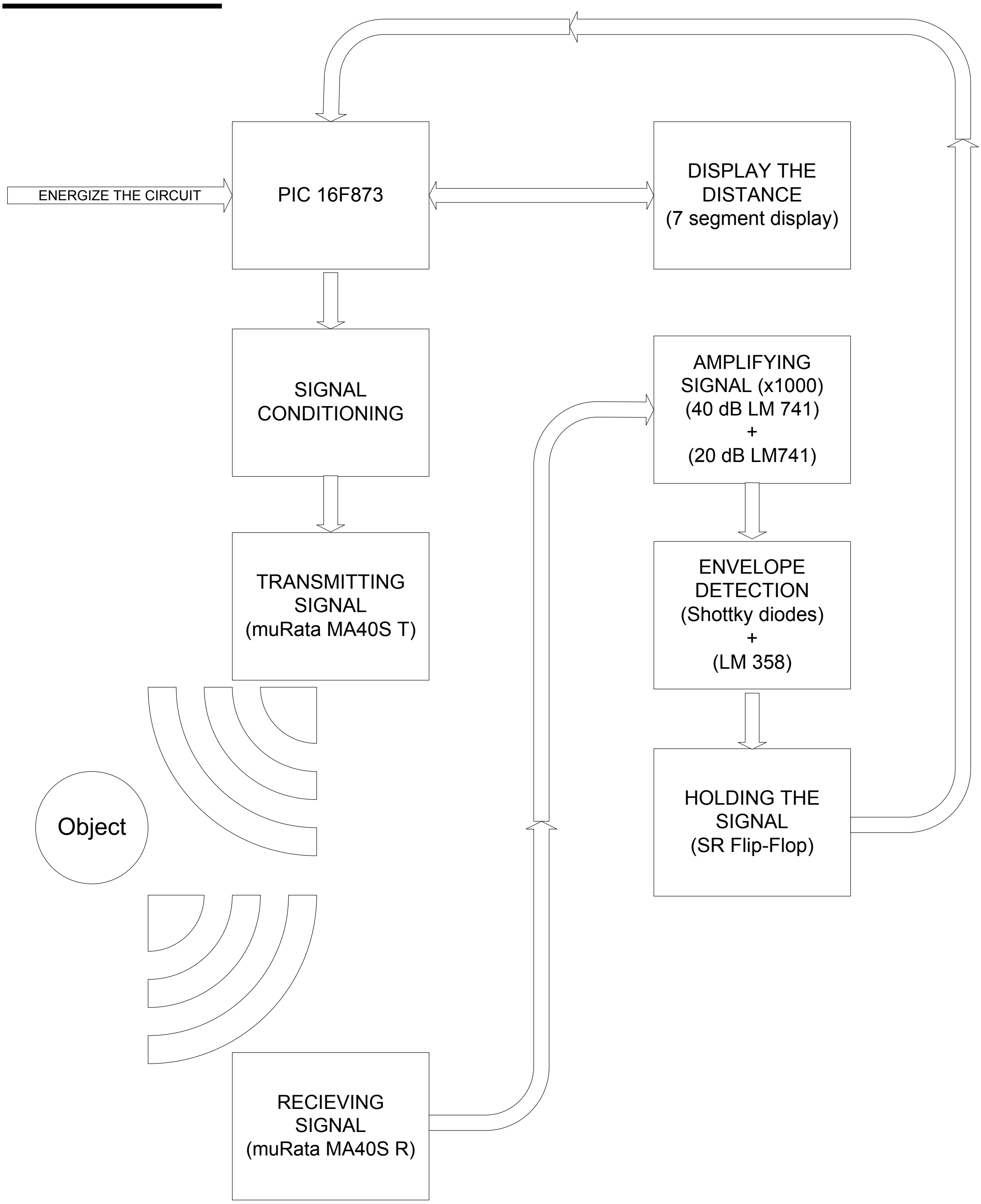
