Lampiran A

LISTING PROGRAM

```
/*****************************************************
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    : 8/9/2008
Author  : F4CG
Company : F4CG
Comments:

Chip type           : ATmega8535
Program type        : Application
Clock frequency     : 8.000000 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 128
*****************************************************/
#include <mega8535.h>
#include <delay.h>
#include <stdio.h>
#include <scankeypadB.h>
#include <math.h>
char temp;
int x1,y1,x2,y2,x,y,i,j,k;
bit s1,s2,s3,s4,s5;
// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>

// Timer 0 output compare interrupt service routine
interrupt [TIM0_COMP] void timer0_comp_isr(void)

{
// Place your code here
s1=PINA.0;
s2=PINA.1;
s3=PINA.2;
s4=PINA.3;
s5=PINA.4;
}

void kanan(void)
                {PORTD=0b00110001;}
void kiri(void)
                {PORTD=0b00110100;}
void maju(void)
```

```
        {PORTD=0b00110101;}
void mundur(void)
        {PORTD=0b00111010;}
void stop(void)
        {PORTD=0b00110000;}


// Declare your global variables here
void main(void)
{
// Declare your local variables here
// 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=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTC=0x00;
DDRC=0xFF;

// Port D initialization
// Func7=Out Func6=Out Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0 State1=0 State0=0
PORTD=0x00;
DDRD=0xFF;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 7.813 kHz
// Mode: CTC top=OCR0
// OC0 output: Disconnected
TCCR0=0x0D;
TCNT0=0x00;
OCR0=0x26;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 125.000 kHz
// Mode: Ph. & fr. cor. PWM top=ICR1
// 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=0xA0;
TCCR1B=0x13;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x02;
ICR1L=0x71;
OCR1AH=0x01;
OCR1AL=0x38;
OCR1BH=0x01;
OCR1BL=0x38;

// 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
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x02;

// 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);
// Global enable interrupts
#asm("sei")


while (1)
{
   // Place your code here
   koordinat_satu:
```

```
   temp=scan_keypadB();
   switch (temp)
   {
   case 1:   lcd_putsf("1"); delay_ms(1000); y1=1;break;
   case 2:   lcd_putsf("2"); delay_ms(1000); y1=2; break;
   case 3:   lcd_putsf("3"); delay_ms(1000); y1=3; break;
   case 'A':  lcd_clear(); lcd_putsf("A"); delay_ms(1000); x1=1; break;
   case 'B':  lcd_clear(); lcd_putsf("B"); delay_ms(1000); x1=2; break;
   case 'C':  lcd_clear(); lcd_putsf("C"); delay_ms(1000); x1=3; break;
   case '*': lcd_putsf(" menuju "); delay_ms(1000); goto koordinat_dua; break;
   case '#': lcd_clear();lcd_putsf("clear");delay_ms(3000);lcd_clear();lcd_putsf(""); goto koordinat_satu; break;
   }
   delay_ms(1);
   goto koordinat_satu;

 //================================================================================
//=================================================================================
//=================================================================================

   koordinat_dua:
   temp=scan_keypadB();

   switch (temp)
   {
   case 1:    lcd_putsf("1"); delay_ms(1000); y2=1; break;
   case 2:    lcd_putsf("2"); delay_ms(1000); y2=2; break;
   case 3:    lcd_putsf("3"); delay_ms(1000); y2=3; break;
   case 'A':  lcd_putsf("A"); delay_ms(1000); x2=1; break;
   case 'B':  lcd_putsf("B"); delay_ms(1000); x2=2;  break;
   case 'C':  lcd_putsf("C"); delay_ms(1000); x2=3;  break;
   case 4: lcd_clear();lcd_putsf("GO!!!"); goto arah_imajiner_positif; break;
   case 5: lcd_clear();lcd_putsf("GO!!!"); goto arah_imajiner_negatif; break;
   case 6: lcd_clear();lcd_putsf("GO!!!"); goto arah_real_positif; break;
   case 7: lcd_clear();lcd_putsf("GO!!!"); goto arah_real_negatif; break;
   case '#':lcd_clear();lcd_putsf("clear");delay_ms(3000);lcd_clear();lcd_putsf(""); goto koordinat_satu; break;
   }
   delay_ms(1);
   goto koordinat_dua;

 //=================================================================================
//=================================================================================
//=================================================================================
//=================================================================================
//=================================================================================

  arah_imajiner_positif:

  delay_ms(1000);
  x=x1-x2;
  y=y1-y2;
  goto sistem_gerak;
 //===========================================================================
//===========================================================================

  arah_imajiner_negatif:

  delay_ms(1000);
  x=x2-x1;
  y=y2-y1;
  goto sistem_gerak;

 //===========================================================================
//===========================================================================

  arah_real_positif:

  delay_ms(1000);
  x=y2-y1;
  y=x1-x2;
  goto sistem_gerak;

 //===========================================================================
//===========================================================================


  arah_real_negatif:

  delay_ms(1000);
  x=y1-y2;
  y=x2-x1;
  goto sistem_gerak;


 //===========================================================================
//===========================================================================
//===========================================================================
//===========================================================================

 sistem_gerak:

if(y<0)
{
                if(y<0)
                {
                 label_satu:
                 maju();
                              if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
                                 {goto counter_satu;}
                 goto label_satu;
                }
               counter_satu:

               PORTD=0b00110101;
               delay_ms(10);

                counter_satu_satu:

                i=0;
if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0)
  {
                i=i+1;
                y=abs(y)-i;

               if(y==0)
               {goto satu;}

                perintah_satu:
          if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0)
             {
               PORTD=0b00110101;
               if((s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0 && s3==0 && s4==1 &&
               s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 && s5==1) || (s1==0 && s2==1 && s3==1 &&
               s4==1 && s5==1) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 &&
               s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 && s3==1 && s4==0 && s5==0) || (s1==1 &&
               s2==1 && s3==1 && s4==1 && s5==0))
                {goto counter_satu;}
                goto perintah_satu;
               }
  }


 if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
  {OCR1AH=0x01; OCR1AL=0x38; OCR1BH=0x01; OCR1BL=0x38;}

 if(s1==0 && s2==0 && s3==0 && s4==1 && s5==1)
  {OCR1AH=0x00; OCR1AL=0xfa; OCR1BH=0x01; OCR1BL=0x38;}

 if(s1==0 && s2==0 && s3==1 && s4==1 && s5==1)
