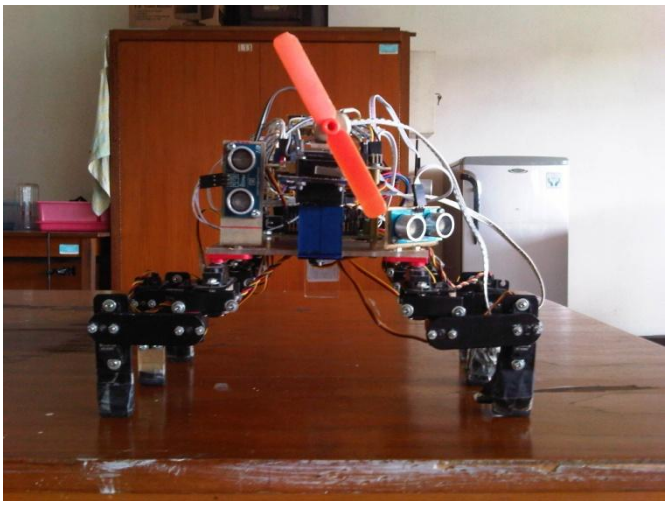
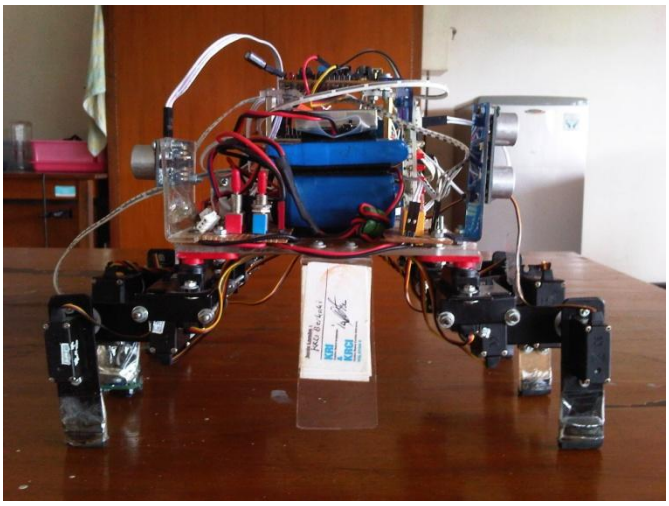
LAMPIRAN A
FOTO ROBOT HEXAPOD

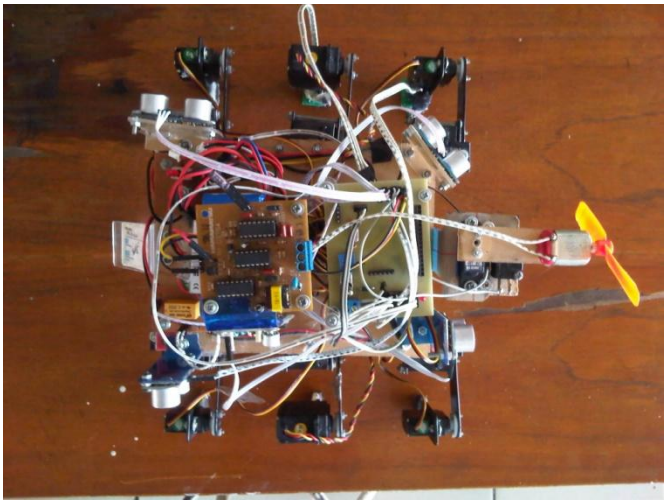
TAMPAK DEPAN



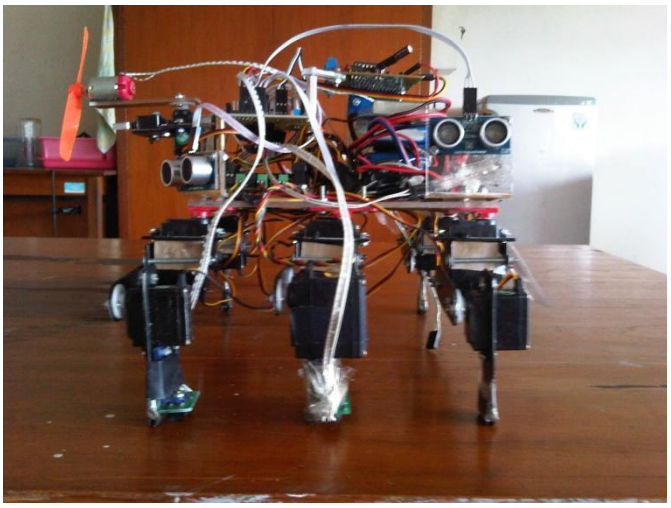
TAMPAK BELAKANG



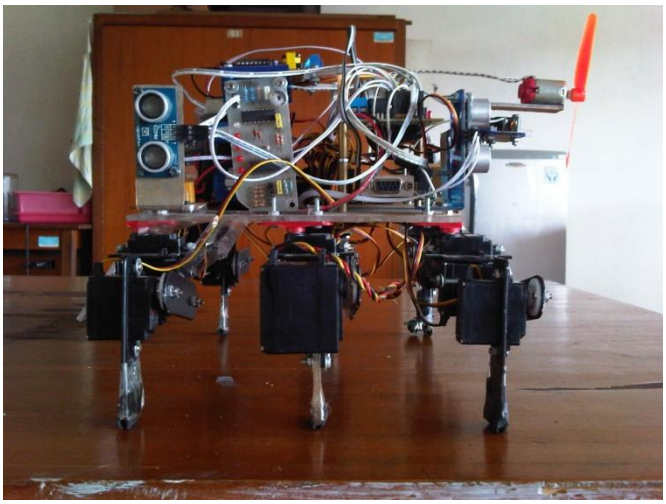
TAMPAK ATAS



TAMPAK KIRI



TAMPAK KANAN



LAMPIRAN B

PROGRAM PADA PENGONTROL MIKRO ATMEGA 128

PROGRAM UTAMA

```
/*****************************************************
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    : 3/30/2011
Author  : F4CG
Company : F4CG
Comments:

Chip type           : ATmega128
Program type        : Application
Clock frequency     : 11.059200 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 1024
*****************************************************/

#include <mega128.h>
#include <delay.h>
#include <stdio.h>

#define ADC_VREF_TYPE 0x40

// 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;
}

// Alphanumeric LCD Module functions
#asm
   .equ __lcd_port=0x1B ;PORTA
#endasm
#include <lcd.h>


// Declare your global variables here
int x0,x1,x2,x3,x4,t;
int c,a,k,z,adc1,adc2,n,m,p,b,q;
// daftar variable :
// c = while utama
// a = pembatas ping
// k = sensor mercury
// z =
// adc1 = nilai adc sensor tengah
// adc2 = nilai adc sensor depan
// n =
// m =
// p = sensor limit switch
```

```
// b = pembatas loop kiri kosong

unsigned char jrk[16];
unsigned char jrk1[16];

char text[32];

// External Interrupt 4 service routine
interrupt [EXT_INT4] void ext_int4_isr(void)
{
// Place your code here
      p=1;
}


// External Interrupt 7 service routine
interrupt [EXT_INT7] void ext_int7_isr(void)
{
// Place your code here
      k=1;
}

#include "arraycoy.c"

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=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;

// Port E 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
PORTE=0x90;
DDRE=0x00;

// Port F 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
PORTF=0x00;
DDRF=0x00;
```

```
// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00;
DDRG=0x1F;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
ASSR=0x00;
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// OC1C 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
// Compare C Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// Timer/Counter 3 initialization
// Clock source: System Clock
// Clock value: Timer 3 Stopped
// Mode: Normal top=FFFFh
// Noise Canceler: Off
// Input Capture on Falling Edge
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Timer 3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
```

```
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
// INT3: Off
// INT4: Off
// INT5: Off
// INT6: Off
// INT7: Off
EICRA=0x00;
EICRB=0xC0;
EIMSK=0x90;
EIFR=0x90;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
ETIMSK=0x00;

// USART1 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART1 Receiver: On
// USART1 Transmitter: On
// USART1 Mode: Asynchronous
// USART1 Baud rate: 115200
UCSR1A=0x00;
UCSR1B=0x18;
UCSR1C=0x06;
UBRR1H=0x00;
UBRR1L=0x05;

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

// ADC initialization
// ADC Clock frequency: 691.200 kHz
// ADC Voltage Reference: AREF pin
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

// LCD module initialization
lcd_init(16);

sprintf(text,"#31 P600 T1000 \r");
usart1Send(text);
```

```
b=0;

aktif();

#asm ("sei")


while (1)
{
      while (c==0)
      {
            z=0;p=0;
            maju();
            if (k==1)
            {      manjat();
                  manjat();
            }
            pingdisp();
            while ((d1==1)||(d3==1)&&(PINC.5==0))
            {
                  steady();
                  extinguisher();
            }
            while (x0<=5)
            {
                  muterkanan();
                  muterkanan();
                  pingdisp();
                  if (x1<5)
                  {      muterkanan();   }
            }
            if (x2>4)
            {
                  if (x2>9)
                  {
                        maju();
                        muterkiri();
                        muterkiri();
                        c=1;
                  }
                  rapatkiri();
            }
            if (x3<9)      //coba r4
            {
                  z=1;
            }
            if (z==1)
            {
                  if (x0>17)
                  {
                        if (x3>=9)
                        {
                              muterkanan();
                              muterkanan();
                              do
                              {      maju(); }
                              while (x2>=9);
                              c=3;
                              }
                  }
            }
      }

      while (c==1)
```

```
    {
        maju();
        if (k==1)
        {    manjat();
             manjat();
        }
        pingdisp();
        if (x2<=9)
        {
             c=0;
             //z=0;
        }
    }
};
}
```