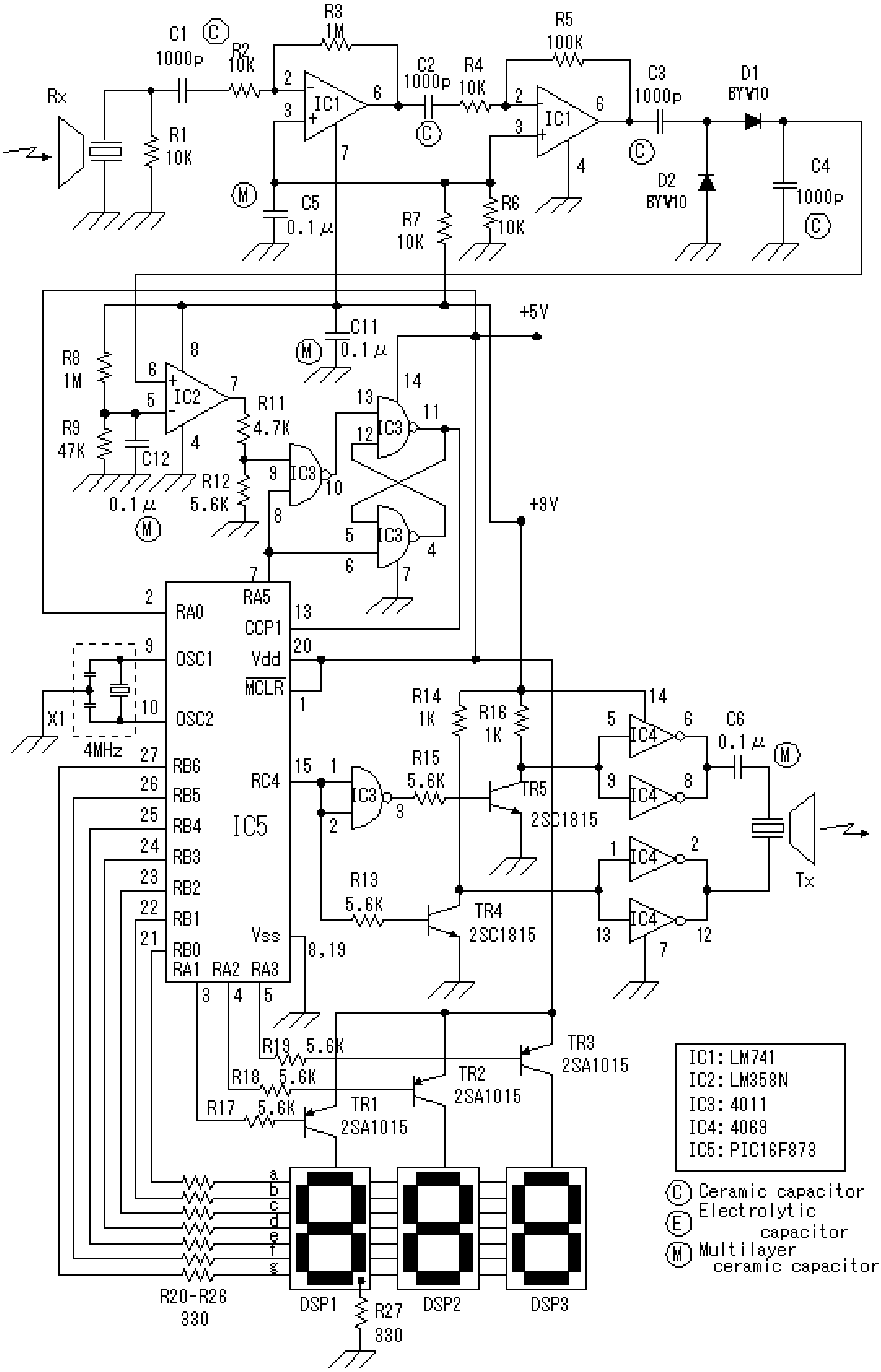
PROCESS