  {OCR1AH=0x00; OCR1AL=0xbb; OCR1BH=0x00; OCR1BL=0xfa;}

 if(s1==0 && s2==1 && s3==1 && s4==1 && s5==1)
  {OCR1AH=0x00; OCR1AL=0x7d; OCR1AH=0x00; OCR1AL=0xbb;}

 one:
 if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
  {          OCR1AH=0x00; OCR1AL=0x7d; OCR1AH=0x00; OCR1AL=0xbb;
                if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
                  {goto one;}
               if((s1==0 && s2==1 && s3==1 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 &&
               s5==1) || (s1==0 && s2==0 && s3==0 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==0 &&
               s4==0 && s5==1) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0 &&
               s3==0 && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==0) || (s1==1 &&
               s2==1 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 && s3==1 && s4==0 && s5==0) ||
               (s1==1 && s2==1 && s3==1 && s4==1 && s5==0))
                 {goto counter_satu_satu;}
 }

 if(s1==1 && s2==1 && s3==0 && s4==0 && s5==0)
  {OCR1AH=0x01; OCR1AL=0x38; OCR1BH=0x00; OCR1BL=0xfa;}

 if(1==1 && s2==1 && s3==1 && s4==0 && s5==0)
 {OCR1AH=0x00; OCR1AL=0xfa;  OCR1BH=0x00; OCR1BL=0xbb;}

 if(s1==1 && s2==1 && s3==1 && s4==1 && s5==0)
 {OCR1AH=0x00; OCR1AL=0xbb; OCR1BH=0x00; OCR1BL=0x7d;}

  two:
 if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
  {            OCR1AH=0x00; OCR1AL=0xbb;  OCR1AH=0x00; OCR1AL=0x7d;
               if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
                {goto two;}
                if((s1==1 && s2==1 && s3==1 && s4==1 && s5==0) ||  (s1==1 && s2==1 && s3==1 && s4==0
               && s5==0) || (s1==1 && s2==1 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 &&
               s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0 &&
               s3==0 && s4==0 && s5==0) || (s1==0 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 &&
               s2==0 && s3==0 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 && s5==1) ||
               (s1==0 && s2==1 && s3==1 && s4==1 && s5==1))
                 {goto counter_satu_satu;}
     }
  goto counter_satu_satu;

 //===================================================================================
//===================================================================================
//===================================================================================
  satu:

        if(x<0)
          {
                sumbu_x_satu_satu:
               kanan();
                if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
                  {goto counter_dua;}
               goto sumbu_x_satu_satu;
          }

//===================================================================================

          if(x>0)
            {
                 sumbu_x_satu_dua:
                 kiri();
                  if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
                    {goto counter_dua;}
                 goto sumbu_x_satu_dua;
            }
```

```
      if(x==0)
        {
                    stop();lcd_clear();lcd_putsf("FINISH");
                    delay_ms(5000);lcd_clear();lcd_putsf("");
                    goto koordinat_satu;
        }


counter_dua:

PORTD=0b00110101;
delay_ms(10);

counter_dua_satu:

              j=0;
              if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0)
{
                  j=j+1;
                 x=abs(x)-j;

                if(x==0)
                 {
                 stop();lcd_clear();lcd_putsf("FINISH");
                 delay_ms(5000);lcd_clear();lcd_putsf("");
                 goto koordinat_satu;
                 }

               perintah_dua:

              if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0)
         {
             PORTD=0b00110101;
              if((s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0 && s3==0 && s4==1 &&
             s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 && s5==1) || (s1==0 && s2==1 && s3==1 &&
             s4==1 && s5==1) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 &&
             s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 && s3==1 && s4==0 && s5==0) || (s1==1 &&
             s2==1 && s3==1 && s4==1 && s5==0))
                {goto counter_dua;}
              goto perintah_dua;
          }
}


if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
{OCR1AH=0x01; OCR1AL=0x38; OCR1BH=0x01; OCR1BL=0x38;}

if(s1==0 && s2==0 && s3==0 && s4==1 && s5==1)
{OCR1AH=0x00; OCR1AL=0xfa; OCR1BH=0x01; OCR1BL=0x38;}

if(s1==0 && s2==0 && s3==1 && s4==1 && s5==1)
{OCR1AH=0x00; OCR1AL=0xbb; OCR1BH=0x00; OCR1BL=0xfa;}
```

```
 if(s1==0 && s2==1 && s3==1 && s4==1 && s5==1)
 {OCR1AH=0x00; OCR1AL=0x7d; OCR1AH=0x00; OCR1AL=0xbb;}

 three:
 if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
   {OCR1AH=0x00; OCR1AL=0x7d; OCR1AH=0x00; OCR1AL=0xbb;
                if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
                  {goto three;}
                if((s1==0 && s2==1 && s3==1 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 &&
               s5==1) || (s1==0 && s2==0 && s3==0 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==0 &&
               s4==0 && s5==1) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0 &&
               s3==0 && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==0) || (s1==1 &&
               s2==1 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 && s3==1 && s4==0 && s5==0) ||
               (s1==1 && s2==1 && s3==1 && s4==1 && s5==0))
                  {goto counter_dua_satu;}
    }

 if(s1==1 && s2==1 && s3==0 && s4==0 && s5==0)
 {OCR1AH=0x01;  OCR1AL=0x38; OCR1BH=0x00;  OCR1BL=0xfa;}

 if(1==1 && s2==1 && s3==1 && s4==0 && s5==0)
 {OCR1AH=0x00; OCR1AL=0xfa;  OCR1BH=0x00; OCR1BL=0xbb;}

 if(s1==1 && s2==1 && s3==1 && s4==1 && s5==0)
 {OCR1AH=0x00; OCR1AL=0xbb;  OCR1BH=0x00; OCR1BL=0x7d;}

  four:
 if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
   {OCR1AH=0x00; OCR1AL=0xbb;  OCR1AH=0x00;  OCR1AL=0x7d;
                if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
                  {goto four;}
                if((s1==1 && s2==1 && s3==1 && s4==1 && s5==0) ||  (s1==1 && s2==1 && s3==1 && s4==0
               && s5==0) || (s1==1 && s2==1 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 &&
               s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0 &&
               s3==0 && s4==0 && s5==0) || (s1==0 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 &&
               s2==0 && s3==0 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 && s5==1) ||
               (s1==0 && s2==1 && s3==1 && s4==1 && s5==1))
               {goto counter_dua_satu;}
    }
 goto counter_dua_satu;

}

 //==============================================================================
//==============================================================================
//==============================================================================

if(y==0)
  {
    stop();
    goto satu;
  }
```

```
//=============================================================================
