LAMPIRAN A
FOTO ROBOT BERODA
TAMPAK DEPAN

TAMPAK BELAKANG
TAMPAK SAMPING KIRI

TAMPAK SAMPING KANAN
TAMPAK ATAS
LAMPIRAN B
PROGRAM PADA PENGONTROL MIKRO
ATMEGA 128
#include <meg128.h>
#include <delay.h>
#include <stdio.h>

#include <lcd.h>

#define ADC_VREF_TYPE 0x00

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

// Declare your global variables here

unsigned int a,b,c,d,e,f,i,j,k,l,m,n,p,z,x,x,a,x,b,x,c,x,d,x,e,x,f,x,g,r,h,counter,adc,ade_1,s1,s2,sensor_7,c1,bm,GakBisaKeluar,counter_GakBisaKeluar,GakBisaKeluarDeui;
unsigned char text[32];

//depan kiri depan kanan
//xa,xb,xc

void aktif()
{
    ulang:
    if(PINE.0==0){
        goto lanjut;
    }
    if(PINA.0==1 & PINA.1==0 & PINA.2==0)
    {
        while(PINA.0==1 & PINA.1==0 & PINA.2==0){delay_us(2);}
    }
    else
    {
        goto ulang;
    }
    if(PINA.0==0 & PINA.1==1 & PINA.2==0)
    {
        while(PINA.0==0 & PINA.1==1 & PINA.2==0) {delay_us(2);}
        if(PINA.0==1 & PINA.1==1 & PINA.2==0)
        {
            while(PINA.0==1 & PINA.1==1 & PINA.2==0) {delay_us(2);}
            if(PINA.0==0 & PINA.1==0 & PINA.2==1)
            {
                while(PINA.0==0 & PINA.1==0 & PINA.2==1) {delay_us(2);}
                goto lanjut;
            }}
//goto ulang;
//}
}
else
{
    goto ulang;
}
}
else
{
    goto ulang;
}
lanjut:
}

void servo_on(){
    for(k=60;k>0;k--)
    {
        for(j=0;j<3;j++){//lama pengulangan silkus on yg sama
            PORTE.4=1;
            delay_us(1400);
            for(i=0;i<k;i++){delay_us(10);};
            PORTE.4=0;
            delay_us(18300);
        }
    }
    for(k=0;k<60;k++)
    {
        for(j=0;j<3;j++){//lama pengulangan silkus on yg sama
            PORTE.4=1;
            delay_us(1400);
            for(i=0;i<k;i++){delay_us(10);};
            PORTE.4=0;
            delay_us(18300);
        }
    }
}

void set_servo(){
    PORTE.4=1;
    delay_us(1780);
PORTA.4 = 0;
delay_us(18020);
}

void maju()
{
PORTA.3 = 0;           // kiri
PORTA.4 = 0;
PORTA.5 = 1;           // kanan
PORTA.6 = 0;
}

void kanan()
{
PORTA.3 = 0;
PORTA.4 = 0;
PORTA.5 = 0;
PORTA.6 = 0;
}

void kiri()
{
PORTA.3 = 1;
PORTA.4 = 0;
PORTA.5 = 1;
PORTA.6 = 0;
}

void kanan_doank()
{
PORTA.3 = 0;
PORTA.4 = 0;
PORTA.5 = 1;
PORTA.6 = 1;
}

void kiri_doank()
{
PORTA.3 = 1;
PORTA.4 = 1;
PORTA.5 = 1;
PORTA.6 = 0;
}

void mundur()
{
PORTA.3 = 1;
PORTA.4 = 0;
PORTA.5 = 0;
PORTA.6 = 0;
}
void brenti()
PORTA.3=1;
PORTA.4=1;
PORTA.5=1;
PORTA.6=1;
}

void kiri_mundur()
PORTA.3=1;
PORTA.4=0;
PORTA.5=1;
PORTA.6=1;
}

void kanan_mundur()
PORTA.3=1;
PORTA.4=1;
PORTA.5=0;
PORTA.6=0;
}

void sensor0()

a=0;
DDRD.0=1;
PORTD.0=1;
delay_us(15);
DDRD.0=0;
PORTD.0=0;
delay_us(750);
while(PIND.0==0)
{
delay_us(1);
}
while(PIND.0==1)
{
a++;
delay_us(1);
}
x=(a/29.034);
void sensor1() {
    b=0;
    DDRD.1=1;
    PORTD.1=1;
    delay_us(15);
    DDRD.1=0;
    PORTD.1=0;
    delay_us(750);
    while(PIND.1==0) {
        delay_us(1);
    }
    while(PIND.1==1) {
        b++;
        delay_us(1);
    }
    xas=(b/29.034);
}

