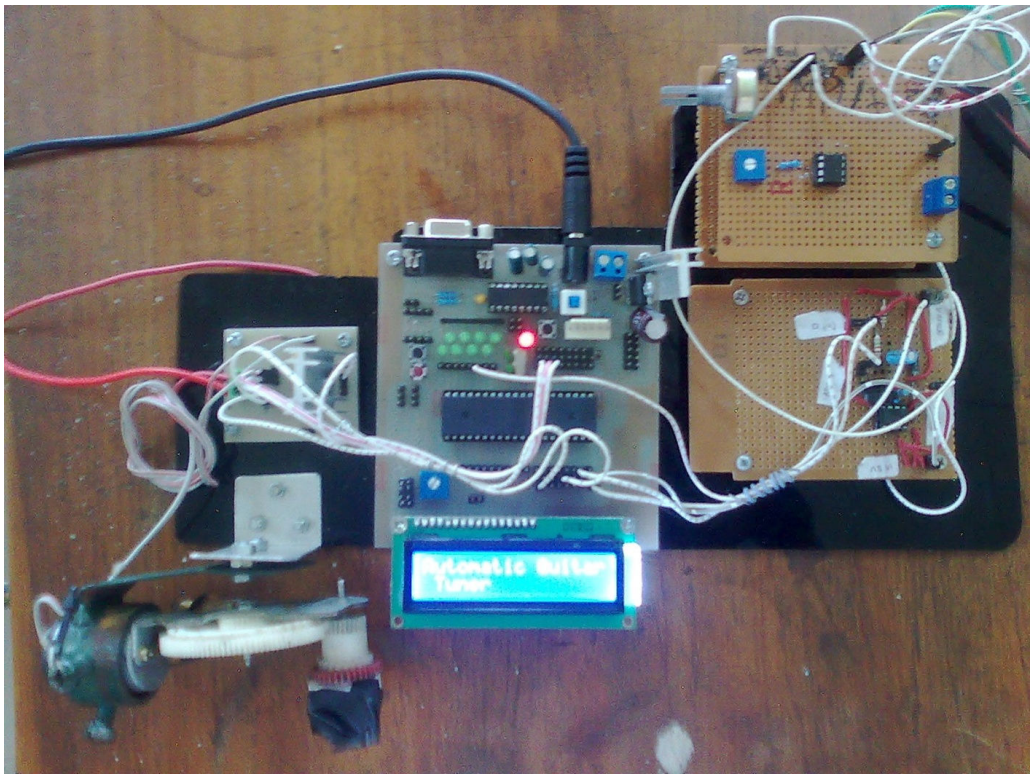
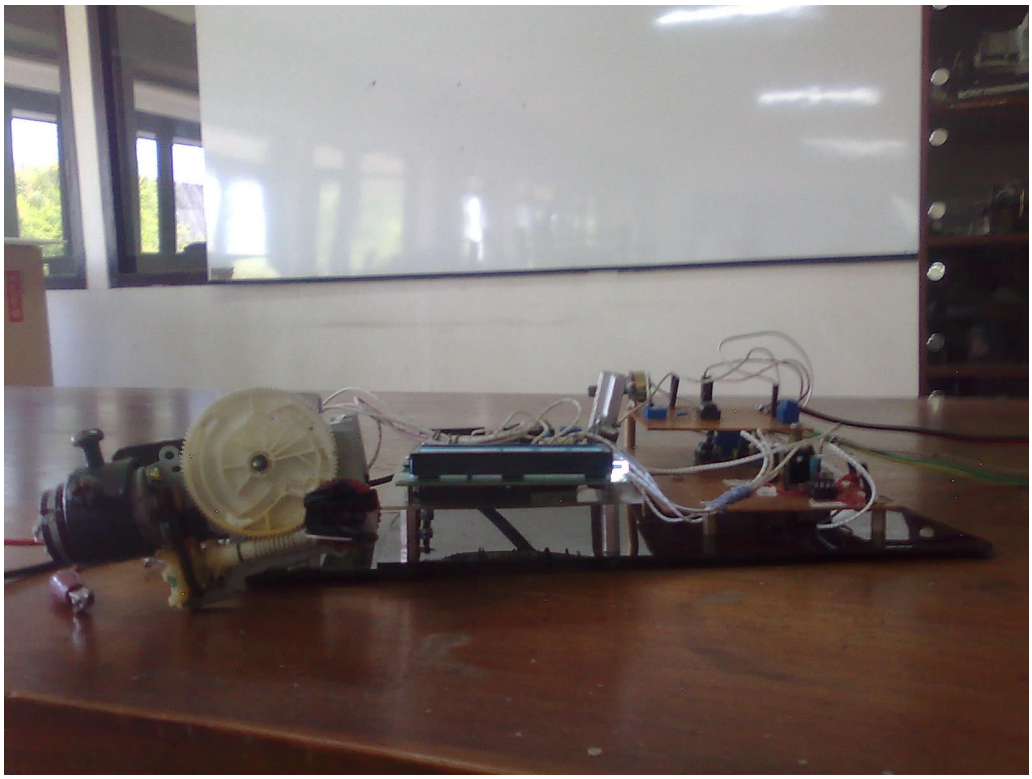
LAMPIRAN A
FOTO PENALA GITAR OTOMATIS