SUBPROGRAM

```
#include <string.h>     /*important coyyy !!!*/

#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// Get a character from the USART1 Receiver
#pragma used+
char getchar1(void)
{
char status,data;
while (1)
    {
    while (((status=UCSR1A) & RX_COMPLETE)==0);
    data=UDR1;
    if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
      return data;
    };
}
#pragma used-

// Write a character to the USART1 Transmitter
#pragma used+
void putchar1(char c)
{
while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
UDR1=c;
}
#pragma used-

int j,d1,d2,d3;
// j = buat servo
// d1,d2,d3 = sensor putih (ADC)

void usart1Send(char * text)
```

```
{
	int i = 0;
	for(i = 0; i < strlen(text);i++)
	{
		putchar1(text[i]);
	}
}

void kirim_tes(flash int * posisi)	//kirim lama
{
	int i = 0;
	char text[32];
	for (i=0;i<18;i++)
	{
		sprintf(text,"#%d P%d T1000 \r",i,posisi[i]);
		usart1Send(text);
	}
	delay_ms(1000);
}

void kirim(flash int * posisi)
{
	int i = 0;
	char text[32];
	for (i=0;i<18;i++)
	{
		sprintf(text,"#%d P%d T100 \r",i,posisi[i]);
		usart1Send(text);
	}
	delay_ms(100);
}

void kirim_tgh(flash int * posisi)
{
	int i = 0;
	char text[32];
	for (i=0;i<18;i++)
	{
		sprintf(text,"#%d P%d T75 \r",i,posisi[i]);
		usart1Send(text);
	}
	delay_ms(75);
}

void kirim_dolphin(flash int * posisi)  //bisa 300 seh tp biar ga cpt rusak aja
{
	int i = 0;
	char text[32];
	for (i=0;i<18;i++)
	{
		sprintf(text,"#%d P%d T500 \r",i,posisi[i]);
		usart1Send(text);
	}
	delay_ms(500);
}

void pingdisp()
{
//	lcd_clear();

	a=0;
	t=0;
	DDRC.0=1;
	PORTC.0=1;
```

```
delay_us(5);
PORTC.0=0;
DDRC.0=0;
PORTC.0=1;
delay_us(750);
while((PINC.0==0) && (a<5000))
{
a++;
delay_us(1);
}
while((PINC.0==1) && (t<5000))
{
t++;
delay_us(1);
}
x0=(t/29.034);
delay_ms(1);

t=0;
a=0;
DDRC.1=1;
PORTC.1=1;
delay_us(5);
PORTC.1=0;
DDRC.1=0;
PORTC.1=1;
delay_us(750);
while((PINC.1==0) && (a<5000))
{
a++;
delay_us(1);
}
while((PINC.1==1) && (t<5000))
{
t++;
delay_us(1);
}
x1=(t/29.034);
delay_ms(1);

t=0;
a=0;
DDRC.2=1;
PORTC.2=1;
delay_us(5);
PORTC.2=0;
DDRC.2=0;
PORTC.2=1;
delay_us(750);
while((PINC.2==0) && (a<5000))
{
a++;
delay_us(1);
}
while((PINC.2==1) && (t<5000))
{
t++;
delay_us(1);
}
x2=(t/29.034);
delay_ms(1);

a=0;
t=0;
```

```
	DDRC.3=1;
	PORTC.3=1;
	delay_us(5);
	PORTC.3=0;
	DDRC.3=0;
	PORTC.3=1;
	delay_us(750);
	while((PINC.3==0) && (a<5000))
	{
	a++;
	delay_us(1);
	}
	while((PINC.3==1) && (t<5000))
	{
	t++;
	delay_us(1);
	}
	x3=(t/29.034);
	delay_ms(1);

	adc1=read_adc(2);	//warna karpet abu 300-400 adc, putih 900-1000, item 0-200
	adc2=read_adc(3);

	if (adc1>800)
	{	d1=1;	}
	if (adc1<=800)
	{	d1=0;	}
	if (adc2>800)
	{	d3=1;	}
	if (adc2<=800)
	{	d3=0;	}

	sprintf(jrk,"dpn %3d  mrg %3d",x0,x1);
	sprintf(jrk1,"kr %3d  kn %3d",x2,x3);
	lcd_gotoxy(0,0);lcd_puts(jrk);
	lcd_gotoxy(0,1);lcd_puts(jrk1);

}

flash int
std[18]={1560,1519,2071,1351,1545,1361,1549,1590,1088,1274,1350,1787,1390,1530,1678,1446,1361,103
6};
void steady()
{	kirim(std);	}

flash int
stdx[18]={1560,1519,1738,1032,1365,1361,1549,1590,755,1474,1350,1454,1390,1530,1678,1646,1361,703
};
void steadyx()
{	kirim(stdx);	}

//langkah 30 angkat 45 (BERES !!!!! YIHAAAAAAAAAA)
flash int mj[6][18]=
{
	{1560,1519,1738,732,865,1361,1549,1590,755,1774,1850,1454,1390,1530,1678,1946,1861,703},
	{1360,1519,2404,732,865,1361,1349,1590,1421,1774,1850,2120,1590,1530,1678,1946,1861,1369},
	{1360,1519,2404,1232,1365,1361,1349,1590,1421,1274,1350,2120,1590,1530,1678,1446,1361,1369},
	{1060,1019,2404,1232,1365,1361,1049,1090,1421,1274,1350,2120,1890,2030,1678,1446,1361,1369},
	{1060,1019,1738,1032,1365,1361,1049,1090,755,1474,1350,1454,1890,2030,1678,1646,1361,703},
	{1560,1519,1738,1032,1365,1361,1549,1590,755,1474,1350,1454,1390,1530,1678,1646,1361,703}
};
void maju()
{
	for(j=0;j<6;j++)
```

```
	{
		kirim_tgh(mj[j]);
	}
}

flash int mdr[6][18]=
{
    {1060,1019,1738,1032,1365,1361,1049,1090,755,1474,1350,1454,1890,2030,1678,1646,1361,703},
    {1060,1019,2404,1232,1365,1361,1049,1090,1421,1274,1350,2120,1890,2030,1678,1446,1361,1369},
    {1360,1519,2404,1232,1365,1361,1349,1590,1421,1274,1350,2120,1590,1530,1678,1446,1361,1369},
    {1360,1519,2404,732,865,1361,1349,1590,1421,1774,1850,2120,1590,1530,1678,1946,1861,1369},
    {1560,1519,1738,732,865,1361,1549,1590,755,1774,1850,1454,1390,1530,1678,1946,1861,703},
    {1560,1519,1738,1032,1365,1361,1549,1590,755,1474,1350,1454,1390,1530,1678,1646,1361,703}
};
void mundur()
{
	for(j=0;j<6;j++)
	{
		kirim_tgh(mdr[j]);
	}
}

void majutest()
{
	for(j=0;j<6;j++)
	{
		kirim_tes(mj[j]);
	}
}

flash int rptkr[6][18]=
{
	{1560,1519,1738,732,865,1361,1549,1590,755,1774,1850,1454,1390,1530,1678,1946,1861,703},
	{1360,1519,2404,732,865,1361,1349,1590,1421,1774,1850,2120,1190,1530,1678,1946,1861,1369},
	{1360,1519,2404,1232,1365,1361,1349,1590,1421,1274,1350,2120,1190,1530,1678,1446,1361,1369},
	{1060,1019,2404,1232,1365,1361,1049,1090,1421,1274,1350,2120,1890,2030,1678,1446,1361,1369},
	{1060,1019,1738,1032,1365,1361,1049,1090,755,1074,1350,1454,1890,2030,1678,1246,1361,703},
	{1560,1519,1738,1032,1365,1361,1549,1590,755,1074,1350,1454,1390,1530,1678,1246,1361,703}
};
void rapatkiri()
{
	for (j=0;j<6;j++)
	{
		kirim_tgh(rptkr[j]);
	}
}

//muter kanan angkat 30
flash int mtrkn[10][18]=
{
    {1560,1519,1572,1351,1545,1361,1549,1590,1467,1274,1350,1289,1390,1530,1678,1446,1361,1487},
    {1227,1186,1572,1351,1545,1361,1216,1257,1467,1274,1350,1289,1723,1863,1678,1446,1361,1487},
    {1227,1186,1150,1351,1545,1361,1216,1257,1096,1274,1350,1787,1723,1863,1678,1446,1361,1966},
    {1560,1519,1150,1351,1545,1361,1549,1590,1096,1274,1350,1787,1390,1530,1678,1446,1361,1966},
    {1560,1519,1150,1018,1212,1361,1549,1590,1096,1607,1683,1787,1390,1530,1678,1779,1694,1966},
    {1560,1519,2096,1018,1212,1361,1549,1590,2061,1607,1683,901,1390,1530,1678,1779,1694,1061},
    {1560,1519,2096,1351,1545,1361,1549,1590,2061,1274,1350,901,1390,1530,1678,1446,1361,1061},
    {1227,1186,2096,1351,1545,1361,1216,1257,2061,1274,1350,901,1723,1863,1678,1446,1361,1061},
    {1227,1186,1572,1351,1545,1361,1216,1257,1467,1274,1350,1289,1723,1863,1678,1446,1361,1487},
    {1560,1519,1572,1351,1545,1361,1549,1590,1467,1274,1350,1289,1390,1530,1678,1446,1361,1487}
};
void muterkanan()
{
	for(j=0;j<10;j++)
```

```
	{
		kirim(mtrkn[j]);
	}
}

//muter kiri angkat 30
flash int mtrkr[10][18]=
{
    {1560,1519,1572,1351,1545,1361,1549,1590,1467,1274,1350,1289,1390,1530,1678,1446,1361,1487},
    {1560,1519,1572,1018,1212,1361,1549,1590,1467,1607,1683,1289,1390,1530,1678,1779,1694,1487},
    {1560,1519,1150,1018,1212,1361,1549,1590,1096,1607,1683,1787,1390,1530,1678,1779,1694,1966},
    {1560,1519,1150,1351,1545,1361,1549,1590,1096,1274,1350,1787,1390,1530,1678,1446,1361,1966},
    {1227,1186,1150,1351,1545,1361,1216,1257,1096,1274,1350,1787,1723,1863,1678,1446,1361,1966},
    {1227,1186,2096,1351,1545,1361,1216,1257,2061,1274,1350,901,1723,1863,1678,1446,1361,1061},
    {1560,1519,2096,1351,1545,1361,1549,1590,2061,1274,1350,901,1390,1530,1678,1446,1361,1061},
    {1560,1519,2096,1018,1212,1361,1549,1590,2061,1607,1683,901,1390,1530,1678,1779,1694,1061},
    {1560,1519,1572,1018,1212,1361,1549,1590,1467,1607,1683,1289,1390,1530,1678,1779,1694,1487},
    {1560,1519,1572,1351,1545,1361,1549,1590,1467,1274,1350,1289,1390,1530,1678,1446,1361,1487}
};
void muterkiri()
{
	for(j=0;j<10;j++)
	{
		kirim(mtrkr[j]);
	}
}

////manjat
flash int mjt[7][18]=
{
	{1560,1519,1150,1351,1545,1028,1549,1590,2061,1274,1350,901,1390,1530,2011,1446,1361,1966},
	{727,686,1150,1851,2045,1028,1049,1090,2061,1774,1850,901,890,1030,2011,2279,2194,1966},
	{2060,1519,1150,1351,1545,1694,2049,1590,2061,774,1350,901,1390,1530,1345,946,1361,1966},
	{1560,1519,1150,1851,2045,1694,1049,1090,2061,1774,1850,901,890,1030,1345,1446,1361,1966},
	{1560,1519,1150,851,1045,1694,1049,1590,2061,1774,1350,901,1890,2030,1345,1446,1361,1966},
	{2060,2019,1150,851,1045,1694,2049,1590,2061,774,1350,901,1890,2030,1345,946,861,1966},
	{2060,2019,1150,851,1045,1028,2049,1590,2061,774,1350,901,1890,2030,2011,946,861,1966}
};
void manjat()
{
	for(j=0;j<7;j++)
	{
		kirim_dolphin(mjt[j]);
	}
	k=0;
}

flash int gsrkr[6][18]=
{
	{1560,1519,2071,732,865,1361,1549,1590,1088,1774,1850,1787,1390,1530,1678,1946,1861,1036},
	{1360,1519,2071,732,865,1361,1349,1590,1088,1774,1850,1787,1190,1530,1678,1946,1861,1036},
	{1360,1519,2071,1232,1365,1361,1349,1590,1088,1274,1350,1787,1190,1530,1678,1446,1361,1036},
	{1060,1019,2071,1232,1365,1361,1049,1090,1088,1274,1350,1787,1890,2030,1678,1446,1361,1036},
	{1060,1019,2071,1032,1365,1361,1049,1090,1088,1074,1350,1787,1890,2030,1678,1246,1361,1036},
	{1560,1519,2071,1032,1365,1361,1549,1590,1088,1074,1350,1787,1390,1530,1678,1246,1361,1036}
};
void geserkiri()
{
	for (j=0;j<6;j++)
	{
		kirim_tgh(gsrkr[j]);
	}
}
```

```
flash int gsrkn[6][18]=
{
	{1560,1519,2071,732,865,1361,1549,1590,1088,1774,1850,1787,1390,1530,1678,1946,1861,1036},
	{1760,1519,2071,732,865,1361,1749,1590,1088,1774,1850,1787,1590,1530,1678,1946,1861,1036},
	{1760,1519,2071,1232,1365,1361,1749,1590,1088,1274,1350,1787,1590,1530,1678,1446,1361,1036},
	{1060,1019,2071,1232,1365,1361,1049,1090,1088,1274,1350,1787,1890,2030,1678,1446,1361,1036},
	{1060,1019,2071,1432,1365,1361,1049,1090,1088,1474,1350,1787,1890,2030,1678,1646,1361,1036},
	{1560,1519,2071,1432,1365,1361,1549,1590,1088,1474,1350,1787,1390,1530,1678,1646,1361,1036}
};
void geserkanan()
{
	for (j=0;j<6;j++)
	{
		kirim_tgh(gsrkn[j]);
	}
}

void extinguisher()
{
	sprintf(text,"#31 P1379 T500 \r");
	usart1Send(text);
	delay_ms(600);
	PORTG=0x11;
	sprintf(text,"#31 P782 T2000 \r");
	usart1Send(text);
	delay_ms(2000);
	sprintf(text,"#31 P1907 T3000 \r");
	usart1Send(text);
	delay_ms(3000);
	sprintf(text,"#31 P1379 T2000 \r");
	usart1Send(text);
	delay_ms(2000);
	PORTG=0x10;
	delay_ms(100);
	sprintf(text,"#31 P600 T1000 \r");
	usart1Send(text);
	delay_ms(1000);
}

void aktif()
{
ulang:
if(PINF.7==1 & PINF.6==0 & PINF.5==0)
{
while(PINF.7==1 & PINF.6==0 & PINF.5==0){delay_us(2);}
}
else
{
goto ulang;
}

if(PINF.7==0 & PINF.6==1 & PINF.5==0)
{
while(PINF.7==0 & PINF.6==1 & PINF.5==0) {delay_us(2);}
	if(PINF.7==1 & PINF.6==1 & PINF.5==0)
	{
	while(PINF.7==1 & PINF.6==1 & PINF.5==0) {delay_us(2);}
		if(PINF.7==0 & PINF.6==0 & PINF.5==1)
		{
		while(PINF.7==0 & PINF.6==0 & PINF.5==1) {delay_us(2);}
		goto lanjut;
		}
		else
		{
```

```
            goto ulang;
            }
        }
        else
        {
        goto ulang;
        }
}
else
{
goto ulang;
}
lanjut:

pingdisp();
while (x2>5)
{
        muterkiri();
        pingdisp();
}
c=0;

}

void scankanan()
{
        if (x2<9)
        {
                if (x3<9)
                {
                        if((300<=adc1<=400)&&(300<=adc2<=400))
                        {       n=1;    }
                }
        }
        maju();
        maju();
        if (x2<9)
        {
                if (x3<9)
                {
                        if((300<=adc1<=400)&&(300<=adc2<=400))
                        {
                                if (n==1)
                                {       n=2;    }
                        }
                }
        }
        if ((n==2)&&(x3>=9))
        {
                muterkanan();
                muterkanan();
                do
                {       maju(); }
                while (x2>=9);
                c=3;
        }

}
```