void sensor2() {
    c=0;
    DDRD.2=1;
    PORTD.2=1;
    delay_us(15);
    DDRD.2=0;
    PORTD.2=0;
    delay_us(750);
    while(PIND.2==0) {
        delay_us(1);
    }
    while(PIND.2==1) {
        c++;
        delay_us(1);
    }
    xbs=(c/29.034);
}
void sensor3(){
    d=0;
    DDRD.3=1;
    PORTD.3=1;
    delay_us(15);
    DDRD.3=0;
    PORTD.3=0;
    delay_us(750);
    while(PIND.3==0)
        { delay_us(1); }
    while(PIND.3==1)
        { d++;
        delay_us(1); }
    xc=(d/29.034);
}

void sensor4(){
    e=0;
    DDRD.4=1;
    PORTD.4=1;
    delay_us(15);
    DDRD.4=0;
    PORTD.4=0;
    delay_us(750);
    while(PIND.4==0)
        { delay_us(1); }
    while(PIND.4==1)
        { e++;
        delay_us(1); }
    xd=(e/29.034);
void sensor5() {
    f=0;
    DDRD.5=1;
    PORTD.5=1;
    delay_us(15);
    DDRD.5=0;
    PORTD.5=0;
    delay_us(750);
    while(PIND.5==0) {
        delay_us(1);
    }
    while(PIND.5==1) {
        f++;
        delay_us(1);
    }
    xe=(f/29.034);
}

void sensor6() {
    g=0;
    DDRD.6=1;
    PORTD.6=1;
    delay_us(15);
    DDRD.6=0;
    PORTD.6=0;
    delay_us(750);
    while(PIND.6==0) {
        delay_us(1);
    }
    while(PIND.6==1) {
        g++;
        delay_us(1);
    }
}
xf=(g/29.034);

void sensor7()
{
sensor_7=0;
DDRD.7=1;
PORTD.7=1;
delay_us(15);
DDRD.7=0;
PORTD.7=0;
delay_us(750);
while(PIND.7==0)
{
delay_us(1);
}
while(PIND.7==1)
{
sensor_7++;
delay_us(1);
}
xg=(sensor_7/29.034);
}

void wallfollow_kiri()
{
if(x>6)
{
if(xd>25 && xb>27)
{
kiri();
OCR1A=40;
OCR1B=200;
delay_ms(100);
}
else if(xc<13)
{
kiri_doank();
OCR1A=150;
OCR1B=150;
}
else if(xb<8)
{
kanan_doank();
OCR1A=100;
OCR1B=100;
}

else
{
maju();
OCR1A=80;
OCR1B=80;
}
}
else
{
kanan();
OCR1A=220;
OCR1B=220;
}
}
}

}
PORTB.7=0;
if(x>10) //wall kanan
{
    if(xe>10 & xc>18)
    {
        kanan();
        OCR1A=255;
        OCR1B=40;
        delay_ms(100);
    }
    else if(xb<10)
    {
        kanan_doank();
        OCR1A=0;
        OCR1B=0;
    }
    else if(xc<8)
    {
        kiri_doank();
        OCR1A=0;
        OCR1B=0;
    }
    else if(xa<=4){
        kiri();
        OCR1A=100;
        OCR1B=100;
    }
    else
    {
        maju();
        OCR1A=165;
        OCR1B=165;
    }
}
else
{
    kiri();
    OCR1A=210;
    OCR1B=210;
}
if(k==2 && xc>25 && xe>25)
h=1;
}

if(read_adc(1)<150 && read_adc(0)<70 && k==2) // balik home ITEM
{
    r=1;
}

if(h==1 && r==1 && read_adc(1)>=350 && read_adc(0)>=300 && k==2) // home???
{
    brenti();
    OCR1A=255;
    OCR1B=255;
    // PORTB.7=0;
    lcd_putsf("aing home iyeu'n");
    delay_ms(5000);
}

if(read_adc(0)<=160 && read_adc(0)>=100 && read_adc(1)<=250 && read_adc(1)>=150 && h==1){ // ABU
    break;
}

if(read_adc(1)>=300) // PUTIH
{
    s1=1;
}

if(s1==1 && read_adc(1)<150 && k!=2) // ITEM
{
    counter_GakBisaKeluar = counter_GakBisaKeluar+1;

    s1 = 0;
}

if(k==2){
    counter_GakBisaKeluar=0;
}

if(counter_GakBisaKeluar>=1) // lorong
{
    adc=read_adc(0);
    adc_1=read_adc(1);
    sensor7();
    sensor6();
    if((adc<=150 && adc>=100) && (adc_1<=250 && adc_1>=150)) { // blkg &&
        dpn // abu
if(xg<=10 && xg>=5){
    if(xf<=10 && xf>=5){
        c1=1;
        delay_ms(5);
    }
    else goto start_2;
}
else goto start_2;
}