//=============================================================================
if(y>0 && x<0)
{

        if(y>0 && x<0)
        {
        sumbu_x_satu_tiga:
        kanan();
        if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
          {goto counter_tiga;}
        goto sumbu_x_satu_tiga;
        }

//=============================================================================
//=============================================================================
//=============================================================================
//=============================================================================

 counter_tiga:

   PORTD=0b00110101;
   delay_ms(10);

 counter_tiga_satu:

                j=0;
                if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0)
            {
                 j=j+1;
                  x=abs(x)-j;

                  if(x==0)
                  {goto dua;}

                  perintah_tiga:
                  if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0)
                    {PORTD=0b00110101;
                   if((s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0 && s3==0 && s4==1
                  && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 && s5==1) || (s1==0 && s2==1 && s3==1
                  && s4==1 && s5==1) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1
                  && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 && s3==1 && s4==0 && s5==0) || (s1==1
                  && s2==1 && s3==1 && s4==1 && s5==0))
                       {goto counter_tiga;}
                   goto perintah_tiga;
                    }
            }

  if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
 {OCR1AH=0x01; OCR1AL=0x38; OCR1BH=0x01; OCR1BL=0x38;}

 if(s1==0 && s2==0 && s3==0 && s4==1 && s5==1)
 {OCR1AH=0x00; OCR1AL=0xfa;  OCR1BH=0x01; OCR1BL=0x38;}
```

```
  if(s1==0 && s2==0 && s3==1 && s4==1 && s5==1)
{OCR1AH=0x00; OCR1AL=0xbb; OCR1BH=0x00; OCR1BL=0xfa;}

if(s1==0 && s2==1 && s3==1 && s4==1 && s5==1)
{OCR1AH=0x00; OCR1AL=0x7d; OCR1AH=0x00; OCR1AL=0xbb;}

                five:
                if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
                {
                 OCR1AH=0x00; OCR1AL=0x7d; OCR1AH=0x00; OCR1AL=0xbb;
                  if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
                    {goto five;}
                   if((s1==0 && s2==1 && s3==1 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1
                  && s5==1) || (s1==0 && s2==0 && s3==0 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==0
                  && s4==0 && s5==1) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0
                  && s3==0 && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==0) || (s1==1
                  && s2==1 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 && s3==1 && s4==0 && s5==0)
                  || (s1==1 && s2==1 && s3==1 && s4==1 && s5==0))
                 {goto counter_tiga_satu;}
                }

if(s1==1 && s2==1 && s3==0 && s4==0 && s5==0)
{OCR1AH=0x01;  OCR1AL=0x38; OCR1BH=0x00; OCR1BL=0xfa;}

if(1==1 && s2==1 && s3==1 && s4==0 && s5==0)
{OCR1AH=0x00; OCR1AL=0xfa; OCR1BH=0x00; OCR1BL=0xbb;}

if(s1==1 && s2==1 && s3==1 && s4==1 && s5==0)
{OCR1AH=0x00; OCR1AL=0xbb; OCR1BH=0x00; OCR1BL=0x7d;}

                six:
                if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
             {
                OCR1AH=0x00; OCR1AL=0xbb; OCR1AH=0x00;  OCR1AL=0x7d;
                  if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
                    {goto six;}
                   if((s1==1 && s2==1 && s3==1 && s4==1 && s5==0) ||  (s1==1 && s2==1 && s3==1 && s4==0
                  && s5==0) || (s1==1 && s2==1 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0
                  && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0
                  && s3==0 && s4==0 && s5==0) || (s1==0 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0
                  && s2==0 && s3==0 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 && s5==1)
                  || (s1==0 && s2==1 && s3==1 && s4==1 && s5==1))
                 {goto counter_tiga_satu;}
             }

 goto counter_tiga_satu;

//================================================================================
//================================================================================
//================================================================================
//================================================================================
                dua:

                 kanan();
                 if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
                    {goto counter_lima;}
                  goto dua;
}

 //==========================================================================
//==========================================================================
//==========================================================================
//==========================================================================

 if(y>0 && x>0)
{
       if(y>0 && x>0)
       {
       sumbu_x_satu_empat:
       kiri();
       if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
         {goto counter_empat;}
       goto sumbu_x_satu_empat;
       }

//==========================================================================
//==========================================================================

 counter_empat:

  PORTD=0b00110101;
  delay_ms(10);

  counter_empat_satu:

               k=0;
               if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0)
           {
                 k=k+1;
                 x=abs(x)-k;

                 if(x==0)
                 {goto tiga;}

                 perintah_empat:
                 if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0)
                   {
                PORTD=0b00110101;
                  if((s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0 && s3==0 && s4==1
                 && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 && s5==1) || (s1==0 && s2==1 && s3==1
                 && s4==1 && s5==1) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1
                 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 && s3==1 && s4==0 && s5==0) || (s1==1
                 && s2==1 && s3==1 && s4==1 && s5==0))
                      {goto counter_empat;}
                  goto perintah_empat;
                      }
              }


 if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
 {OCR1AH=0x01;  OCR1AL=0x38;  OCR1BH=0x01; OCR1BL=0x38;}

 if(s1==0 && s2==0 && s3==0 && s4==1 && s5==1)
 {OCR1AH=0x00; OCR1AL=0xfa; OCR1BH=0x01; OCR1BL=0x38;}

 if(s1==0 && s2==0 && s3==1 && s4==1 && s5==1)
 {OCR1AH=0x00; OCR1AL=0xbb; OCR1BH=0x00; OCR1BL=0xfa;}

 if(s1==0 && s2==1 && s3==1 && s4==1 && s5==1)
 {OCR1AH=0x00; OCR1AL=0x7d; OCR1AH=0x00; OCR1AL=0xbb;}

               seven:
                 if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
               {
                  OCR1AH=0x00; OCR1AL=0x7d; OCR1AH=0x00; OCR1AL=0xbb;