LAMPIRAN C
DATASHEET

SENSOR JARAK ULTRASONIK PING

PING)))™ Ultrasonic Distance Sensor (#28015)

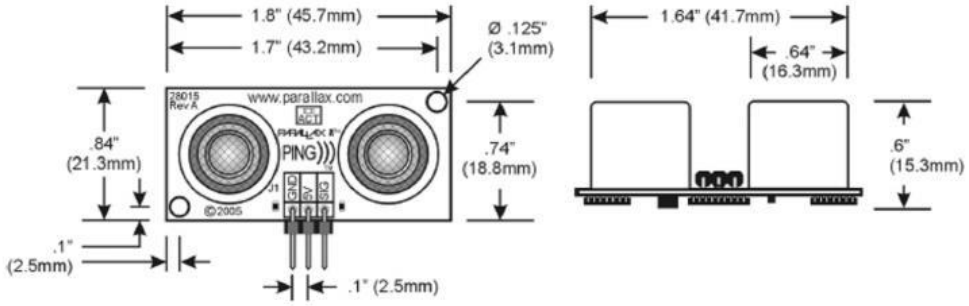
The Parallax PING))) ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to BASIC Stamp® or Javelin Stamp modules, SX or Propeller chips, or other microcontrollers, requiring only one I/O pin.