Block diagram of our design

HARDWARE



Rx
C1 1000p
R2 10K
R3 1M
R1 10K
IC1
C5 0.1μ
C2 1000p
R4 10K
R5 100K
R6 10K
R7 10K
C3 1000p
D1 BYV10
D2 BYV10
C4 1000p
+5V
+9V
R8 1M
R9 47K
C12 0.1μ
IC2
R11 4.7K
R12 5.6K
C11 0.1μ
IC3
RA0
RA5
CCP1
OSC1
Vdd
MCLR
OSC2
X1
4MHz
RB6
RB5
RB4
RB3
RB2
RB1
RB0
RC4
Vss
RA1 RA2 RA3
IC5
R14 1K
R16 1K
R15 5.6K
TR5 2SC1815
R13 5.6K
TR4 2SC1815
IC4
C6 0.1μ
Tx
R19 5.6K
R18 5.6K
R17 5.6K
TR3 2SA1015
TR2 2SA1015
TR1 2SA1015
R20-R26 330
DSP1
DSP2
DSP3
R27 330
IC1: LM741
IC2: LM358N
IC3: 4011
IC4: 4069
IC5: PIC16F873
C Ceramic capacitor
E Electrolytic capacitor
M Multilayer ceramic capacitor

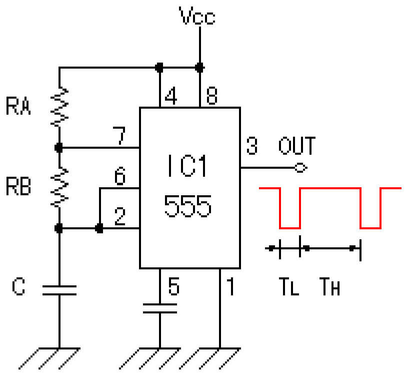
THE TRANSMITTER PART

OSCILLATOR CIRCUITS

By using 555 timers,we tried to set up several oscillator circuits.
The circuit explanatons as follows;

OSCILLATOR CIRCUIT I

We used two 555 timer ICs for the transmitter circuit of the ultrasonic. The first 555 timer used as ultrasonic pulse oscillotor. The IC1 is the oscillation circuit to control the sending-out time of the ultrasonic pulse. The time of the oscillation pulse can be calculated by the following formula. ,



RA = 9.1M-ohm, RB = 150K-ohm, C = 0.01µF

$T_L = 0.69 \times RB \times C$

$= 0.69 \times 150 \times 10^3 \times 0.01 \times 10^{-6}$

$= 1 \times 10^{-3}$

= **1 msec**

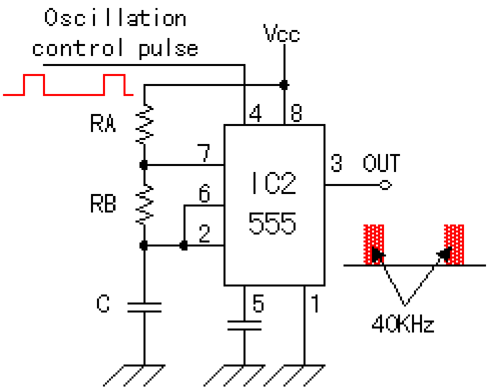
$T_H = 0.69 \times (RA + RB) \times C$

$= 0.69 \times 9250 \times 10^3 \times 0.01 \times 10^{-6}$

$= 64 \times 10^{-3}$

= **64 msec**

IC2 is the circuit to make oscillate the ultrasonic frequency of 40KHz. Oscillation's operation is same as IC1 and makes oscillate at the frequency of about 40 KHz. It makes RB>RA to bring the duty(Ratio of ON/OFF) of the oscillation wave close to 50%. The frequency of the ultrasonic must be adjusted to the resonant frequency of the ultrasonic sensor.



The condition : RA = 1.5K-ohm, RB = 15K-ohm. C = 1000pF

$T_L = 0.69 \times RB \times C$

$= 0.69 \times 15 \times 10^3 \times 1000 \times 10^{-12}$

$= 10.35 \times 10^{-6}$

= **10 µsec**

$T_H = 0.69 \times (RA + RB) \times C$

$= 0.69 \times 16.5 \times 10^3 \times 1000 \times 10^{-12}$

$= 11.39 \times 10^{-6}$

= **11 µsec**

$f = 1 / (T_L + T_H)$

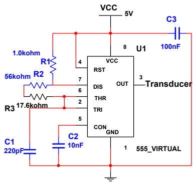
$= 1 / ((10.36 + 11.39) \times 10^{-6})$

$= 46.0 \times 10^3$

= **46.0 KHz**

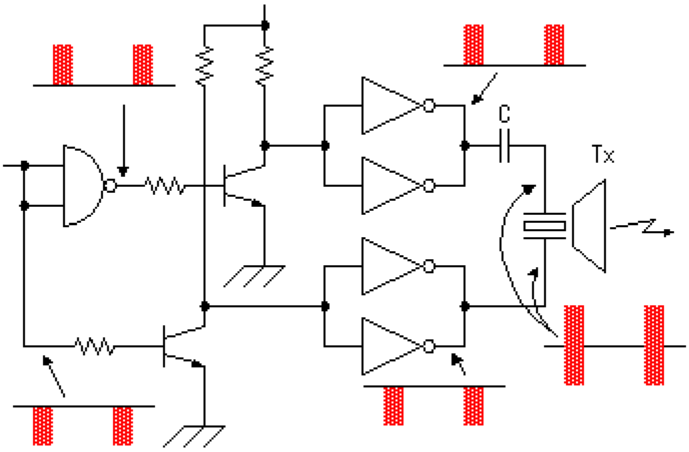
OSCILLATOR CIRCUIT II

The oscillator circuit is taken from the DIGIAC2000's application module circuit diagram.