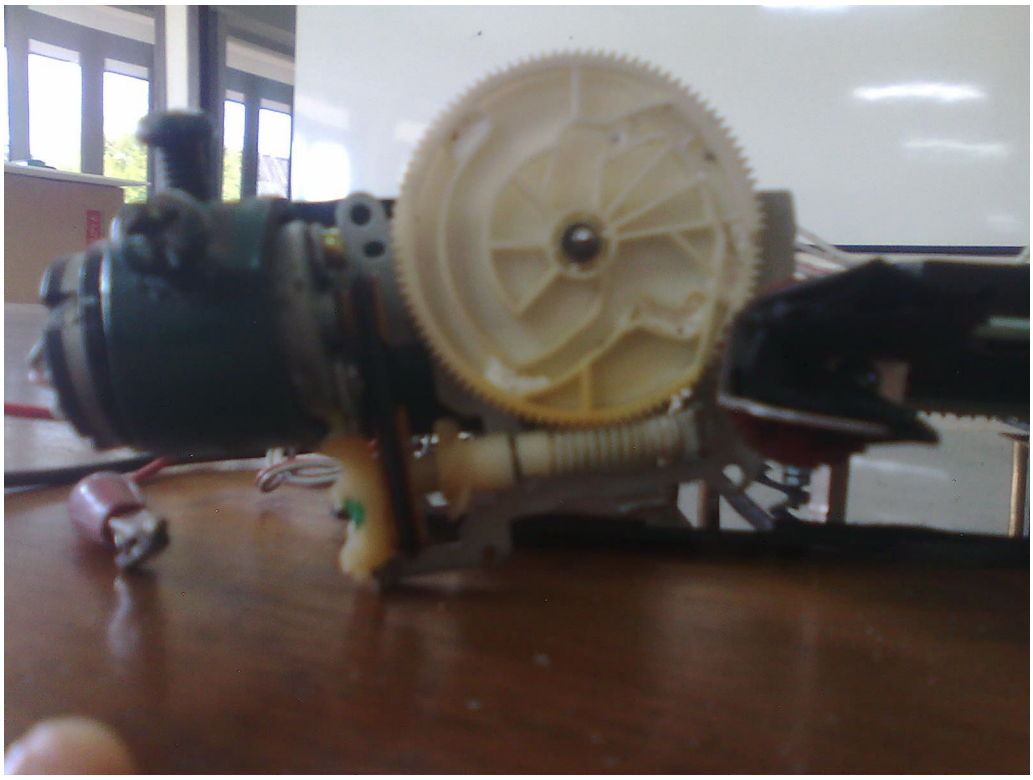
TAMPAK ATAS



TAMPAK SAMPING DEPAN

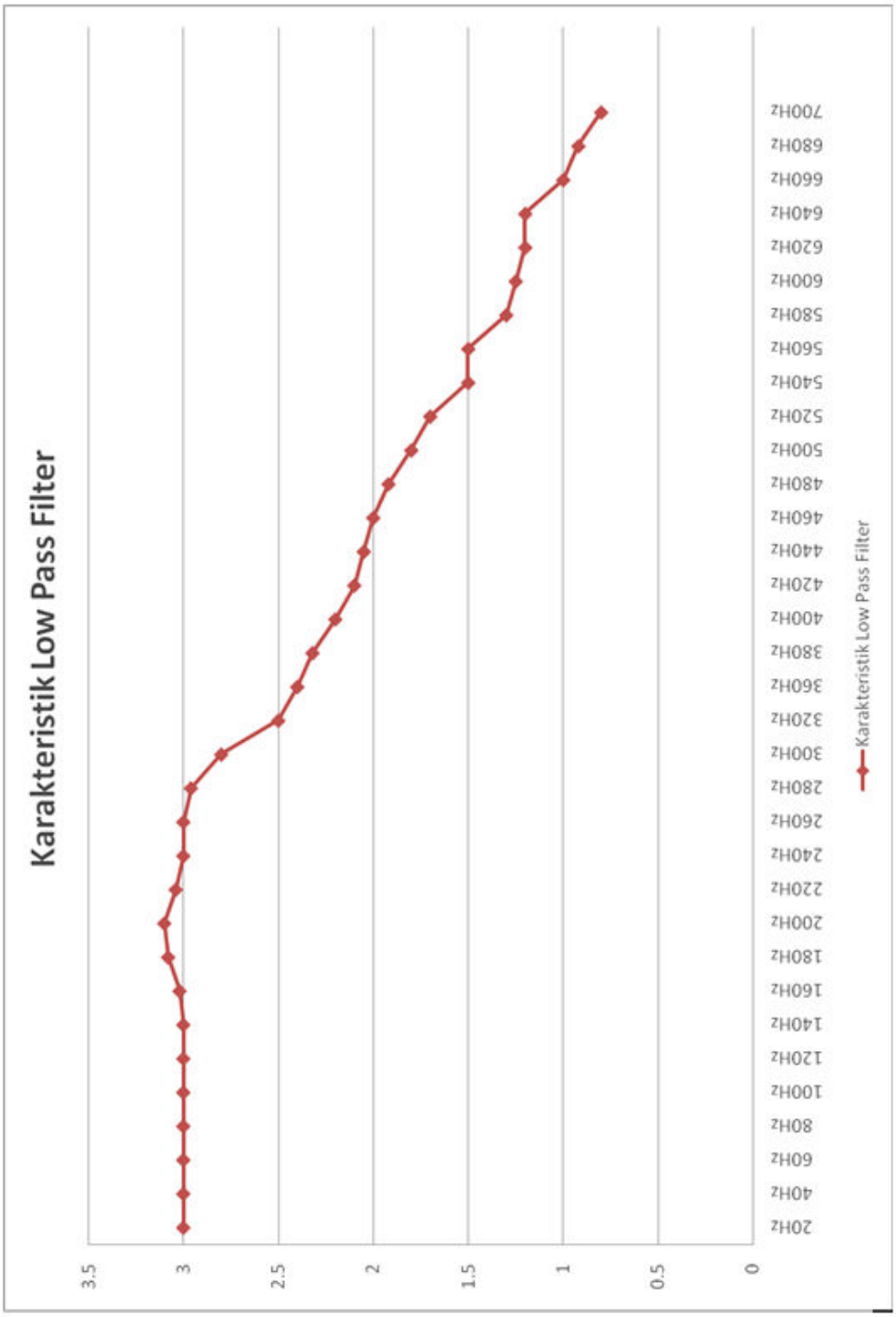


GEAR MOTOR DC TAMPAK DEPAN



GEAR MOTOR DC TAMPAK ATAS

LAMPIRAN B
GAMBAR KARAKTERISTIK LOW PASS FILTER


Karakteristik Low Pass Filter
3.5
3
2.5
2
1.5
1
0.5
0
20Hz
40Hz
60Hz
80Hz
100Hz
120Hz
140Hz
160Hz
180Hz
200Hz
220Hz
240Hz
260Hz
280Hz
300Hz
320Hz
360Hz
380Hz
400Hz
420Hz
440Hz
460Hz
480Hz
500Hz
520Hz
540Hz
560Hz
580Hz
600Hz
620Hz
640Hz
660Hz
680Hz
700Hz
Karakteristik Low Pass Filter