The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.

Features

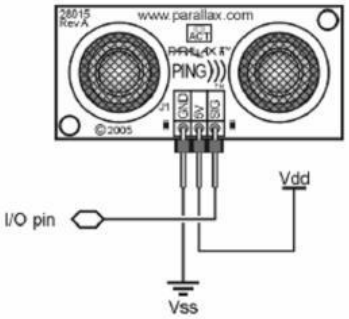
- Supply voltage – +5 VDC
- Supply current – 30 mA typ; 35 mA max
- Range – 2 cm to 3 m (0.8 inches to 3.3 yards)
- Input trigger – positive TTL pulse, 2 µs min, 5 µs typ.
- Echo pulse – positive TTL pulse, 115 µs to 18.5 ms
- Echo hold-off – 750 µs from fall of trigger pulse
- Burst frequency – 40 kHz for 200 µs
- Burst indicator LED shows sensor activity
- Delay before next measurement – 200 µs
- Size – 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Operating temperature: 0 – 70° C.

Dimensions



Pin Definitions

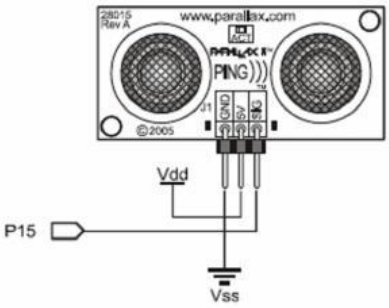
GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

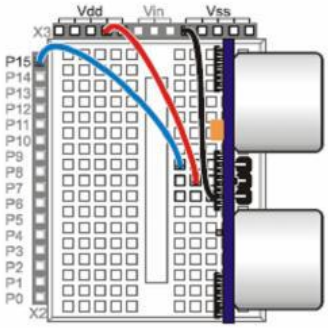


The PING))) sensor has a male 3-pin header used to supply ground, power (5 VDC) and signal. The header allows the sensor to be plugged into a solderless breadboard, or to be located remotely through the use of a standard servo extender cable (Parallax part #805-00002). Standard connections are shown in the diagram to the right.

Quick-Start Circuit

This circuit allows you to quickly connect your PING))) sensor to a BASIC Stamp® 2 via the Board of Education® breadboard area. The PING))) module's GND pin connects to Vss, the 5 V pin connects to Vdd, and the SIG pin connects to I/O pin P15. This circuit will work with the example BASIC Stamp program listed below.

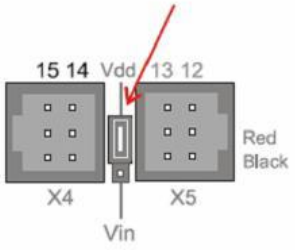




Extension Cable and Port Cautions

If you want to connect your PING))) sensor to a Board of Education platform using an extension cable, follow these steps:

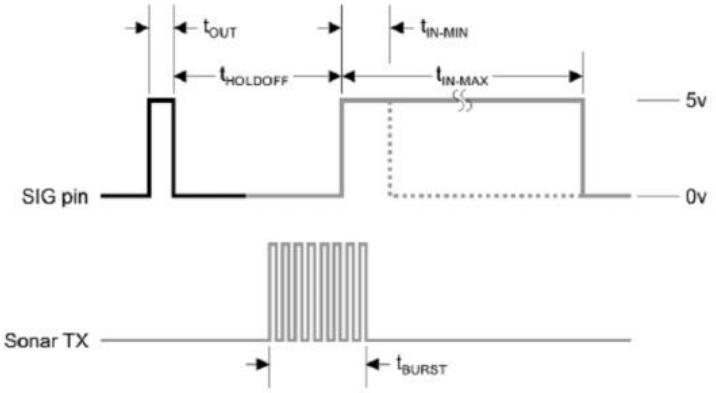
1. When plugging the cable onto the PING))) sensor, connect Black to GND, Red to 5 V, and White to SIG.
2. Check to see if your Board of Education servo ports have a jumper, as shown at right.
3. If your Board of Education servo ports have a jumper, set it to Vdd as shown. Then plug the cable into the port, matching the wire color to the labels next to the port.
4. If your Board of Education servo ports do not have a jumper, **do not use them with the PING))) sensor.** These ports only provide Vin, not Vdd, and this may damage your PING))) sensor. Go to the next step.
5. Connect the cable directly to the breadboard with a 3-pin header as shown above. Then, use jumper wires to connect Black to Vss, Red to Vdd, and White to I/O pin P15.



Board of Education Servo Port Jumper, Set to Vdd

Theory of Operation

The PING))) sensor detects objects by emitting a short ultrasonic burst and then "listening" for the echo. Under control of a host microcontroller (trigger pulse), the sensor emits a short 40 kHz (ultrasonic) burst. This burst travels through the air at about 1130 feet per second, hits an object and then bounces back to the sensor. The PING))) sensor provides an output pulse to the host that will terminate when the echo is detected, hence the width of this pulse corresponds to the distance to the target.

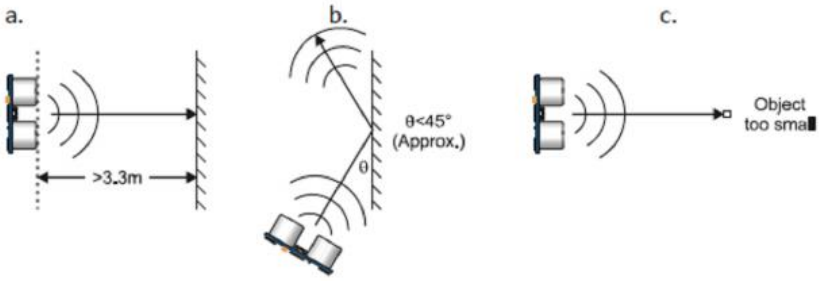


	Host Device	t_{OUT}	2 µs (min), 5 µs typical
	PING))) Sensor	$t_{HOLDOFF}$	750 µs
		t_{BURST}	200 µs @ 40 kHz
		t_{IN-MAX}	115 µs
		t_{IN-MIN}	18.5 µs

Practical Considerations for Use

Object Positioning

The PING))) sensor cannot accurately measure the distance to an object that: a) is more than 3 meters away, b) that has its reflective surface at a shallow angle so that sound will not be reflected back towards the sensor, or c) is too small to reflect enough sound back to the sensor. In addition, if your PING))) sensor is mounted low on your device, you may detect sound reflecting off of the floor.



Target Object Material

In addition, objects that absorb sound or have a soft or irregular surface, such as a stuffed animal, may not reflect enough sound to be detected accurately. The PING))) sensor will detect the surface of water, however it is not rated for outdoor use or continual use in a wet environment. Condensation on its transducers may affect performance and lifespan of the device. See the Water Level with PING))) document on the 28015 product page.

Air Temperature

Temperature has an effect on the speed of sound in air that is measurable by the PING))) sensor. If the temperature (ºC) is known, the formula is:

$$C_{air} = 331.5 + (0.6 \times T_C)\ m/s$$

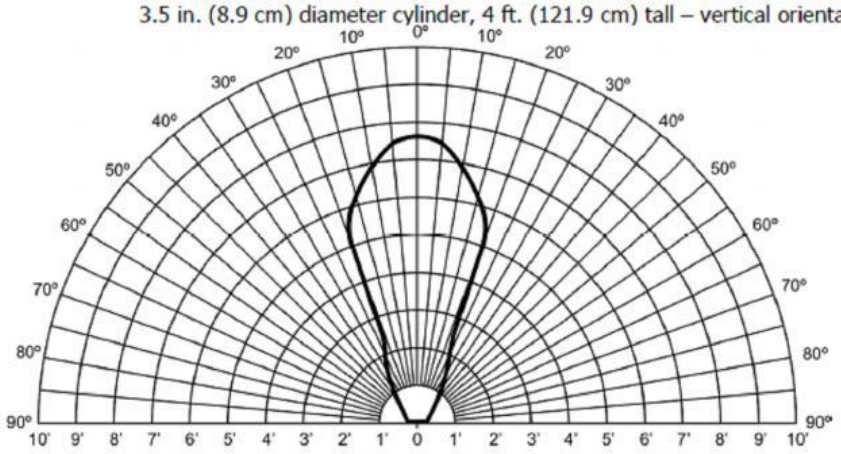
The percent error over the sensor's operating range of 0 to 70 ° C is significant, in the magnitude of 11 to 12 percent. The use of conversion constants to account for air temperature may be incorporated into your program (as is the case in the BS2 program below). Percent error and conversion constant calculations are introduced in Chapter 2 of *Smart Sensors and Applications*, a Stamps in Class text available for download from the 28029 product page.