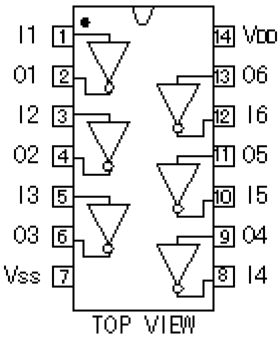
When we used these circuits, to calculate the distance we had to use the change of amplitudes of the reflected signals. The noise could not be reduced for healthy calculation, so we decided to use PIC16F873 for oscillation. PIC16F873 calculates the distance by using time of flight principle.

THE TRANSMITTER CIRCUIT



The inverter is used for the drive of the ultrasonic sensor. The two inverters are connected in parallel because of the transmission electric power increase.
The phase with the voltage to apply to the positive terminal and the negative terminal of the sensor has been 180 degrees shifted. Because it is cutting the direct current with the capacitor, about twice of voltage of the inverter output are appied to the sensor.

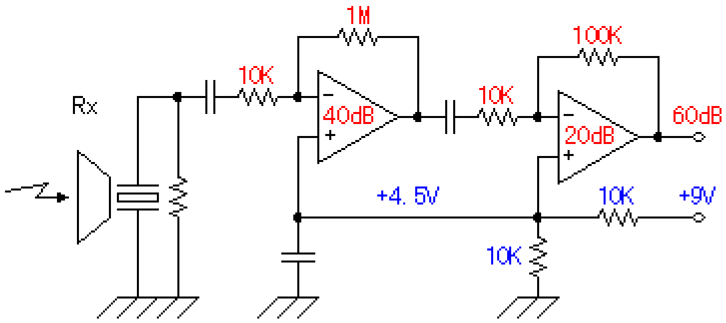
The power supply voltage of this drive circuit is +9V. It is converting voltage with the transistor to make control at the operating voltage of PIC(+5V). Because C-MOS inverters are used, it is possible to do ON/OFF at high speed comparatively.



This IC is the IC of the CMOS which the six inverters are housed in. At the transmitter circuit, it is used for the drive circuit of the ultrasonic sensor.

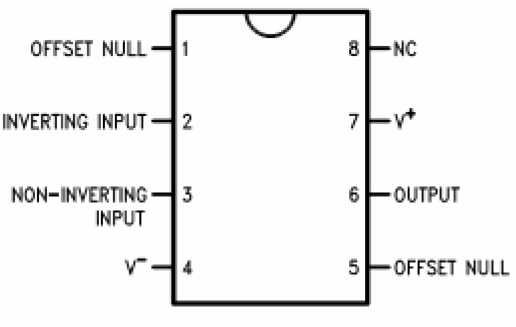
THE RECEIVER PART

SIGNAL AMPLIFICATION CIRCUIT



The ultrasonic signal which was received with the reception sensor is amplified by 1000 times(60dB) of voltage with the operational amplifier with two stages. It is 100 times at the first stage (40dB) and 10 times (20dB) at the next stage.

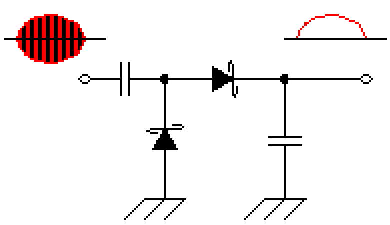
Generally, the positive and the negative power supply are used for the operational amplifier. The circuit this time works with the single power supply of +9 V. Therefore, for the positive input of the operational amplifiers, the half of the power supply voltage is appied as the bias voltage. Then the alternating current signal can be amplified on 4.5V central voltage. When using the operational amplifier with the negative feedback, the voltage of the positive input terminal and the voltage of the negative input terminal become equal approximately. This is called virtual grounding. So, by this bias voltage, the side of the positive and the side of the negative of the alternating current signal can be equally amplified. When not using this bias voltage, the distortion causes the alternating current signal. This technique is often used when using the operational amplifier which needs two kinds of powers in the single power.
We used two LM741 low noise operational amplifiers.