                  if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
                  {goto seven;}
                  if((s1==0 && s2==1 && s3==1 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1
                 && s5==1) || (s1==0 && s2==0 && s3==0 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==0
                 && s4==0 && s5==1) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0
                 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==0) || (s1==1
                 && s2==1 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 && s3==1 && s4==0 && s5==0)
                 || (s1==1 && s2==1 && s3==1 && s4==1 && s5==0))
                 {goto counter_empat_satu;}
               }

 if(s1==1 && s2==1 && s3==0 && s4==0 && s5==0)
 {OCR1AH=0x01; OCR1AL=0x38; OCR1BH=0x00;  OCR1BL=0xfa;}

 if(1==1 && s2==1 && s3==1 && s4==0 && s5==0)
 {OCR1AH=0x00; OCR1AL=0xfa; OCR1BH=0x00; OCR1BL=0xbb;}

 if(s1==1 && s2==1 && s3==1 && s4==1 && s5==0)
 {OCR1AH=0x00; OCR1AL=0xbb; OCR1BH=0x00;  OCR1BL=0x7d;}

                eight:
                 if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
             {
               OCR1AH=0x00;  OCR1AL=0xbb; OCR1AH=0x00; OCR1AL=0x7d;
                  if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
                    {goto eight;}
                   if((s1==1 && s2==1 && s3==1 && s4==1 && s5==0) ||  (s1==1 && s2==1 && s3==1 && s4==0
                 && s5==0) || (s1==1 && s2==1 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0
                 && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0
                 && s3==0 && s4==0 && s5==0) || (s1==0 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0
                 && s2==0 && s3==0 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 && s5==1)
                 || (s1==0 && s2==1 && s3==1 && s4==1 && s5==1))
```

```
                   {goto counter_empat_satu;}
                }

                   goto counter_empat_satu;


//==========================================================================
//==========================================================================
//==========================================================================
//==========================================================================
       tiga:

       kiri();
       if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
       {goto counter_lima;}
       goto tiga;
}

//==========================================================================
//==========================================================================
//==========================================================================

if(y>0 && x==0)
{

 tahap_satu:
 kanan();
 if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
   {goto tahap_dua;}
 goto tahap_satu;

 tahap_dua:
 mundur();
 if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0)
   {goto tahap_tiga;}
 goto tahap_dua;

 tahap_tiga:
 kanan();
 if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
   {goto counter_lima;}
 goto tahap_tiga;

}

//==========================================================================
//==========================================================================
//==========================================================================
counter_lima:

   PORTD=0b00110101;
   delay_ms(10);

                  counter_lima_satu:

                i=0;
                if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0)
            {
                   i=i+1;
                   y=abs(y)-i;

                  if(y==0)
                  {
                   stop();lcd_clear();lcd_putsf("FINISH");
                   delay_ms(5000);lcd_clear();lcd_putsf("");
                   goto koordinat_satu;
                  }

                  perintah_lima:
                  if(s1==0 && s2==0 && s3==0 && s4==0 && s5==0)
                   {PORTD=0b00110101;
                   if((s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0 && s3==0 && s4==1
                  && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 && s5==1) || (s1==0 && s2==1 && s3==1
                  && s4==1 && s5==1) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1
                  && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 && s3==1 && s4==0 && s5==0) || (s1==1
                  && s2==1 && s3==1 && s4==1 && s5==0))
                        {goto counter_lima;}
                     goto perintah_lima;
                   }
              }


 if(s1==1 && s2==0 && s3==0 && s4==0 && s5==1)
 {OCR1AH=0x01;  OCR1AL=0x38; OCR1BH=0x01; OCR1BL=0x38;}

 if(s1==0 && s2==0 && s3==0 && s4==1 && s5==1)
 {OCR1AH=0x00; OCR1AL=0xfa; OCR1BH=0x01; OCR1BL=0x38;}

 if(s1==0 && s2==0 && s3==1 && s4==1 && s5==1)
 {OCR1AH=0x00; OCR1AL=0xbb; OCR1BH=0x00; OCR1BL=0xfa;}

 if(s1==0 && s2==1 && s3==1 && s4==1 && s5==1)
 {OCR1AH=0x00; OCR1AL=0x7d; OCR1AH=0x00; OCR1AL=0xbb;}

                nine:
                if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
             {
                  OCR1AH=0x00; OCR1AL=0x7d; OCR1AH=0x00; OCR1AL=0xbb;
                  if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
                   {goto nine;}
                 if((s1==0 && s2==1 && s3==1 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1
                && s5==1) || (s1==0 && s2==0 && s3==0 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==0
                && s4==0 && s5==1) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0
                && s3==0 && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==0) || (s1==1
                && s2==1 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==1 && s3==1 && s4==0 && s5==0)
                || (s1==1 && s2==1 && s3==1 && s4==1 && s5==0))
                    {goto counter_lima_satu;}
            }

    if(s1==1 && s2==1 && s3==0 && s4==0 && s5==0)
    {OCR1AH=0x01; OCR1AL=0x38; OCR1BH=0x00; OCR1BL=0xfa;}

    if(1==1 && s2==1 && s3==1 && s4==0 && s5==0)
    {OCR1AH=0x00; OCR1AL=0xfa; OCR1BH=0x00; OCR1BL=0xbb;}

    if(s1==1 && s2==1 && s3==1 && s4==1 && s5==0)
    {OCR1AH=0x00; OCR1AL=0xbb; OCR1BH=0x00; OCR1BL=0x7d;}

                ten:
               if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
            {
                OCR1AH=0x00; OCR1AL=0xbb; OCR1AH=0x00;  OCR1AL=0x7d;
                if(s1==1 && s2==1 && s3==1 && s4==1 && s5==1)
                   {goto ten;}
                 if((s1==1 && s2==1 && s3==1 && s4==1 && s5==0) ||  (s1==1 && s2==1 && s3==1 && s4==0
                && s5==0) || (s1==1 && s2==1 && s3==0 && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0
                && s4==0 && s5==0) || (s1==1 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0 && s2==0
                && s3==0 && s4==0 && s5==0) || (s1==0 && s2==0 && s3==0 && s4==0 && s5==1) || (s1==0
                && s2==0 && s3==0 && s4==1 && s5==1) || (s1==0 && s2==0 && s3==1 && s4==1 && s5==1)
                || (s1==0 && s2==1 && s3==1 && s4==1 && s5==1))
                    {goto counter_lima_satu;}
            }

                goto counter_lima_satu;

    };
    }
```