if(counter_GakBisaKeluar>=1 && c1==1 && GakBisaKeluar>=3)
{
    adc=read_adc(0);
    adc_1=read_adc(1);
    sensor7();
    sensor6();
    if((adc<=150 && adc>=100) && (adc_1<=250 && adc_1>=150)) {
        if(xg<=10 && xg>=5){
            if(xf<=10 && xf>=5){
                GakBisaKeluar=0;
                break;
                c1=0;
            }
            else goto start_2;
        }
        else goto start_2;
    }
    else goto start_2;
}
GakBisaKeluarDeui=0;
while(PINE.6==0){
    GakBisaKeluar=0;
    GakBisaKeluarDeui=GakBisaKeluarDeui+1;
    lcd_gotoxy(0,0);
    lcd_clear();
}
sprintf(text,"%d",GakBisaKeluarDeui);
lcd_puts(text);
delay_ms(500);
while(GakBisaKeluarDeui>=3){
mundur();
OCR1A=255;
OCR1B=255;
delay_ms(1000);
break;
GakBisaKeluarDeui=0;
}
}

if(PINE.2==0){
kiri_mundur();
OCR1A=200;
OCR1B=200;
}
if(PINE.5==0){
kanan_mundur();
OCR1A=200;
OCR1B=200;
}
}
else   //ada api
{
   if(read_adc(1)<350)                                     // ABU PALING BESAR, putih paling kcl
   {
      if(x>10)                                                                //wall kanan
      {
         if(xc>10 && xc>18)
         {
            kanan();
            OCR1A=255;
            OCR1B=40;
            delay_ms(100);
         }
         else if(xb<10)
         {
            kanan_doank();
            OCR1A=0;
            OCR1B=0;
         }
      }
else if(xc<8) {
    kiri_doank();
    OCR1A=0;
    OCR1B=0;
}
else if(xa<=4){
    kiri();
    OCR1A=100;
    OCR1B=100;
}
else {
    maju();
    OCR1A=150;
    OCR1B=150;
}
}
else {
    kiri();
    OCR1A=210;
    OCR1B=210;
}
if(PINE.2==0){
    kiri_mundur();
    OCR1A=200;
    OCR1B=200;
}
if(PINE.5==0){
    kanan_mundur();
    OCR1A=200;
    OCR1B=200;
}
}
else {
    maju();
    OCR1A=255;
    OCR1B=255;
k=1;
delay_ms(100);
void main(void)
{
    // Declare your local variables here

    // Input/Output Ports initialization
    // Port A initialization
    // Func? = In Func6 = Out Func5 = Out Func4 = Out Func3 = Out Func2 = In Func1 = In Func0 = In
    // State7 = T State6 = 0 State5 = 0 State4 = 0 State3 = 0 State2 = T State1 = T State0 = T
    PORTA = 0x00;
}

if (read_adc(1) > 350 && k == 1)  // msh putih
{
    brenti();
    OCR1A = 255;
    OCR1B = 255;
    PORTB.7 = 1;
    servo_on();
    k = 2;
    bm = 1;
}

if (read_adc(1) > 350 && k == 1 && bm == 2)  //
{
    brenti();
    OCR1A = 255;
    OCR1B = 255;
    PORTB.7 = 1;
    servo_on();
    k = 2;
    bm = 1;
}

if (bm == 1)
{
    kiri_mundur();
    OCR1A = 130;
    OCR1B = 130;
    delay_ms(250);
    bm = 2;
}

};
DDRA=0x78;

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=0 State6=0 State5=0 State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0xE0;

// 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=P State5=T State4=T State3=T State2=T State1=T State0=P
PORTE=0x41;
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=0x00;

// 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: 10.800 kHz
// Mode: Ph. correct PWM top=00FFh
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// OC1C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0xA1;
TCCR1B=0x05;
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: Timer2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// Timer/Counter 3 initialization
// Clock source: System Clock
// Clock value: Timer3 Stopped
// Mode: Normal top=FFFFh
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer3 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=0x00;
EIMSK=0x00;

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

// aktif();
p=0;    //cek lorong (ga pake ping>>
n=0;    //ada d lorong
m=0;    //var belum dapat posisi
k=0;
l=0;    //posisi island
z=0;    //api sudah padam cari home
counter=0;