This IC is the low noise operational amplifier. It is used for the amplification of the received ultrasonic signal.
The low noise type operational amplifier should be used because it does the about 60dB (1000 times) amplification.

ENVELOPE DETECTOR

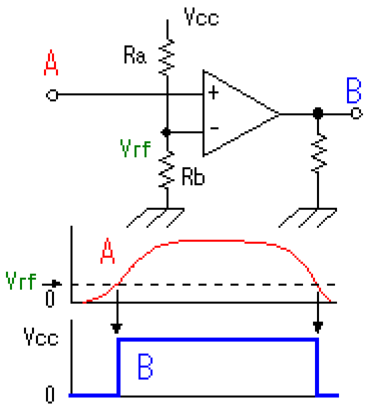
DETECTION CIRCUIT



The detection is done to detect the received ultrasonic signal. This is the half-wave rectification circuit with Shottky barrier diodes. The DC voltage according to the level of the detection signal is output to the capacitor behind the diode. The Shottky barrier diodes are used because the high frequency characteristic is good.

These diodes are used to detect the received ultrasonic. The ultrasonic frequency is about 40KHz, so, the diode with the good high frequency characteristic is used.

SIGNAL DETECTOR

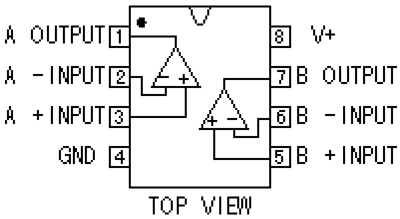


This circuit is the circuit which detects the ultrasonic which returned from the measurement object. The output of the detection circuit is detected using the comparator. At the circuit this time, the operational amplifier of the single power supply is used instead of the comparator. The operational amplifier amplifies and outputs the difference between the positive input and the negative input.
In case of the operational amplifier which doesn't have the negative feedback, the output becomes the saturation state by a little input voltage. Generally, the operational amplifier has over 10000 times of mu factors. So, when the positive input becomes higher a little than the negative input, the difference is tens of thousands of times amplified and the output becomes the same as the power supply almost.(It is the saturation state) Oppositely, when the positive input becomes lower a little than the negative input, the difference is tens of thousands of times amplified and the output becomes 0 V almost.(It is in the OFF condition) This operation is the same as the operation of the comparator. However, because the inner circuit of the comparator is different from the operational amplifier, the comparator can not be used as the operational amplifier.

At the circuit this time, the output of the detection circuit is connected with the positive input of the signal detector and the voltage of the negative input is made constant.

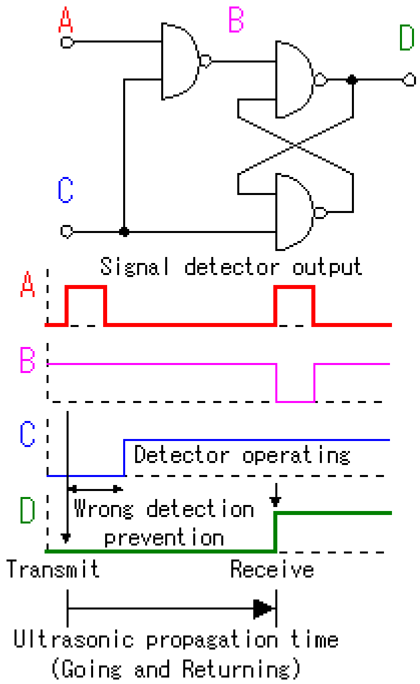
$$\mathbf{Vrf} = (\text{Rb} \times \text{Vcc}) / (\text{Ra} + \text{Rb})$$
$$= (47^{\text{K-ohm}} \times 9^{\text{V}}) / (1^{\text{M-ohm}} + 47^{\text{K-ohm}})$$
$$= \mathbf{0.4V}$$

So, when the rectified ultrasonic signal becomes more than 0.4 V, the output of the signal detector becomes the H level (Approximately 9V).
This output is lowered with the resistor to make fit with the input of signal holding circuit (TTL:0V to 5V).