Lampiran B

DATASHEET MIKROKONTROLER

8-bit AVR® Microcontroller with 8K Bytes In-System Programmable Flash

ATmega8535
ATmega8535L

Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 130 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
  - 8K Bytes of In-System Self-Programmable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 512 Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 512 Bytes Internal SRAM
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    8 Single-ended Channels
    7 Differential Channels for TQFP Package Only
    2 Differential Channels with Programmable Gain at 1x, 10x, or 200x for TQFP Package Only
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, 44-lead PLCC, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega8535L
  - 4.5 - 5.5V for ATmega8535
- Speed Grades
  - 0 - 8 MHz for ATmega8535L
  - 0 - 16 MHz for ATmega8535

Pin Configurations

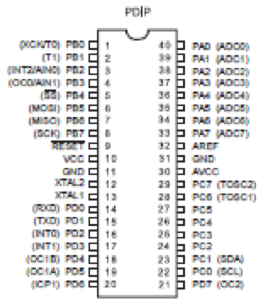
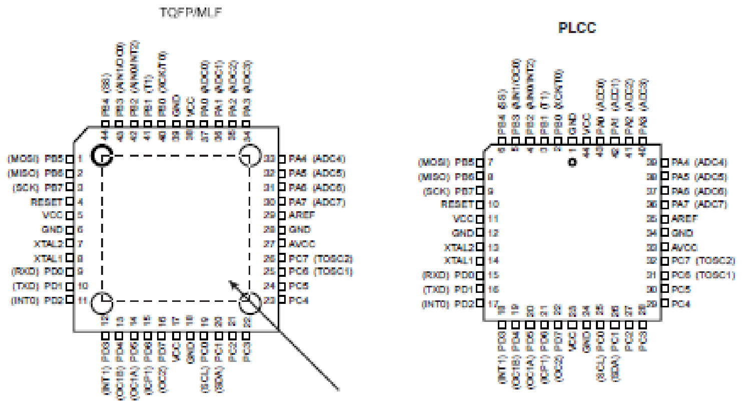



Figure 1. Pinout ATmega8535

NOTE: MLF Bottom pad should be soldered to ground.