//OCR1A = motor kiri
//b= kanan
while (1)
{
    sensor0();  //x depan
    sensor2();  //xb
    sensor3();  //xc kanan
    sensor4();  //xd
    sensor5();  //xe kanan
    sensor6();  //xf
    sensor7();  //xg kanan

    if(m==0)  //cari kiri
    {
        kiri();
        OCR1A=150;
        OCR1B=150;
    }
if (xb<10 && xd<9 && xf<8)
{k=1;}  //sudah dapat tembok kiri

if (k==1)
{ brenti();
 OCR1A=100;
 OCR1B=100;
delay_ms(1000);
if (x>15 && xg>25)                  //cari posisi
{n=0;  //ini sebagai variable posisi robot di lorong
 k=0;  //ini sebagai variable posisi home non arbitary
 m=1;  //pengecekan posisi dihentikan
}
else
{nz=1;  //ini sebagai variable posisi home dan posisi robot......
 k=0;  //pengecekan posisi dihentikan
 m=1;}}
}  //a1

if (m==1)  ///sudah dapat posisi
{                  //ini program navigasinya.
If(l==0)         //step 2   loop counter 3
if(x>12)        //wall follower kiri
{
if(xd>25 && xb>27)
{
 kiri();
 OCR1A=100;
 OCR1B=180;
delay_ms(100);
}
else if(xc<13)
{
 kiri_doank();
 OCR1A=180;
 OCR1B=180;
}
else if(xb<10) //asli 8
{
    kanan_doank();
    OCR1A=180;
    OCR1B=180;
}
else
{
    maju();
    OCR1A=150;
    OCR1B=150;
}
else
{
    kanan();
    OCR1A=180;
    OCR1B=180;
}

if(read_adc(1)>500 && read_adc(0)<=300)
{
    s1=1;
}

if(s1==1) //depan sudah pintu putih.
{
    if(read_adc(0)>500 && read_adc(1)<=300)
    {
        brenti();
        OCR1A=200;
        OCR1B=200;
        delay_ms(400);
        counter=counter+1;
        p=1; //cek ruang
    }
}

if(p==1)
{
    if(PINE.3==1&&lz==0) //tidak ada api dan posisi di lorong
    {mundur(); //balik lorong}
OCR1A=120;
OCR1B=120;
delay_ms(600);
kanan();
OCR1A=200;
OCR1B=200;
delay_ms(900);
maju();
OCR1A=150;
OCR1B=150;
delay_ms(900);
p=0;     //cek ruang selesai
s1=0;    //cek pintu selesai
}
else if(PINE.3==0&&lz==0)               //ada api dan posisi di lorong
  
{}
if (counter>=3)
{
    if (xf<13 && xg<10) //pindah island
        {if (read_adc(1)<300 && read_adc(0)<300 && read_adc(1)>200 && read_adc(0)>200)
             {l=1;} //harus pindah island
        }
}
}

if (l==1)
{

    if(x>8) //wall follower kanan
    {
        if(xc>25 && xc>27)
        {
            kanan();
            OCR1A=180;
            OCR1B=100;
            delay_ms(100);
        }
        else if(xc<10)
        {
            kiri_doank();
            OCR1A=180;
            OCR1B=180;
        }
        else if(xb<13) //asli 8
        {
            kanan_doank();
            OCR1A=180;
            OCR1B=180;
        }
        else
        {
            maju();
            OCR1A=150;
            OCR1B=150;
        }
    }
}
else
{
kiri();
OCR1A=180;
OCR1B=180;
}

if(read_adc(1)>500 && read_adc(0)<=300)
  [s1=1;
];

if(s1==1) //depan sudah pintu putih.
  [if(read_adc(0)>500 && read_adc(1)<=300)
  
  brenti();
  OCR1A=200;
  OCR1B=200;
  delay_ms(400);
  counter=counter+1;
  p=1; //cek ruang
  ]

if(p==1)
{
  if(PINE.3==1&&lz==0) //tidak ada api dan posisi di lorong
    [mundur(); //balik lorong()
    OCR1A=120;
    OCR1B=120;
    delay_ms(600);
    kanan();
    OCR1A=200;
    OCR1B=200;
    delay_ms(900);
    maju();
    OCR1A=150;
    OCR1B=150;
    delay_ms(900);
    p=0; //cek ruang selesai
    s1=0; //cek pintu selesai
  ]
else if(PINE.3==0&&lz==0) //ada api dan posisi di lorong
{maju(); //masuk ruang
OCR1A=130;
OCR1B=130;
delay_ms(250);
brenti();
OCR1A=130;
OCR1B=130;
z=1;
p=0;
s1=0;
lz=1;
}
else if(lz==1) //keluar ruang
{maju();
OCR1A=130;
OCR1B=130;
delay_ms(500);
p=0;
s1=0;
counter=0;
lz=0;
};
} //b4

If(z==1)
{
if(x>12)
{
if(xd>25 && xb>27)
{
kiri();
OCR1A=100;
OCR1B=180;
delay_ms(100);
}
else if(xc<13)
{
if(n==0)
{
    if(s2==0)
    {
    
    
    