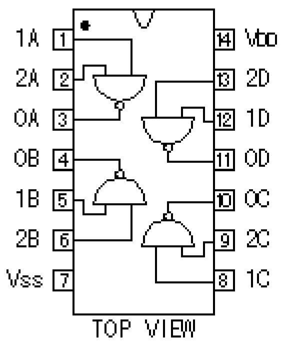


This IC is the single power supply-type operational amplifier.
This IC is used for the detection of the received signal.

SIGNAL HOLDING CIRCUIT

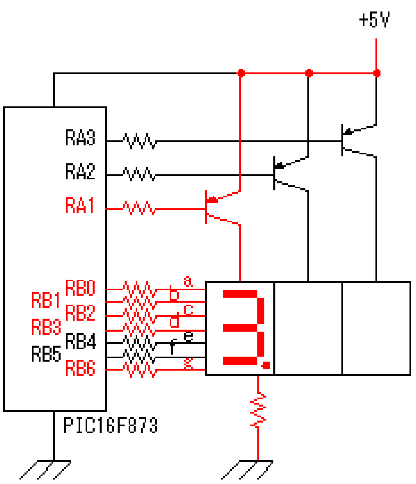


This is the holding circuit of detected signal. RS (Set and Reset) flip-flop is used.
The detector is made to be not operate in the constant time(About 1.5 milliseconds) after sending out a transmission pulse to prevent from the wrong detection which is due to the influence of the transmission pulse. This operation is controlled with the software of PIC.
When using the capture feature of PIC, this circuit isn't indispensable. Capture operation is done by the change of the capture input in the once. The reason for using this circuit is to confirm signal detection operation within the reflected signal detection time(About 65 milliseconds). When sending out next ultrasonic pulse, the output of this circuit is checked. And when the output is L level, an error display is done because the reflected signal could not be detected.

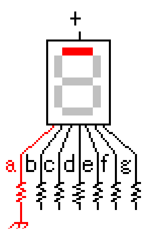


As for this IC, the four NAND circuits of 2 inputs are accommodated. It is used to compose SR-FF and to hold the detection condition of the ultrasonic.

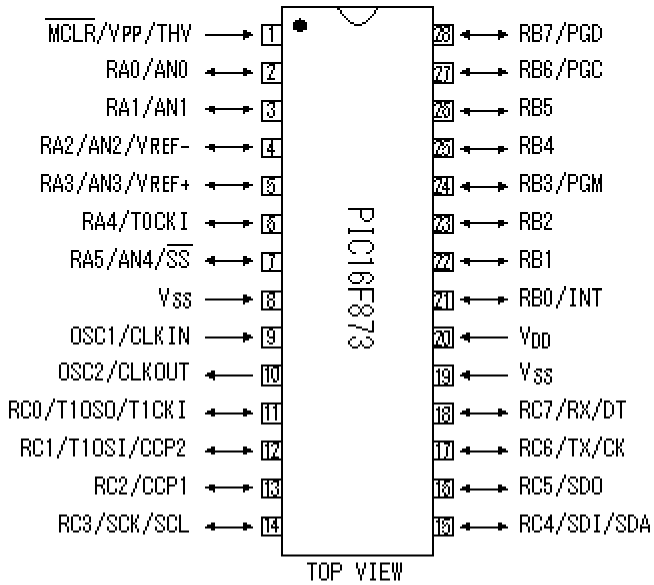
SEVEN SEGMENT DISPLAY CIRCUIT



Three 7 segment LEDs are used for 3-digit display. As for the lighting-up of the LED, 1 digit is displayed in the order with the software of PIC.
At the circuit this time, I make light up it when the terminal of PIC is L level. So, anode common type is used as the LED. The anode common type is the type which the side of the positive(Anode) of the LED is connected inside. It lights up when grounding(L level) a cathode in the segment to want to make light up.
As the 7 segment LED, the others have a cathode common type. When you buy them, the specification of the type should be checked.

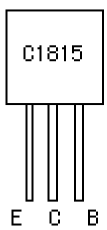


PIC16F873



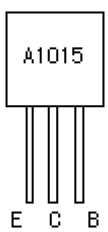
In the circuit this time, capture feature and A/D converter feature are used.

Transmitter drive transistor (2SC1815)



This is the transistor to drive the C-MOS inverter which works at 9V with the output of PIC. The output of PIC is from 0V to 5V. This transistor converts into the voltage from 0V to 9V to control the inverter.