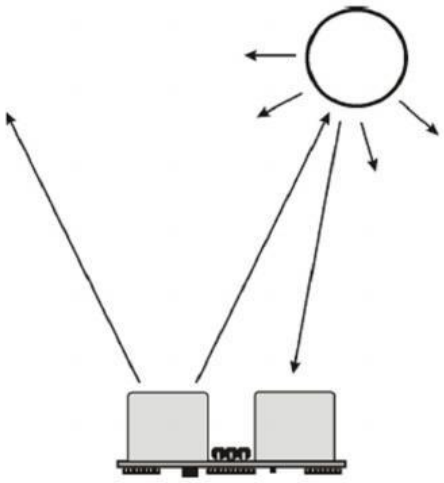
Test Data

The test data on the following pages is based on the PING))) sensor, tested in the Parallax lab, while connected to a BASIC Stamp microcontroller module. The test surface was a linoleum floor, so the sensor was elevated to minimize floor reflections in the data. All tests were conducted at room temperature, indoors, in a protected environment. The target was always centered at the same elevation as the PING))) sensor.

Test 1

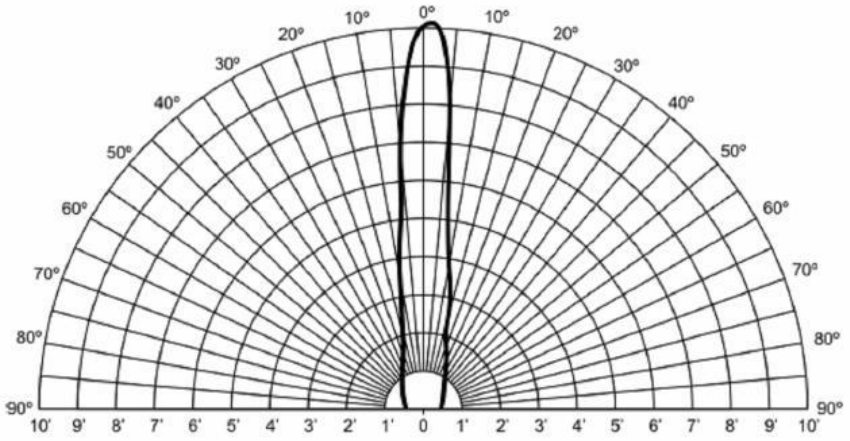
Sensor Elevation: 40 in. (101.6 cm)
Target: 3.5 in. (8.9 cm) diameter cylinder, 4 ft. (121.9 cm) tall – vertical orientation

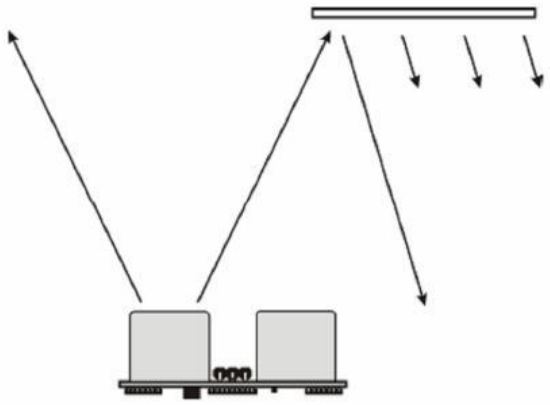




Test 2

Sensor Elevation: 40 in. (101.6 cm)
Target: 12 in. x 12 in. (30.5 cm x 30.5 cm) cardboard, mounted on 1 in. (2.5 cm) pole
Target positioned parallel to backplane of sensor



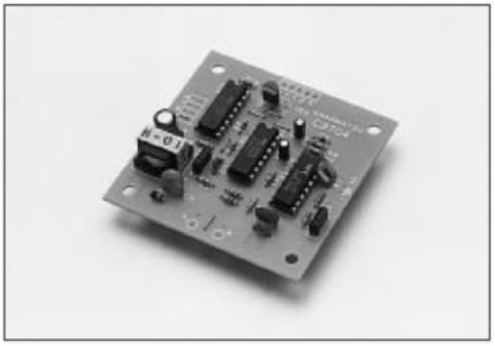


Compact, Lightweight, Low Current Consumption, Low Cost Operates as High Sensitivity UV Sensor with UV TRON Suitable for Flame Detectors and Fire Alarms

Hamamatsu C3704 series UV TRON driving circuits are low current consuming, signal processing circuits for the UV TRON, well known as a high sensitivity ultraviolet detecting tube. The C3704 series can be operated as a UV sensor by connecting the UV TRON and applying DC low voltage, as they have both a high-voltage power supply and a signal processing circuit on the same printed circuit board.

Since background discharges of the UV TRON caused by natural excitation lights (such as a cosmic ray, scattered sunlight, etc.) can be cancelled in the signal processing circuit, the output signals from the C3704 series can be used without errors.

When the high sensitivity sensor "UV TRON R2868" (sold separately) is used, the flame from a cigarette lighter (flame length: 25mm) can be detected even from a distance of more than 5m.



APPLICATIONS

- **Flame detectors for gas and oil lighters**
- **Fire alarms**
- **Combustion monitors for burners**
- **Electric spark detector**
- **UV photoelectric counter**

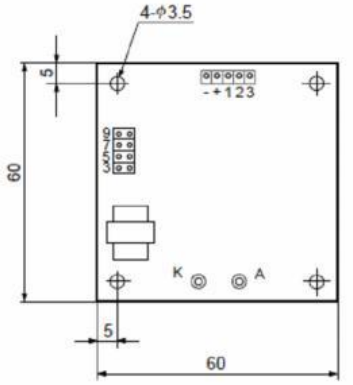
SPECIFICATIONS

Dimensional outline ..... Figure 1
Weight ..... Approx. 20g
Output signal ..... Open collector Output (50 V, 100 mA Max.)
10 ms width pulse output (Note : 1)
UV TRON supply voltage ..... DC 350 V (Note : 2)
Quenching time ..... Approx. 50 ms
Operating temperature ..... -10 to +50°C
(with no condensation)
Suitable UV TRON ..... Low voltage operation UV TRON
(such as R2868)

	C3704	C3704-02	C3704-03
Input Voltage	10 to 30 Vdc	5Vdc ± 5%	6 to 9 Vdc
Current consumption	3 mA Max.	300µA Max.	300µA Max.

Note 1: The output pulse width can be extended up to about 100s by adding a capacitor to the circuit board.
Note 2: Since the output impedance of this power supply is extremely high, an ordinary voltmeter cannot be used. Use a voltmeter that has an input impedance of more than 10 GΩ.

Figure 1: Dimensional Outline (Unit : mm)



<Top View>

<Side View>

TPT A0024EA

Figure 2: Schematic Diagram

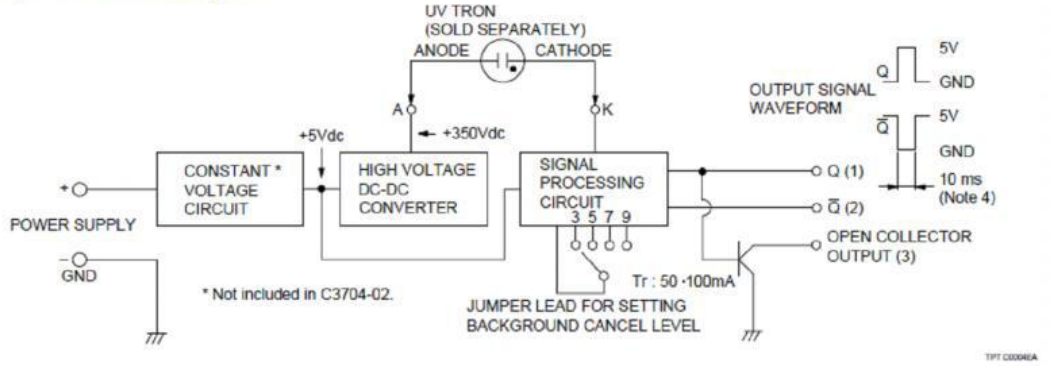


Figure 3: Method of Connection

POWER SUPPLY INPUT AND SIGNAL OUTPUT POINT
− : GND
+ : POWER SUPPLY INPUT
1 : SIGNAL OUTPUT Q
2 : SIGNAL OUTPUT Q̄ (Note 3)
3 : OPEN COLLECTOR OUTPUT

JUMPER LEAD FOR SETTING
BACKGROUND CANCEL LEVEL

This background cancel circuit outputs signal pulses of 10ms width only when 3 to 9 pulses enter the circuit with a time interval of 2 seconds or less from the UV TRON. Cancel level can be set in 4 steps (number of pulses: 3, 5, 7, 9) by this jumper lead. Connect to "3" for general use. When using where much natural excitation light is present, set the cancel level at a higher step (5, 7, 9).