    
    } // asli 8
else if(xb<10)
{
    kiri_doank();
    OCR1A=180;
    OCR1B=180;
}
else
{
    maju();
    OCR1A=150;
    OCR1B=150;
}
}
else
{
    kanan();
    OCR1A=180;
    OCR1B=180;
}
}

if(read_adc(1)>500 && read_adc(0)<=300)
{ s1=1;
}

if(s1==1)
{
    if(read_adc(0)>500 && read_adc(1)<=300)
    { mundur();     //balik lorong()
        OCR1A=120;
        OCR1B=120;
        delay_ms(600);
        kanan();
        OCR1A=200;
        OCR1B=200;
    }
delay_ms(900);
maju();
OCR1A=150;
OCR1B=150;
delay_ms(900);
s1=0; }
}
else
{
if(read_adc(1)>500 && read_adc(0)<=300)
{s1=1;
};

if(s1==1)
{if(read_adc(0)>500 && read_adc(1)<=300)
{ maju(); //keluar ke lorong()
OCR1A=120;
OCR1B=120;
delay_ms(200);
s2=0; }
};
}
};

};
}
LAMPIRAN C
DATASHEET

Sensor Ultrasonik (SRF05)......................................................... C-1
Sensor Api (UVTron)................................................................. C-4
Modul C3704............................................................................. C-6
Sensor Warna (TCRT5000)....................................................... C-8
**SRF05 - Ultra-Sonic Ranger**

**Technical Specification**

**Introduction**
The SRF05 is an evolutionary step from the SRF04, and has been designed to increase flexibility, increase range, and to reduce costs still further. As such, the SRF05 is fully compatible with the SRF04. Range is increased from 3 meters to 4 meters. A new operating mode (tying the mode pin to ground) allows the SRF05 to use a single pin for both trigger and echo, thereby saving valuable pins on your controller. When the mode pin is left unconnected, the SRF05 operates with separate trigger and echo pins, like the SRF04. The SRF05 includes a small delay before the echo pulse to give slower controllers such as the Basic Stamp and Picaxe time to execute their pulse in commands.

**Mode 1 - SRF04 compatible - Separate Trigger and Echo**
This mode uses separate trigger and echo pins, and is the simplest mode to use. All code examples for the SRF04 will work for the SRF05 in this mode. To use this mode, just leave the mode pin unconnected - the SRF05 has an internal pull up resistor on this pin.

**Connections for 2-pin Trigger/Echo Mode (SRF04 compatible)**

[Diagram of connections]

Page 1
Mode 2 - Single pin for both Trigger and Echo

This mode uses a single pin for both Trigger and Echo signals, and is designed to save valuable pins on embedded controllers. To use this mode, connect the mode pin to the 5V Ground pin. The echo signal will appear on the same pin as the trigger signal. The SRF05 will not raise the echo line until 700μS after the end of the trigger signal. You have that long to turn the trigger pin around and make it an input and to have your pulse measuring code ready. The PULSIN command found on many popular controllers does this automatically.
To use mode 2 with the Basic Stamp BS2, you simply use PULSOUT and PULSIN on the same pin, like this:

SRF05 PIN 15
Range VAR Word

SRF05 = 0
PULSOUT SRF05, 5
PULSIN SRF05, 1, Range
Range = Range/20

* use any pin for both trigger and echo
* define the 16 bit range variable
* start with pin low
* issue 10us trigger pulse (5 x 2us)
* measure echo time
* convert to cm (divide by 74 for inches)

Calculating the Distance
The SRF05 Timing diagrams are shown above for each mode. You only need to supply a short 10us pulse to the trigger input to start the ranging. The SRF05 will send out an 8 cycle burst of ultrasound at 40kHz and raise its echo line high (or trigger line in mode 2), it then listens for an echo, and as soon as it detects one it lowers the echo line again. The echo line is therefore a pulse whose width is proportional to the distance to the object. By timing the pulse it is possible to calculate the range in inches/centimeters or anything else. If nothing is detected then the SRF05 will lower its echo line anyway after about 30ms.

The SRF04 provides an echo pulse proportional to distance. If the width of the pulse is measured in usecs, then dividing by 58 will give you the distance in cm, or dividing by 148 will give the distance in inches, usecs/58=cm or usecs/148= inches.

The SRF05 can be triggered as fast as every 50ms, or 20 times each second. You should wait 50ms before the next trigger, even if the SRF05 detects a close object and the echo pulse is shorter. This is to ensure the ultrasonic "beep" has faded away and will not cause a false echo on the next ranging.