LED control transistor (2SA1015)



This transistor is used to control the 7 segment LED. PNP type is used for controlling the anode side of the LED.

Crystal

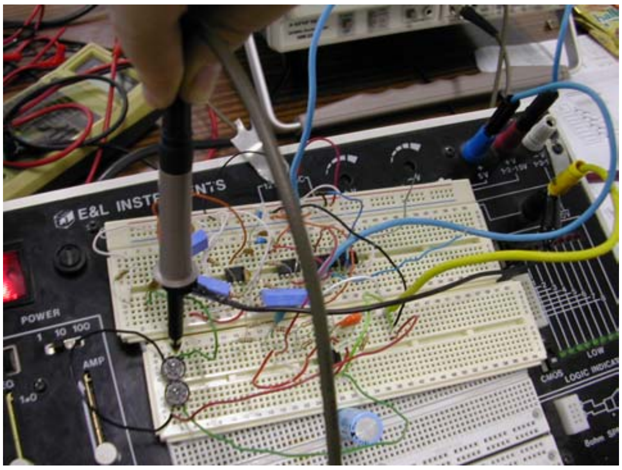
4 MHz crystal is used for triggering PIC16F873. The timing is very important in this project, so we did not used RC oscillator.

Resistors

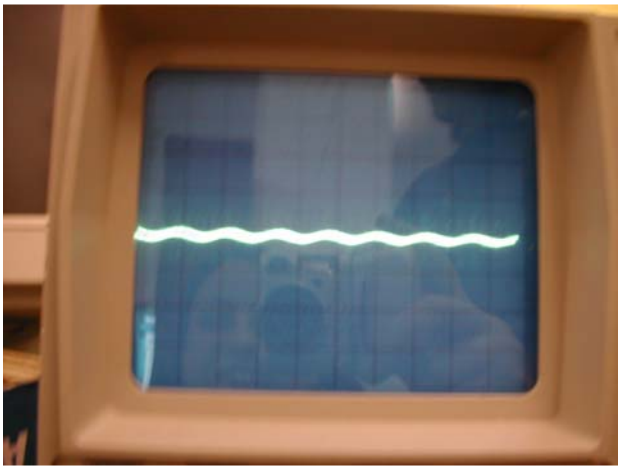
We used 1/8W as all resistors in different values; 10KΩ, 1MΩ, 100KΩ, 47KΩ, 5.6KΩ, 4.7KΩ, 330Ω, 1KΩ.

RESULTS

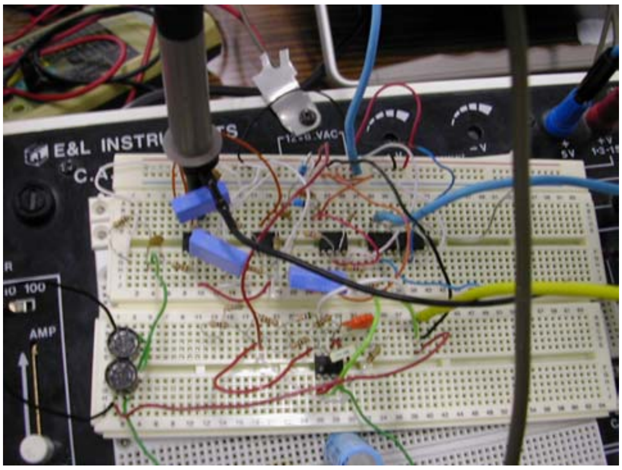
These pictures are taken with a system that gives pulses continuously. In our final design PIC triggers the transmitter circuit for 12 µmsec.