Disclaimer

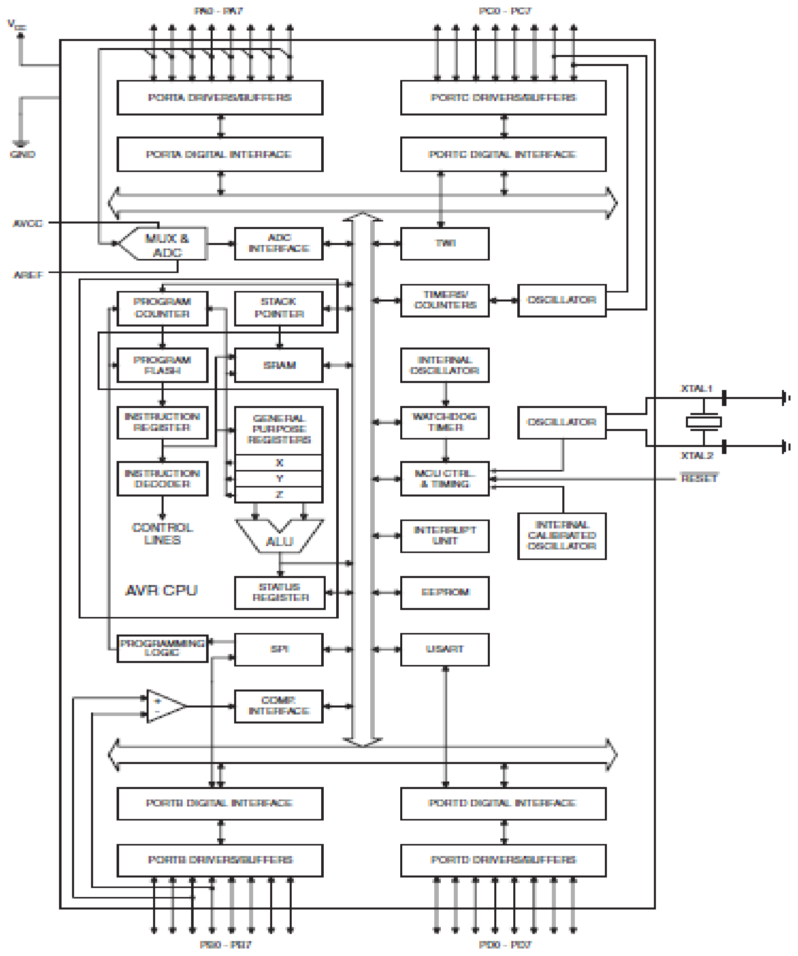
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega8535 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing instructions in a single clock cycle, the ATmega8535 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega8535 provides the following features: 8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 512 bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain in TQFP package, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the asynchronous timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega8535 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega8535 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, In-Circuit Emulators, and evaluation kits.

AT90S8535 Compatibility

The ATmega8535 provides all the features of the AT90S8535. In addition, several new features are added. The ATmega8535 is backward compatible with AT90S8535 in most cases. However, some incompatibilities between the two microcontrollers exist. To solve this problem, an AT90S8535 compatibility mode can be selected by programming the S8535C fuse. ATmega8535 is pin compatible with AT90S8535, and can replace the AT90S8535 on current Printed Circuit Boards. However, the location of fuse bits and the electrical characteristics differs between the two devices.

AT90S8535 Compatibility Mode

Programming the S8535C fuse will change the following functionality:

- The timed sequence for changing the Watchdog Time-out period is disabled. See "Timed Sequences for Changing the Configuration of the Watchdog Timer" on page 45 for details.
- The double buffering of the USART Receive Register is disabled. See "AVR USART vs. AVR UART – Compatibility" on page 146 for details.

Pin Descriptions

V_{CC}

Digital supply voltage.

GND

Ground.

Port A (PA7..PA0)

Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega8535 as listed on page 60.

Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega8535 as listed on page 64.

$\overline{\text{RESET}}$

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 37. Shorter pulses are not guaranteed to generate a reset.

XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting Oscillator amplifier.

AVCC

AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

AREF

AREF is the analog reference pin for the A/D Converter.

Lampiran C

DATASHEET HAMAMATSU P5587

Photoreflector
P5587, P5588

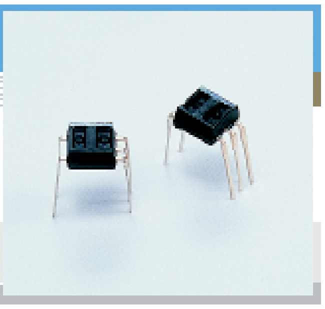


Photo IC output (digital) photoreflectors

P5587 and P5588 are photoreflectors combining a high power infrared LED and low voltage photo IC. The photo IC consists of a high sensitivity photodiode, amplifier, schmitt trigger circuit, and output phototransistor, etc. on a single chip.

Features

- Miniature package
- Low voltage operation
- Photo IC, open collector output
- P5587: "H" level output at light input
  P5588: "L" level output at light input

Applications

- Paper detection in copiers and printers, etc.
- Tape end detection in VTRs, tape recorders, etc.