The other set of 5 pins
The 5 pins marked "programming pins" are used once only during manufacture to program the Flash memory on the PIC16F630 chip. The PIC16F630's programming pins are also used for other functions on the SRF05, so make sure you don't connect anything to these pins, or you will disrupt the modules operation.
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 gusininspiration effect. It has a narrow spectral sensitivity of 105 to 250 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 (e.g., 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
- Inspector of ultraviolet leakage
- Detection of discharge
- Ultraviolet switching

GENERAL

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Rating</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spectral Response</td>
<td>155 to 250</td>
<td>nm</td>
</tr>
<tr>
<td>Window Material</td>
<td>UV/glass</td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>Approx. 1.5</td>
<td>g</td>
</tr>
<tr>
<td>Dimensional Outline</td>
<td>See Fig. 3</td>
<td></td>
</tr>
</tbody>
</table>

MAXIMUM RATINGS

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Rating</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply Voltage</td>
<td>400</td>
<td>Vdc</td>
</tr>
<tr>
<td>Peak Current</td>
<td>30</td>
<td>V</td>
</tr>
<tr>
<td>Average Discharge Current</td>
<td>1</td>
<td>V</td>
</tr>
<tr>
<td>Operating Temperature</td>
<td>-20 to +40</td>
<td>°C</td>
</tr>
</tbody>
</table>

CHARACTERISTICS (at 25°C)

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Rating</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discharge Starting Voltage (with UV radiation)</td>
<td>280</td>
<td>Vdc Max</td>
</tr>
<tr>
<td>Recommended Operating Voltage</td>
<td>330 ± 25</td>
<td>1500</td>
</tr>
<tr>
<td>Recommended Average Discharge Current</td>
<td>100</td>
<td>µA</td>
</tr>
<tr>
<td>Background</td>
<td>1</td>
<td>ppm Max</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>5000</td>
<td>ppm Tpp</td>
</tr>
</tbody>
</table>

NOTES:
1. This is the maximum momentary current that can be handled if its full width at half maximum is less than 13 µs.
2. If the tube is operated near the higher, the sensitivity is noticeably reduced.
3. Use the tube within the recommended current values.
4. Measured under normal conditions (approximately 500 lux) and recommended operating conditions.
5. Mercury lamps, fluorescent lamps, or halogen lamps are located nearby.
6. Direct or reflected sunlight is incident on the tube.

Electrical sparks such as welding sparks are present.

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

Subject to local technical requirements and regulations, availability of products included in this promotional material may vary. Please consult with our sales office for further details.

Specifications are subject to change without notice. No patent rights are granted to any of the results described herein. © 1987 Hamamatsu Photonics K.K.
Figure 2: Angular Sensitivity (Directivity)

Figure 3: Dimensional Outline (Unit: mm)

Figure 4: Recommended Operating Circuit

PRECAUTIONS FOR USE

- Ultraviolet Radiation
  The UV TRON absorbs ultraviolet radiation in operation. When using two or more UV TRONs at the same time in close proximity, care should be taken to ensure 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 and MIL-STD-202F (Method 211B/006, 11ms, Half-sine, 3 times). However, a strong shock or excessive vibration may result in damage to the anode or cathode leads, and the glass bulb may crack or deform, resulting in a decrease in sensitivity or electrical characteristics. Therefore, care should be taken in handling the tube.

- Polarity
  Connect the UV TRON with correct polarity. An incorrect connection with reverse polarity may result in damage to the tube.

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.

HAMAMATSU PHOTONICS K.K., Electron Tube Center


DA: Hamamatsu Corporation, 360 E. Freedom Road, Irvine, CA 92618, U.S.A., Telephone: (714) 251-2000, Fax: (714) 251-2121

Hamamatsu Photonics GmbH, Am Oberen Holzband 1, 42279, Rüthen, Germany, Telephone: (49) 2302-920, Fax: (49) 2302-9210

Hamamatsu Photonics Nippon, Ltd., 1-10-30, Nakanoshima, Tokyo, 105-0025, Japan. Telephone: (81) 3-3508-8000, Fax: (81) 3-3508-8010

Copyright © 1992, 1996 Hamamatsu Photonics K.K. All rights reserved. Printed in Japan.
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 R 296" 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 (50V, 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 R296)

<table>
<thead>
<tr>
<th>Type</th>
<th>C3704</th>
<th>C3704-02</th>
<th>C3704-03</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Voltage</td>
<td>10 to 30 Vdc</td>
<td>± 5%</td>
<td>6 to 9 Vdc</td>
</tr>
<tr>
<td>Current consumption</td>
<td>3 mA Max</td>
<td>300 μA Max</td>
<td>300 μA Max</td>
</tr>
</tbody>
</table>

Note 1: The output pulse width can be extended up to about 100 μs by adding a capacitor to the circuit board.

Note 2: Since the output impedance of the power supply is extremely high, an ordinary voltmeter cannot be used. Use a voltmetre that has an input impedance of more than 10 GΩ.