Cx : POINT FOR ADDING A CAPACITOR TO EXTEND OUTPUT PULSE WIDTH (Note 4)

CONNECTING POINT OF UV TRON
A : Connecting Anode of UV TRON
K : Connecting Cathode of UV TRON

Note 3: No load can be driven by an output from points "1" and "2" because these signals are output from the only C-MOS IC directly. When a load such as a buzzer and a relay is connected to this circuit, it should be connected to the point open collector output. The transistor ratings of the open collector is 50V, 100mA. Be careful not to exceed the ratings.

Note 4: The output pulse width is set to 10ms at shipping. If the pulse width needs to be extended, add a capacitor to this point. (When using an electrolytic condenser, make sure the polarity is correct.)
e.g. CX = 1 µF: Pulse Width ≃ 1s, CX=10 µF : Pulse Width ≃ 10s

PRECAUTIONS FOR USE

- Since the operation impedance is extremely high, the UV TRON should be connected as close as possible to the circuit board within 5 cm.
- Take care to avoid external noise since a C-MOS IC is used in the circuit. It is recommended that the whole PC board be put in the shield box when it is used.
- To reduce current consumption, oscillating frequency is very low (approx. 20 Hz) in this DC-DC converter. Thus, the output impedance of the high voltage power supply is extremely high. If the surrounding humidity is high, electrical leakage on the PC board surface may lead to a drop in the supply voltage to the UV TRON. This voltage drop may result in lowered detection performance, so a moistureproof material (silicone compound, etc.) should be applied at the connecting point of the UV TRON, etc., if using the unit in a humid environment.

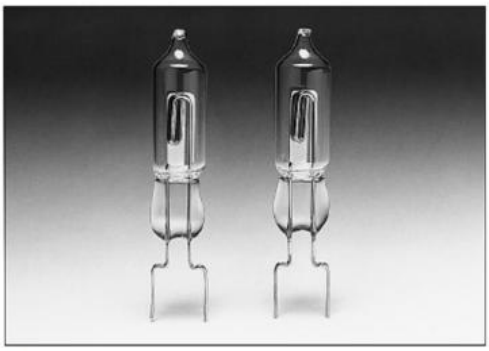
- A model equipped with a flame sensor (R2868) is also available.

Quick Detection of Flame from Distance, Compact UV Sensor with High Sensitivity and Wide Directivity, Suitable for Flame Detectors and Fire Alarms.

Hamamatsu R2868 is a UV TRON ultraviolet detector that makes use of the photoelectric effect of metal and the gas multiplication effect. It has a narrow spectral sensitivity of 185 to 260 nm, being completely insensitive to visible light. Unlike semiconductor detectors, it does not require optical visible-cut filters, thus making it easy to use.

In spite of its small size, the R2868 has wide angular sensitivity (directivity) and can reliably and quickly detect weak ultraviolet radiations emitted from flame due to use of the metal plate cathode (eg. it can detect the flame of a cigarette lighter at a distance of more than 5 m.).

The R2868 is well suited for use in flame detectors and fire alarms, and also in detection of invisible discharge phenomena such as corona discharge of high-voltage transmission lines.



APPLICATIONS

- **Flame detectors for gas/oil lighters and matches**
- **Fire alarms**
- **Combustion monitors for burners**
- **Inspection of ultraviolet leakage**
- **Detection of discharge**
- **Ultraviolet switching**

GENERAL

Parameters	Rating	Units
Spectral Response	185 to 260	nm
Window Material	UV glass	—
Weight	Approx. 1.5	g
Dimensional Outline	See Fig. 3	—

MAXIMUM RATINGS

Parameters	Rating	Units
Supply Voltage	400	Vdc
Peak Current [1]	30	mA
Average Discharge Current [2]	1	mA
Operating Temperature	-20 to +60	°C

CHARACTERISTICS (at 25°C)

Parameters	Rating	Units
Discharge Starting Voltage (with UV radiation)	280	Vdc Max.
Recommended Operating Voltage	325±25	Vdc
Recommended Average Discharge Current	100	µA
Background [3]	10	cpm Max
Sensitivity [4]	5000	cpm Typ.

NOTES:

1) This is the maximum momentary current that can be handled if its full width at half maximum is less than 10 µs.
2) If the tube is operated near this or higher, the service life is noticeably reduced. Use the tube within the recommended current values.
3) Measured under room illuminations (approximately 500 lux) and recommended operating conditions. Note that these values may increase if the following environmental factors are present.
   1. Mercury lamps, sterilization lamps, or halogen lamps are located nearby.
   2. Direct or reflected sunlight is incident on the tube.
   3. Electrical sparks such as welding sparks are present.
   4. Radiation sources are present.
   5. High electric field (including static field) generates across the tube.
4) These are representative values for a wavelength of 200 nm and a light input of 10 pW/cm². In actual use, the sensitivity will vary with the wavelength of the ultraviolet radiation and the drive circuitry employed.

Figure 1: UV TRON's Spectral Response and Various Light Sources

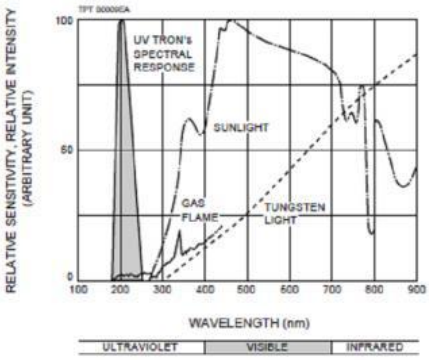


Figure 2: Angular Sensitivity (Directivity)

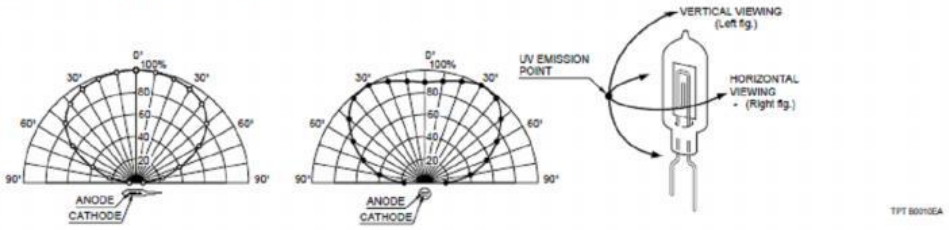


Figure 3: Dimensional Outline (Unit: mm)

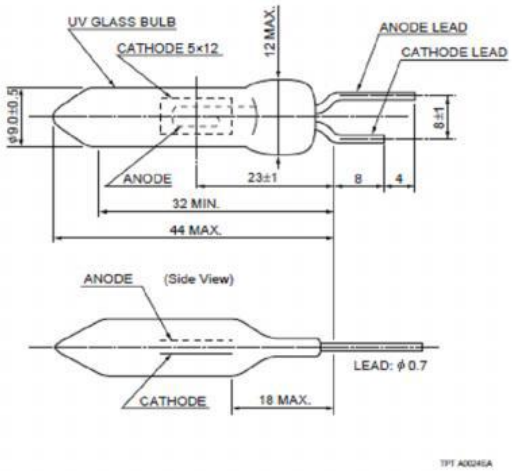
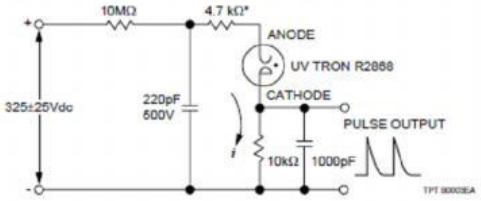
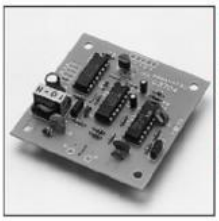


Figure 4: Recommended Operating Circuit



* Be sure to connect the 4.7 kΩ resistor within 2.5 cm from the anode lead end of UV TRON.

- UV TRON Driving Circuit C3704 series (Option)



Hamamatsu also provide the driving circuit C3704 series for R2868 operation.
C3704 series include a high voltage power supply and a signal processing circuit in printed circuit board, which allows to operate R2868 easily as a flame sensor with the low input voltage (DC 6 to 30 V) only.
For the details, please refer to the datasheet of C3704 series.