■ Absolute maximum ratings (Ta=25 °C)

Parameter		Symbol	Value	Unit
Input (LED)	Forward current	I_F	50	mA
	Reverse voltage	V_R Max.	5	V
	Power dissipation	P	80	mW
Output (photo IC)	Supply voltage	Vcc	-0.5 to +7	V
	Output voltage	Vo	-0.5 to +7	V
	Output current	Io	8	mA
	Power dissipation	P	80	mW
Operating temperature		Topr	-25 to +85	°C
Storage temperature		Tstg	-30 to +85	°C
Soldering		-	260 °C, 3 s, refer to Dimensional outline	-

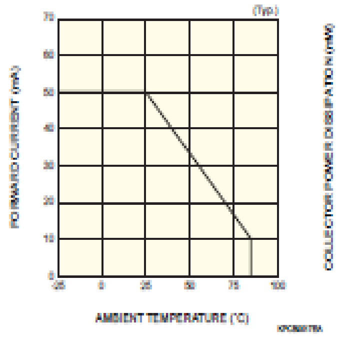
■ Electrical and optical characteristics (Ta=25 °C, Vcc=5 V, unless otherwise noted)

Parameter		Symbol	Condition	P5587			P5588			Unit
				Min.	Typ.	Max.	Min.	Typ.	Max.	
Input (LED)	Forward voltage	V_F	I_F=20 mA	-	1.23	1.45	-	1.23	1.45	V
	Reverse current	I_R	V_R=5 V	-	-	10	-	-	10	μA
	Terminal capacitance	Ct	V=0 V, f=1 MHz	-	30	-	-	30	-	pF
Output (photo IC)	Supply voltage	Vcc		2.2	-	7	2.2	-	7	V
	Low level output voltage	V_{OL}	I_{OL}=4 mA [*1]	-	0.1	0.4	-	0.1	0.4	V
	High level output current	I_{OH}	V_O=5 V [*2]	-	-	10	-	-	10	μA
	Current consumption	Icc		-	1.3	3.0	-	1.3	3.0	mA
Transfer characteristics	L→H Threshold input current	I_{FLH}	R_L=1.2 kΩ, d=3 mm Reflecting surface:	-	-	10	-	-	-	mA
	H→L Threshold input current	I_{FHL}	white paper (reflectivity 90 % or more)	-	-	-	-	-	10	mA
	Hysterisis	-	[*3]	-	0.8	-	-	0.8	-	-
	L→H Propagation delay time	t_{PLH}	I_F=15 mA R_L=1.2 kΩ d=3 mm	-	-	20	-	-	30	μs
	H→L Propagation delay time	t_{PHL}		-	-	30	-	-	20	μs
	Rise time	tr		-	0.07	-	-	0.07	-	μs
	Fall time	tf		-	0.03	-	-	0.03	-	μs

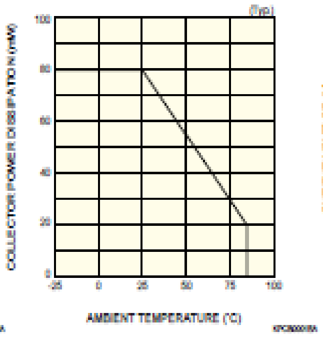
[*1]: P5587: I_F=0 mA, P5588: I_F=15 mA
[*2]: P5587: I_F=15 mA, P5588: I_F=0 mA
[*3]: P5587: I_{FHL}/I_{FLH}, P5588: I_{FLH}/I_{FHL}

Note) Connect a 0.01 μF capacitor or larger between Vcc and GND.

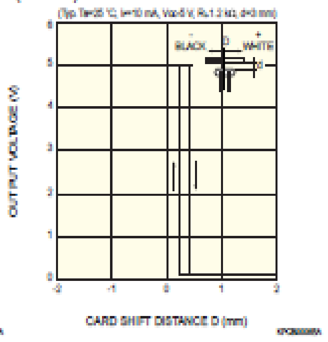
■ LED forward current vs. ambient temperature



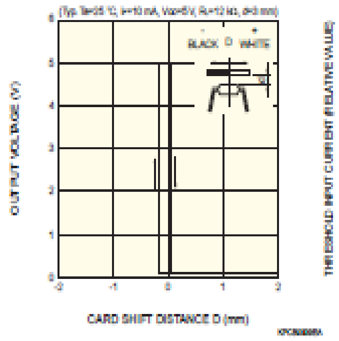
■ Photo IC power dissipation vs. ambient temperature



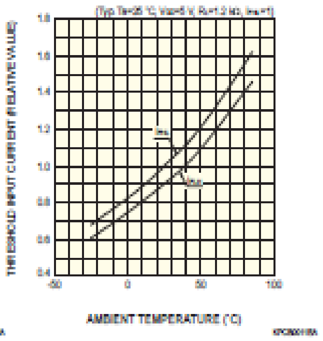
■ Position detection characteristic (P5587)



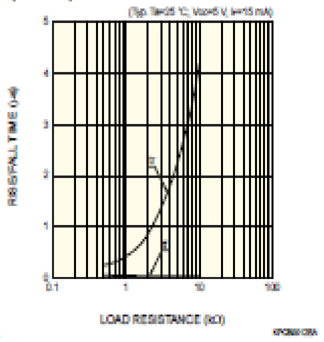
■ Position detection characteristic (P5587)