Figure 1: Dimensional Outline (Unit: mm)

<Top View>
<Side View>

Subject to local technical requirements and regulations, availability of products included in this promotional material may vary. Please consult with your sales office.

Information furnished by HAMAMATSU is believed to be reliable. However, no responsibility is assumed for possible inaccuracies or omissions.

Specifications are subject to change without notice. No patent rights are granted to any of the circuits, descriptions herein © 1997 Hamamatsu, Photonics K.K.
Figure 2: Schematic Diagram

Figure 3: Method of Connection

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 6 cm.
- Take care to avoid external noise since a CMOS 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 (R2886) is also available.

HAMA MATSU

314-5, Shinokuma, Toyotaka-ku, Kunitachi-shi, Edogawa-ku, Tokyo, Japan, Telephone: (03) 5839-5245, Fax: (03) 5839-5246, Tel. 4226-1800

L.P.I. Hamamatsu Corporation, 229 Pomona Road, Woodcliff Lake N.J. 07675-0005, U.S.A., Telephone: 201-933-1800, Fax: 201-933-1848

GERMANY: Hamamatsu Photonics Deutschland GmbH, Koebergstr. 16, D-22701 Hamburg, Germany, Telephone: 040-8051-0, Fax: 040-8051-11

FRANCE: Hamamatsu Photonics France S.A.R.L., 8 Rue du Clos Tron, 92150 Antony, France, Telephone: 01-47 28 37 61, Fax: 01-47 28 37 60

ITALY: Hamamatsu Photonics Italia S.r.l., Via Dante, 3, 20121 Milano, Italy, Telephone: 02-3752-50, Fax: 02-3752-51

U.K.: Hamamatsu Photonics UK Ltd., 15-17 Euston Road, London NW1 2NT, Telephone: 01-844-0036, Fax: 01-844-0037

ITALY: Hamamatsu Photonics Italia Srl., Via Dante, 3, 20121 Milano, Italy, Telephone: 02-3752-50, Fax: 02-3752-51

HAMS Matsu

TFT HH0721

JUL 1997 CR

Created in Japan

C-7
Reflective Optical Sensor with Transistor Output

Description
The TCRT5000 and TCRT5000L 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_{out} = 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
<table>
<thead>
<tr>
<th>Part Number</th>
<th>Remarks</th>
<th>Minimum Order Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCRT5000</td>
<td>3.5 mm lead length</td>
<td>4500 pcs, 50 pcs/pack</td>
</tr>
<tr>
<td>TCRT5000L</td>
<td>15 mm lead length</td>
<td>2400 pcs, 48 pcs/pack</td>
</tr>
</tbody>
</table>

Absolute Maximum Ratings
T_{ case } = 25 °C, unless otherwise specified

Input (Emitter)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Test condition</th>
<th>Symbol</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reverse voltage</td>
<td></td>
<td>V_{R}</td>
<td>5</td>
<td>V</td>
</tr>
<tr>
<td>Forward current</td>
<td></td>
<td>I_F</td>
<td>60</td>
<td>mA</td>
</tr>
<tr>
<td>Forward surge current</td>
<td>I_{P}</td>
<td>I_{p}</td>
<td>3</td>
<td>A</td>
</tr>
<tr>
<td>Power dissipation</td>
<td>T_{J} = 25 °C</td>
<td>P_{J}</td>
<td>100</td>
<td>mW</td>
</tr>
<tr>
<td>Junction temperature</td>
<td></td>
<td>T_{J}</td>
<td>100</td>
<td>°C</td>
</tr>
</tbody>
</table>
TCRT5000(L)

**Vishay Semiconductors**

**Output (Detector)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Test condition</th>
<th>Symbol</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collector emitter voltage</td>
<td></td>
<td>V&lt;sub&gt;CEO&lt;/sub&gt;</td>
<td>70</td>
<td>V</td>
</tr>
<tr>
<td>Emitter collector voltage</td>
<td></td>
<td>V&lt;sub&gt;ECC&lt;/sub&gt;</td>
<td>5</td>
<td>V</td>
</tr>
<tr>
<td>Collector current</td>
<td></td>
<td>I&lt;sub&gt;C&lt;/sub&gt;</td>
<td>100</td>
<td>mA</td>
</tr>
<tr>
<td>Power dissipation</td>
<td></td>
<td>P&lt;sub&gt;V&lt;/sub&gt;</td>
<td>100</td>
<td>mW</td>
</tr>
<tr>
<td>Junction temperature</td>
<td></td>
<td>T&lt;sub&gt;J&lt;/sub&gt;</td>
<td>100</td>
<td>°C</td>
</tr>
</tbody>
</table>