PRECAUTIONS FOR USE

- **Ultraviolet Radiation**

The UV TRON itself emits ultraviolet radiation in operation. When using two or more UV TRONs at the same time in close position, care should be taken so that they do not optically interfere with each other.

- **Vibration and Shock**

The UV TRON is designed in accordance with the standards of MIL-STD-202F (Method 204D/0.06 inch or 10g, 10- 500Hz, 15 minutes, 1 cycle) and MIL-STD-202F (Method 213B/100g, 11ms, Half-sine, 3 times). However, should a strong shock be sustained by the UV TRON (e.g. if dropped), the glass bulb may crack or the internal electrode may be deformed, resulting in deterioration of electrical characteristics. So extreme care should be taken in handling the tube.

- **Polarity**

Connect the UV TRON with correct polarity. Should it be connected with reverse polarity, operating errors may occur.

WARRANTY.

The UV TRON is covered by a warranty for a period of one year after delivery. The warranty is limited to replacement of any defective tube due to defects traceable to the manufacturer.

VISHAY.

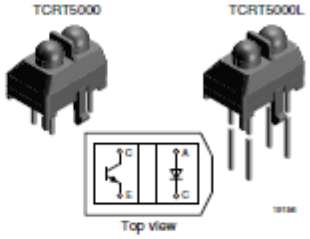
TCRT5000(L)

Vishay Semiconductors

Reflective Optical Sensor with Transistor Output

Description

The TCRT5000 and TCRT500L are reflective sensors which include an infrared emitter and phototransistor in a leaded package which blocks visible light. The package includes two mounting clips. TCRT5000L is the long lead version.



Features

- Package type: Leaded
- Detector type: Phototransistor
- Dimensions:
  L 10.2 mm x W 5.8 mm x H 7.0 mm
- Peak operating distance: 2.5 mm
- Operating range: 0.2 mm to 15 mm
- Typical output current under test: I_C = 1 mA
- Daylight blocking filter
- Emitter wavelength 950 nm
- Lead (Pb)-free soldering released
- Lead (Pb)-free component in accordance to RoHS 2002/95/EC and WEEE 2002/96/EC

Applications

- Position sensor for shaft encoder
- Detection of reflective material such as paper, IBM cards, magnetic tapes etc.
- Limit switch for mechanical motions in VCR
- General purpose - wherever the space is limited

Order Instructions

Part Number	Remarks	Minimum Order Quantity
TCRT5000	3.5 mm lead length	4500 pcs, 50 pcs/tube
TCRT5000L	15 mm lead length	2400 pcs, 48 pcs/tube

Absolute Maximum Ratings

T_{amb} = 25 °C, unless otherwise specified

Input (Emitter)

Parameter	Test condition	Symbol	Value	Unit
Reverse voltage		V_R	5	V
Forward current		I_F	60	mA
Forward surge current	$t_p \leq 10$ µs	I_{FSM}	3	A
Power dissipation	$T_{amb} \leq 25$ °C	P_V	100	mW
Junction temperature		T_j	100	°C

Output (Detector)

Parameter	Test condition	Symbol	Value	Unit
Collector emitter voltage		V_{CEO}	70	V
Emitter collector voltage		V_{ECO}	5	V
Collector current		I_C	100	mA
Power dissipation	$T_{amb} \leq 55$ °C	Pv	100	mW
Junction temperature		T_j	100	°C

Sensor

Parameter	Test condition	Symbol	Value	Unit
Total power dissipation	$T_{amb} \leq 25$ °C	P_{tot}	200	mW
Operation temperature range		T_{amb}	- 25 to + 85	°C
Storage temperature range		T_{stg}	- 25 to + 100	°C
Soldering temperature	2 mm from case, t ≤ 10 s	T_{sd}	260	°C

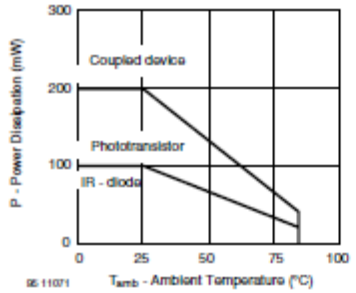


Figure 1. Power Dissipation Limit vs. Ambient Temperature

Electrical Characteristics

T_{amb} = 25 °C, unless otherwise specified

Input (Emitter)

Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Forward voltage	I_F = 60 mA	V_F		1.25	1.5	V
Junction capacitance	V_R = 0 V, f = 1 MHz	C_j		17		pF
Radiant intensity	I_F = 60 mA, t_p = 20 ms	I_E			21	mW/sr
Peak wavelength	I_F = 100 mA	λ_p	940			nm
Virtual source diameter	Method: 63 % encircled energy	Ø		2.1		mm

Output (Detector)

Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Collector emitter voltage	I_C = 1 mA	V_{CEO}	70			V
Emitter collector voltage	I_E = 100 µA	V_{ECO}	7			V
Collector dark current	V_{CE} = 20 V, I_F = 0, E = 0	I_{CEO}		10	200	nA

Sensor

Parameter	Test condition	Symbol	Min	Typ.	Max	Unit
Collector current	V_{CE} = 5 V, I_F = 10 mA, D = 12 mm	I_C [1,2]	0.5	1	2.1	mA
Collector emitter saturation voltage	I_F = 10 mA, I_C = 0.1 mA, D = 12 mm	V_{CEsat} [1,2]			0.4	V

[1] See figure 3

[2] Test surface: Mirror (Mfr. Spindler a. Hoyer, Part No 340005)