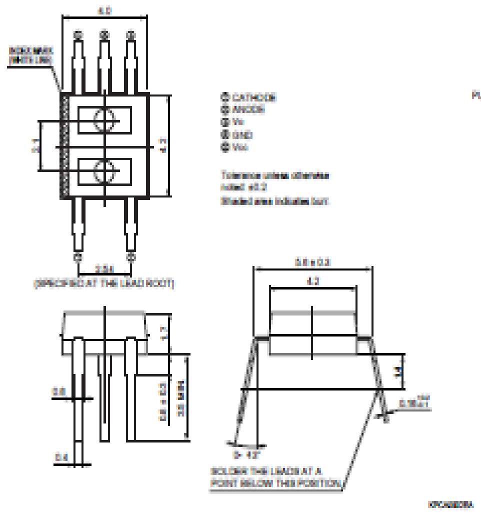
■ Threshold input current vs. ambient temperature (P5587)



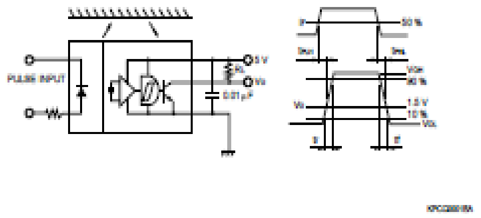
■ Rise/fall time vs. load resistance (P5587)



■ Dimensional outline (unit: mm)


4.0
INDEX MARK (WHITE LINE)
① CATHODE
② ANODE
③ Vo
④ GND
⑤ Vcc
Tolerance unless otherwise noted: ±0.2
Shaded area indicates burr.
2.54
(SPECIFIED AT THE LEAD ROOT)
4.3
5.6 ± 0.3
4.2
1.7
0.8
0.4
3.5 MIN.
0° - 4.5°
SOLDER THE LEADS AT A POINT BELOW THIS POSITION

■ Response time measurement circuit (P5587)



Lampiran D

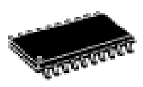
DATASHEET IC L293D

SGS-THOMSON MICROELECTRONICS

L293D
L293DD

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES



SO(12+4+4) Powerdip (12+2+2)

ORDERING NUMBERS:

L293DD L293D

DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoides, DC and stepping motors) and switching power transistors.

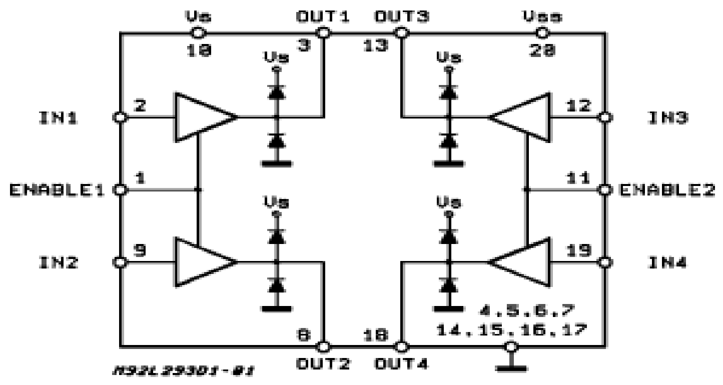
To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

This device is suitable for use in switching applications at frequencies up to 5 kHz.

The L293D is assembled in a 16 lead plastic packaage which has 4 center pins connected together and used for heatsinking

The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

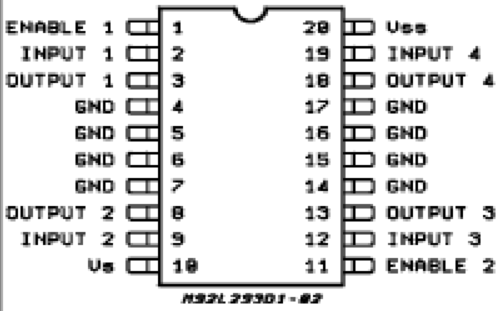
BLOCK DIAGRAM



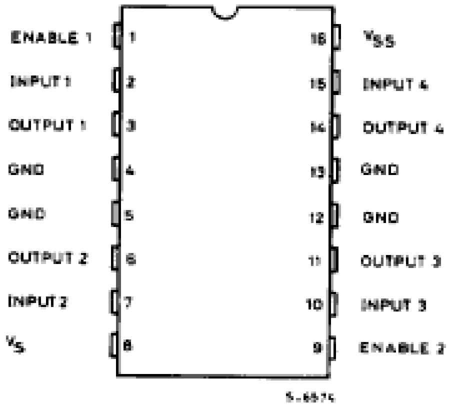
ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Supply Voltage	36	V
V_{SS}	Logic Supply Voltage	36	V
V_i	Input Voltage	7	V
V_{en}	Enable Voltage	7	V
I_o	Peak Output Current (100 µs non repetitive)	1.2	A
P_{tot}	Total Power Dissipation at T_{pins} = 90 °C	4	W
T_{stg}, T_j	Storage and Junction Temperature	– 40 to 150	°C

PIN CONNECTIONS (Top view)



SO(12+4+4)



Powerdip(12+2+2)

THERMAL DATA

Symbol	Decription		DIP	SO	Unit
$R_{th\,j\text{-}pins}$	Thermal Resistance Junction-pins	max.	–	14	°C/W
$R_{th\,j\text{-}amb}$	Thermal Resistance junction-ambient	max.	80	50 (*)	°C/W
$R_{th\,j\text{-}case}$	Thermal Resistance Junction-case	max.	14	–	

(*) With 6sq. cm on board heatsink.