**Sensor**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Test condition</th>
<th>Symbol</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total power dissipation</td>
<td></td>
<td>P&lt;sub&gt;tot&lt;/sub&gt;</td>
<td>200</td>
<td>mW</td>
</tr>
<tr>
<td>Operation temperature range</td>
<td></td>
<td>T&lt;sub&gt;amb&lt;/sub&gt;</td>
<td>-25 to +65</td>
<td>°C</td>
</tr>
<tr>
<td>Storage temperature range</td>
<td></td>
<td>T&lt;sub&gt;stg&lt;/sub&gt;</td>
<td>-25 to +100</td>
<td>°C</td>
</tr>
<tr>
<td>Soldering temperature</td>
<td></td>
<td>T&lt;sub&gt;sd&lt;/sub&gt;</td>
<td>260</td>
<td>°C</td>
</tr>
</tbody>
</table>

![Power Dissipation Limit vs. Ambient Temperature](image)

**Electrical Characteristics**

T<sub>amb</sub> = 25°C, unless otherwise specified

**Input (Emitter)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Test condition</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ.</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward voltage</td>
<td>I&lt;sub&gt;F&lt;/sub&gt; = 50 mA</td>
<td>V&lt;sub&gt;F&lt;/sub&gt;</td>
<td>1.25</td>
<td>1.5</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>Junction capacitance</td>
<td>V&lt;sub&gt;N&lt;/sub&gt; = 0 V, f = 1 MHz</td>
<td>C&lt;sub&gt;j&lt;/sub&gt;</td>
<td>17</td>
<td>21</td>
<td>pF</td>
<td></td>
</tr>
<tr>
<td>Radiant intensity</td>
<td>I&lt;sub&gt;r&lt;/sub&gt; = 50 mA, t&lt;sub&gt;r&lt;/sub&gt; = 20 ms</td>
<td>I&lt;sub&gt;r&lt;/sub&gt;</td>
<td>21</td>
<td>960</td>
<td>mW/m²</td>
<td></td>
</tr>
<tr>
<td>Peak wavelength</td>
<td>I&lt;sub&gt;p&lt;/sub&gt; = 100 mA</td>
<td>λ&lt;sub&gt;p&lt;/sub&gt;</td>
<td>0.65</td>
<td>2.1</td>
<td>nm</td>
<td></td>
</tr>
<tr>
<td>Virtual source diameter</td>
<td>Method: 63 % encircled energy</td>
<td>Φ</td>
<td>9400</td>
<td>2100</td>
<td>mm</td>
<td></td>
</tr>
</tbody>
</table>

**Output (Detector)**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Test condition</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ.</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collector emitter voltage</td>
<td></td>
<td>V&lt;sub&gt;CEO&lt;/sub&gt;</td>
<td>70</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Emitter collector voltage</td>
<td></td>
<td>V&lt;sub&gt;ECC&lt;/sub&gt;</td>
<td>7</td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collector dark current</td>
<td></td>
<td>V&lt;sub&gt;C&lt;/sub&gt;</td>
<td>10</td>
<td>200</td>
<td>mA</td>
<td></td>
</tr>
</tbody>
</table>

[www.vishay.com](http://www.vishay.com)  
Divisão 5000, 81760, 04-09-03

C-9
Sensor

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Test condition</th>
<th>Symbol</th>
<th>Min</th>
<th>Typ.</th>
<th>Max</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collector current</td>
<td>$V_{CC} = 5,\text{V}, I_p = 13,\text{mA}, D = 12,\text{mm}$</td>
<td>$I_C^{(1), (2)}$</td>
<td>0.5</td>
<td>1</td>
<td>2.1</td>
<td>mA</td>
</tr>
<tr>
<td>Collector emitter saturation voltage</td>
<td>$I_C = 10,\text{mA}, I_E = 3.1,\text{mA}, D = 12,\text{mm}$</td>
<td>$V_{CEsat}^{(1), (2)}$</td>
<td></td>
<td>0.4</td>
<td></td>
<td>V</td>
</tr>
</tbody>
</table>

$^{(1)}$ See figure 3
$^{(2)}$ Test surface: Mirror (Mtr. Spindler & Hoyer, Part No 346005)

![Test Circuit](image)

Figure 2. Test Circuit

Typical Characteristics

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

![Forward Current vs. Forward Voltage](image)

Figure 4. Forward Current vs. Forward Voltage

![Relative Current Transfer Ratio vs. Ambient Temperature](image)

Figure 5. Relative Current Transfer Ratio vs. Ambient Temperature
Figure 6. Collector Current vs. Forward Current

Figure 7. Collector Emitter Saturation Voltage vs. Collector Current

Figure 8. Current Transfer Ratio vs. Forward Current

Figure 9. Relative Collector Current vs. Distance