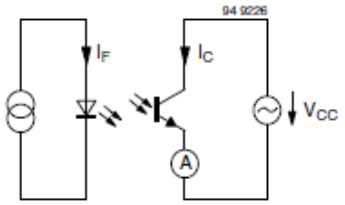


Figure 2. Test Circuit

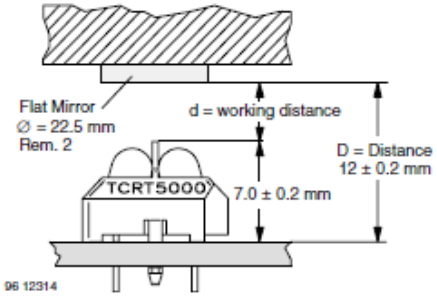


Figure 3. Test Circuit

Typical Characteristics

T_{amb} = 25 °C, unless otherwise specified

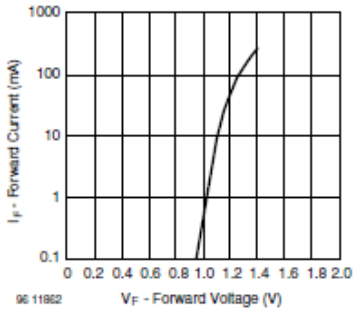


Figure 4. Forward Current vs. Forward Voltage

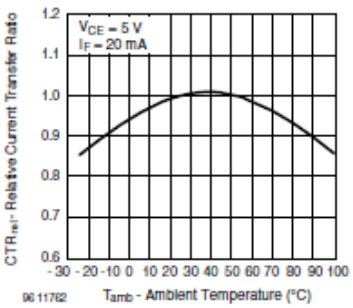


Figure 5. Relative Current Transfer Ratio vs. Ambient Temperature

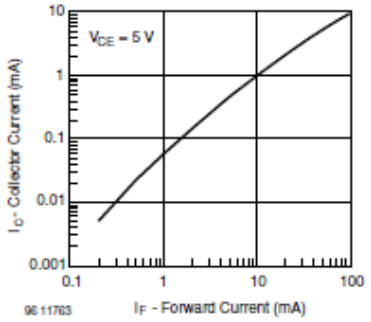


Figure 6. Collector Current vs. Forward Current

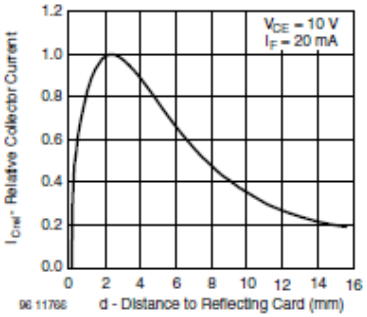


Figure 9. Relative Collector Current vs. Distance

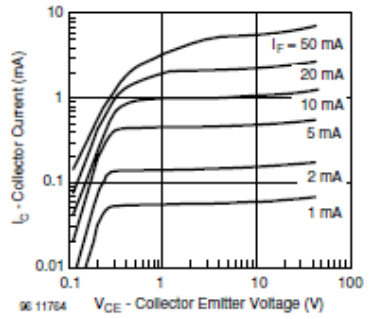


Figure 7. Collector Emitter Saturation Voltage vs. Collector Current

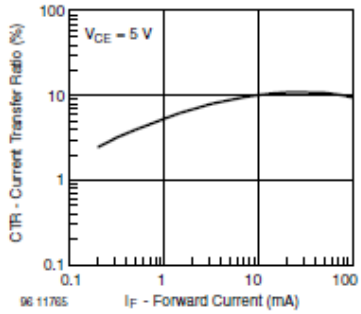
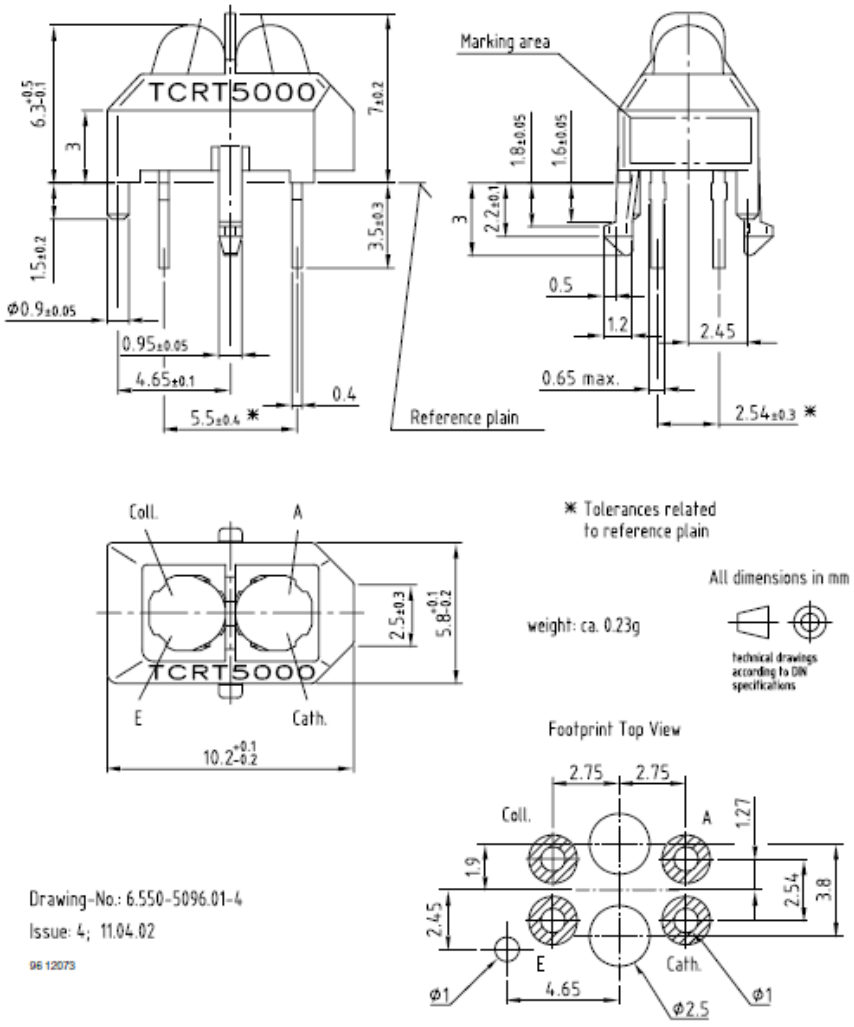
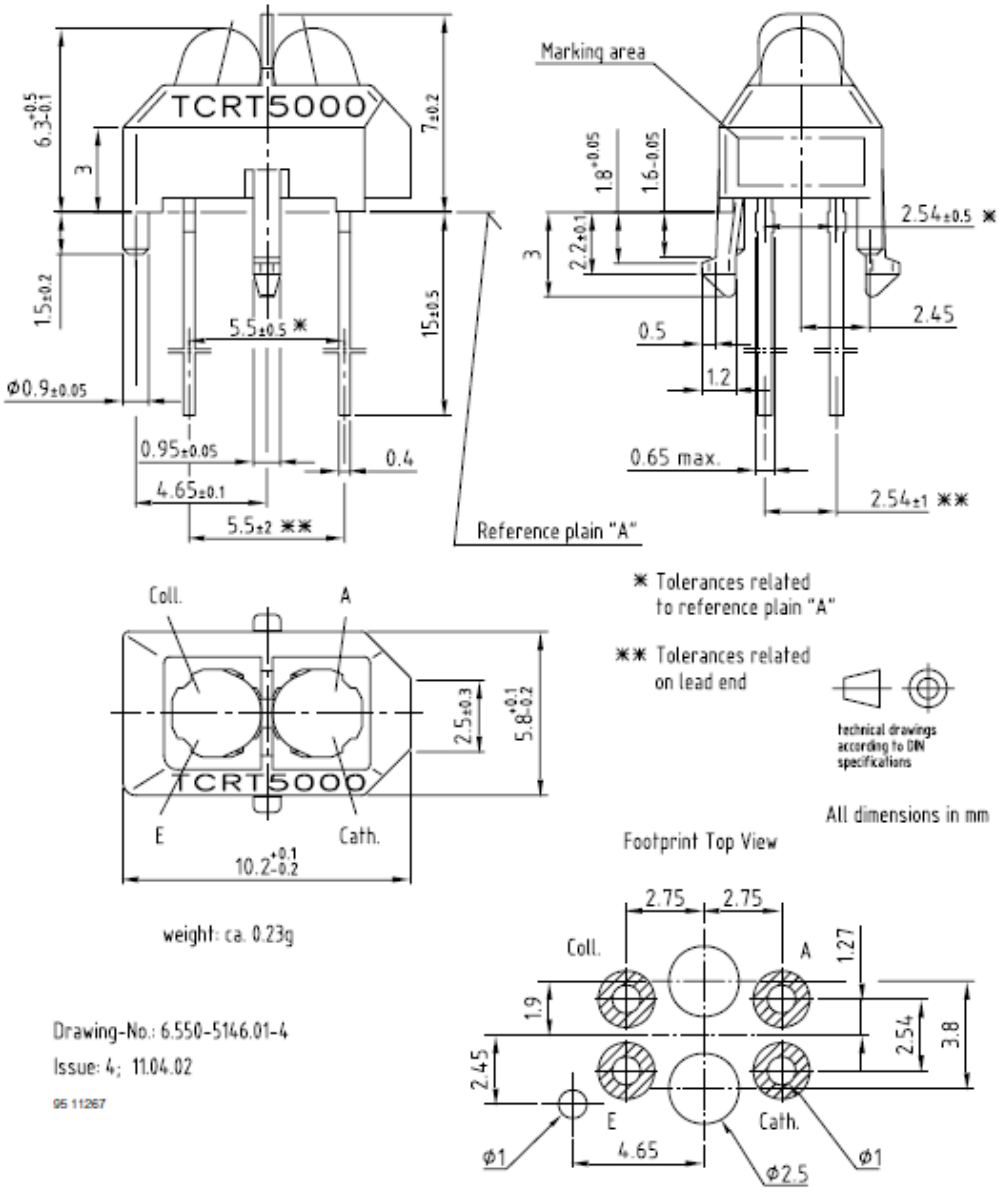


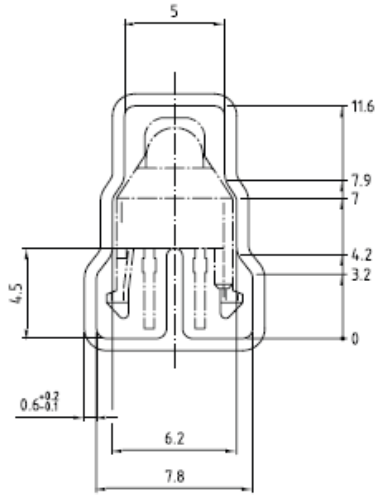
Figure 8. Current Transfer Ratio vs. Forward Current

Package Dimensions in mm




TCRT5000
Marking area
Reference plain "A"
Coll.
A
E
Cath.
TCRT5000
* Tolerances related to reference plain "A"
** Tolerances related on lead end
technical drawings according to DIN specifications
All dimensions in mm
Footprint Top View
weight: ca. 0.23g
Drawing-No.: 6.550-5146.01-4
Issue: 4; 11.04.02
95 11267

TCRT5000, Tube Dimensions

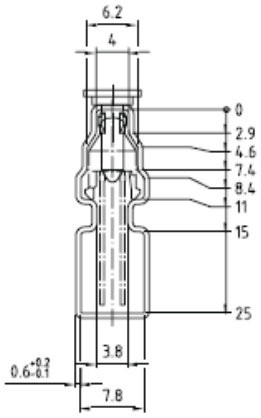


With rubber stopper
Tolerance: ±0.5mm
Length: 575±1mm

All dimensions in mm

Drawing-No.: 9.700-5139.01-4
Issue: 1; 10.05.00

TCRT5000L, Tube Dimensions



With stopper pins
Tolerance: ±0.5mm
Length: 575±1mm

All dimensions in mm

Drawing-No.: 9.700-5178.01-4
Issue: 1; 25.02.00