara otomatis.

Tegangan output dari modul penerima berupa PWM (pulse width modulation) yang tidak dapat dibaca mikrokontroler secara sempurna melalui ADC (Analog to Digital Converter), sehingga dibutuhkan rangkaian PWM to Analog Converter (PAC) agar mikrokontroler dapat membaca tegangan pada modul penerima (gambar 4.1).



Gambar 4.1 Rangkaian PAC

Tabel 4.4 Pengukuran Tegangan Input Driver pada Modul Penerima

No	$V_{(in)}$ Modul Penerima Sebelum PAC (Volt)	$V_{(out)}$ Modul Penerima Setelah PAC (Volt)	Tampilan Pada LCD Setelah PAC
1	0	0	0
2	1.67	0.22	89
3	2.46	0.32	130
4	3.17	0.41	168
5	3.77	0.49	200
6	4.53	0.59	240

Terdapat pengurangan tegangan (Voltage Reduction) dengan digunakannya rangkaian PAC. Tegangan yang berkurang sekitar 87% dari tegangan setelah rangkaian PAC, yang didapatkan dengan metode pengambilan sampel (sampling), dengan formula :

$$100\% - \left(\frac{V_{(out)}}{V_{(in)}} \right) \times 100\% = V_reduction, \text{ maka :}$$

$$\text{Sample pertama : } 100\% - \left(\frac{0.22}{1.67} \right) \times 100\% = 86.8\%$$

$$\text{Sample Kedua : } 100\% - \left(\frac{0.49}{3.77} \right) \times 100\% = 87.0\%$$

Sehingga dapat ditarik kesimpulan bahwa besarnya tegangan yang berkurang sekitar 87% dari tegangan $V_{(in)}$ setelah rangkaian PAC. Tampilan nilai pada Lcd setelah rangkaian PAC merupakan hasil konversi tegangan menjadi nilai biner pada ADC oleh mikrokontroler, yang nantinya menjadi nilai acuan PWM pada mikrokontroler.

Tegangan yang dihasilkan pada modul penerima yang sebelumnya menjadi acuan cepatnya putaran motor pada helikopter (RPM), dan mikrokontroler harus dapat mereplika nilai-nilai dan perubahan pada tegangan tersebut, sehingga input tegangan (PWM) pada modul penerima dapat sama dengan output tegangan pada mikrokontroler (PWM) yang menjadi acuan baru pada cepatnya putaran motor pada helikopter (RPM).

Tabel 4.5 Perbandingan $V_{(in)}$ Modul Penerima – $V_{(out)}$ pada Mikrokontroler

No.	$V_{(in)}$ Modul Penerima Sebelum PAC (Volt)	Tampilan Pada LCD Setelah PAC	$V_{(out)}$ pada Mikrokontroler (Volt)
1	0	0	0
2	1.67	89	1.74
3	2.46	130	2.49
4	3.17	168	3.26
5	3.77	200	3.89
6	4.53	240	4.64

Perbedaan nilai tegangan yang dihasilkan mikrokontroler ($V_{(out)}$) terhadap nilai tegangan dari modul penerima ($V_{(in)}$) merupakan replikasi tegangan yang menjadi

acuan cepatnya putaran pada baling-baling utama helikopter, dalam pengukuran ini terjadi perubahan nilai tegangan dari tegangan input yang lebih kecil dibandingkan tegangan output, nilai pada Lcd menjadi acuan tegangan replika pada mikrokontroler (PWM).

4.4 Pengujian Rangkaian Keseluruhan dan Pengoperasian Alat

Untuk menguji rangkaian keseluruhan maka IC Mikrokontroler harus diisi dengan program keseluruhan yang terdapat pada lampiran di akhir laporan. Pengoperasian alat dilakukan dalam dua tahap untuk membandingkan hasil kinerja alat tanpa dan menggunakan rangkaian ATmega8.

1. Pengoperasian tanpa rangkaian ATmega8 terhubung pada modul penerima.

Helikopter diterbangkan dengan ketinggian 1-2 meter dan saat helikopter dalam posisi stabil, kemudian pemancar dinonaktifkan.

2. Pengoperasian dengan rangkaian ATmega8 terhubung pada modul penerima.

Helikopter diterbangkan dengan ketinggian 1-2 meter dan saat helikopter dalam posisi stabil, kemudian pemancar dinonaktifkan.

4.4.1 Hasil Pengoperasian

Helikopter yang dioperasikan tanpa terhubung dengan rangkaian ATmega8 terjatuh dengan tiba-tiba, sehingga dapat merusak helikopter tersebut. Tetapi, helikopter yang dioperasikan dengan rangkaian ATmega8 yang terhubung dengan modul penerima akan mendarat secara otomatis pada landasan yang datar. Video dapat dilihat pada alamat: <http://www.youtube.com/watch?v=wUn1IbBwAfI>

http://www.youtube.com/watch?v=SWZfTalPX_c

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Alat ini telah bekerja sesuai dengan yang diinginkan dengan memanfaatkan mikrokontroler ATmega8 sebagai kontrol utama secara otomatis untuk pendaratan saat komunikasi helikopter dengan pemancar terputus.
2. Fungsi sensor jarak ping ultrasonik disini sebagai sensor yang memberikan output pada mikrokontroler dalam menentukan bagaimana dan dimana infrastruktur tempat helikopter mendarat .
3. Simulasi pendaratan darurat pada helikopter mainan ini dapat mendarat dengan baik bila dilakukan saat tidak ada angin atau dalam ruangan, karena angin berpengaruh saat proses pendaratan.

5.2 Saran

1. Sebaiknya alat ini dapat dikembangkan lagi, karena kita dapat menambahkan fitur lainnya seperti penggunaan giroskop elektronik **ADXL 345** yang dapat menyesuaikan kesetimbangan posisi helikopter.
2. Percobaan untuk pengembangan pendaratan alat ini sebaiknya dilakukan dengan helikopter yang daya angkatnya lebih besar sehingga beban sensor dan mikrokontroller tidak begitu mempengaruhi proses pendaratan.
3. Tingkat kesempurnaan pendaratan dapat dikembangkan dengan menambahkan beberapa sensor ping ultrasonik pada bagian badan helikopter dan ekor helikopter untuk mencegah terjadinya benturan pada benda atau pohon disaat pendaratan. Dan diusahakan pin dapat bekerja saat modul penerima hilang komunikasi dari pemancar agar dapat menghemat daya baterai.