LAMPIRAN C
PROGRAM PADA PENGONTROL MIKRO ATMEGA16

```
/*****************************************************
This program was produced by the
CodeWizardAVR V1.25.3 Standard
Automatic Program Generator
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com

Project :
Version :
Date    : 5/19/2009
Author  : F4CG
Company : F4CG
Comments:


Chip type           : ATmega16
Program type        : Application
Clock frequency     : 11.0592 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/

#include <mega16.h>
#include <delay.h>
#include <stdio.h>
// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
unsigned long int a,frek;
int text[33];



// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{a++;
// Place your code here

}

// Timer 1 output compare A interrupt service routine
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{ frek=a;
a=0;
// Place your code here

}

#define ADC_VREF_TYPE 0xC0

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
```

```
return ADCW;
}

// Declare your global variables here

 float v;
 int adc;
 void baca_frek(void);
 void maju(void);
 void mundur(void);
 void stop(void)  ;

 void main(void)
{

// Declare your local variables her
// Input/Output Ports initialization
// Port A 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
PORTA=0x00;
DDRA=0x00;

// 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
// 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
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
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 15.625 kHz
// Mode: CTC top=OCR1A
// 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: On
// Compare B Match Interrupt: Off
```

```
TCCR1A=0x00;
TCCR1B=0x0D;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x3D;
OCR1AL=0x09;
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: On
// INT0 Mode: Falling Edge
// INT1: Off
// INT2: Off
GICRl=0x40;
MCUCR=0x02;
MCUCSR=0x00;
GIFR=0x40;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x10;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: Int., cap. on AREF
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// LCD module initialization
lcd_init(16);
lcd_putsf("Automatic Guitar Tuner");
delay_ms(1500);
lcd_clear();
// Global enable interrupts
#asm("sei")

// awal:
while (1)
      {
//          if (PINA.4 == 1)
          baca_frek();
//          {
//              if (PINA.5 ==1 && PINA.6 == 0 && PINA.7 == 0)
```

```
//            {

          if (frek == 0)
          {
           lcd_clear();
           lcd_putsf("Belum Ada Input");
              while (frek == 0)
              {
               stop();
              }

           }
          if (frek >=1 && frek <= 79)
           {
           lcd_clear();
           lcd_putsf("E (low) >>");
              while (frek >=1 && frek <= 79)
              {
               maju();
              }

           }


           if (frek >= 80 && frek <= 84)
           {
           lcd_clear();
           lcd_putsf("E (low)");
              while (frek >= 80 && frek <= 84)
              {
               stop();
              }
           }

           if (frek >= 85 && frek <= 96)
           {
           lcd_clear();
           lcd_putsf("E (low) <<");
              while (frek >= 85 && frek <= 96)
              {

               mundur();
               delay_ms(800);
              }
          }


           if (frek >= 97 && frek <= 107)
           {
           lcd_clear();
           lcd_putsf("A >>");
              while (frek >= 97 && frek <= 107)
              {
               maju();
              }

          }


           if (frek >= 108 && frek <= 112)
           {
           lcd_clear();
```

```
lcd_putsf("A");
  while (frek >= 108 && frek <= 112)
  {
  stop();
  }
 }

 if (frek >= 113 && frek <= 129)
 {
 lcd_clear();
 lcd_putsf("A <<");
  while (frek >= 113 && frek <= 129)
  {

  mundur();
  delay_ms(800);
  }
 }

 if (frek >= 130 && frek <= 143)
 {
 lcd_clear();
 lcd_putsf("D >>");
    while (frek >= 130 && frek <= 143)
  {
   maju();
  }

}


 if (frek >= 144 && frek <= 148)
 {
 lcd_clear();
 lcd_putsf("D");
  while (frek >= 144 && frek <= 148)
  {
  stop();
  }
 }


 if (frek >= 149 && frek <= 172)
 {
 lcd_clear();
 lcd_putsf("D <<");
  while (frek >= 149 && frek <= 172)
  {

  mundur();
  delay_ms(800);
  }
 }



  if (frek >= 173 && frek <= 193)
 {
 lcd_clear();
 lcd_putsf("G >>");
    while (frek >= 173 && frek <= 193)
  {
```

```
   maju();
  }

}


if (frek >= 194 && frek <= 198)
{
lcd_clear();
lcd_putsf("G");
  while (frek >= 194 && frek <= 198)/
  {
  stop();
  }
}


if (frek >= 199 && frek <= 222)
{
lcd_clear();
lcd_putsf("G <<");
  while (frek >= 199 && frek <= 222)
  {

  mundur();
  delay_ms(800);
  }
}



if (frek >= 223 && frek <= 243)
{
lcd_clear();
lcd_putsf("B >>");
   while (frek >= 223 && frek <= 243)
  {
   maju();
  }

}


if (frek >= 244 && frek <= 248)
{
lcd_clear();
lcd_putsf("B");
  while (frek >= 244 && frek <= 248)/
  {
  stop();
  }
}


if (frek >= 249 && frek <= 289)
{
lcd_clear();
lcd_putsf("B <<");
  while (frek >= 249 && frek <= 289)
  {

  mundur();
```

```
  delay_ms(800);
  }
 }


if (frek >= 290 && frek <= 326)
 {
 lcd_clear();
 lcd_putsf("E (High) >>");
    while (frek >= 290 && frek <= 326)
  {
   maju();
  }

}


 if (frek >= 327 && frek <= 331)
 {
 lcd_clear();
 lcd_putsf("E (High)");
  while (frek >= 327 && frek <= 331)
  {
  stop();
  }
 }


 if (frek >= 332 && frek <= 800)
 {
 lcd_clear();
 lcd_putsf("E (High) <<");
  while (frek >= 332 && frek <= 800)
  {

  mundur();
  delay_ms(800);
  }
 }
```

```
// fungsi-fungsi

void baca_frek()
{
adc=read_adc(1);
if(frek>=5) v=((float)adc*(5/1.075)/1024);
else  v=((float)adc*5/1024);


      frek=(frek/1.45);

      sprintf(text,"frek = %li \n V = %0.3f V ",frek,v);
```

```
      lcd_puts(text);
      delay_ms(1000);
      lcd_clear();
      }


void maju()
{
PORTB.5=1;
PORTB.6=0;
PORTB.7=1;
}

void mundur()
{
PORTB.5=0;
PORTB.6=1;
PORTB.7=1;
}

void stop()
{
PORTB.5=0;
PORTB.6=0;
PORTB.7=0;
}
```

LAMPIRAN D
DATASHEET