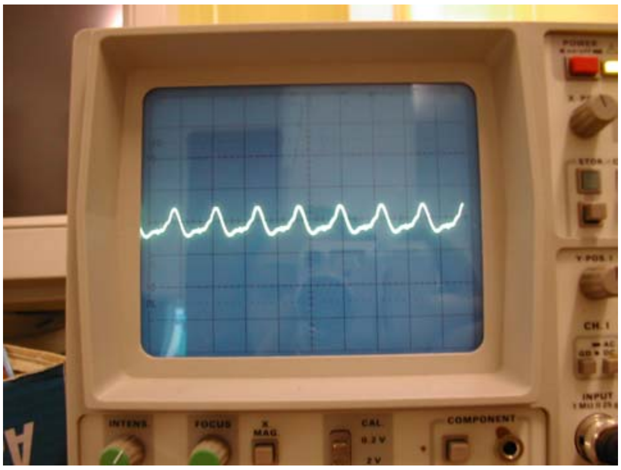
The reciever part of the transducer



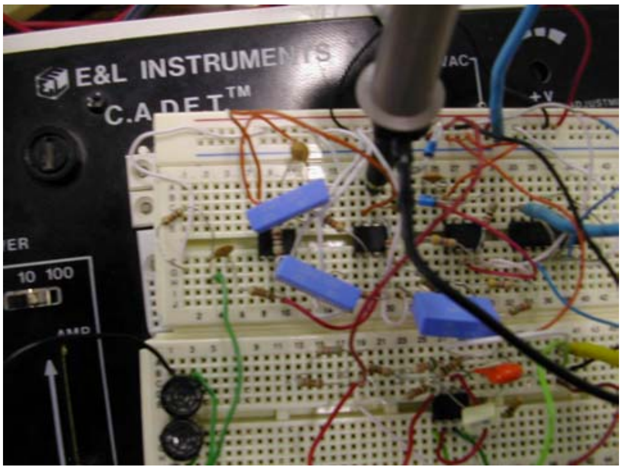
Output at the reciever part of the transducer



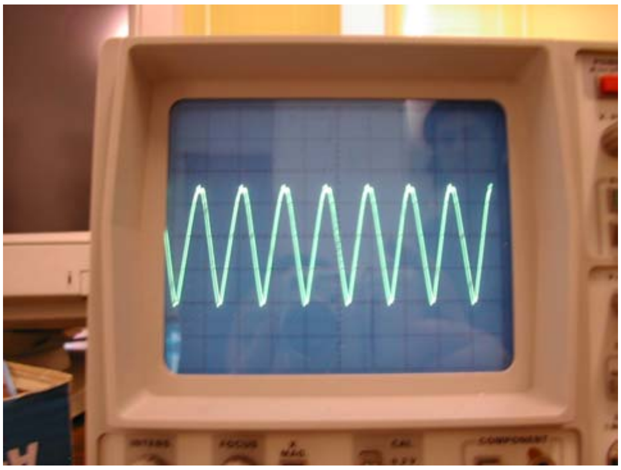
The output of the first LM 741 Opamp



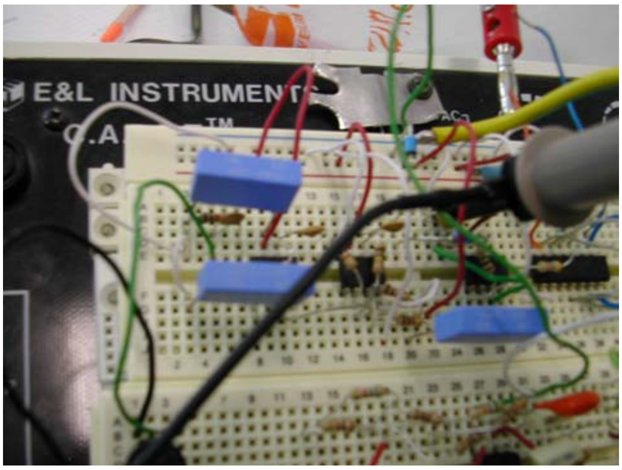
The output signal of the first LM 741 Opamp



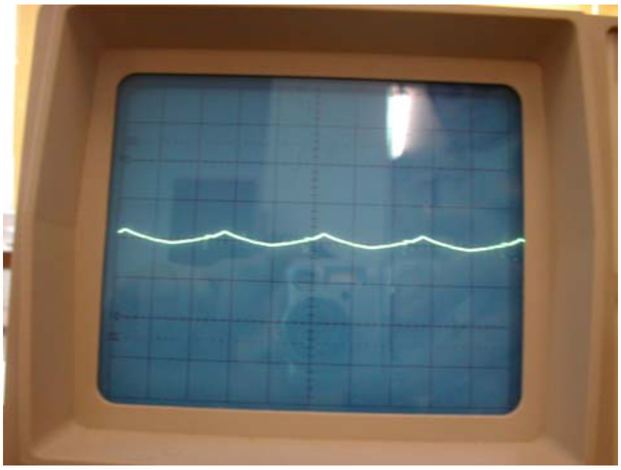
The output of the second LM 741 Opamp



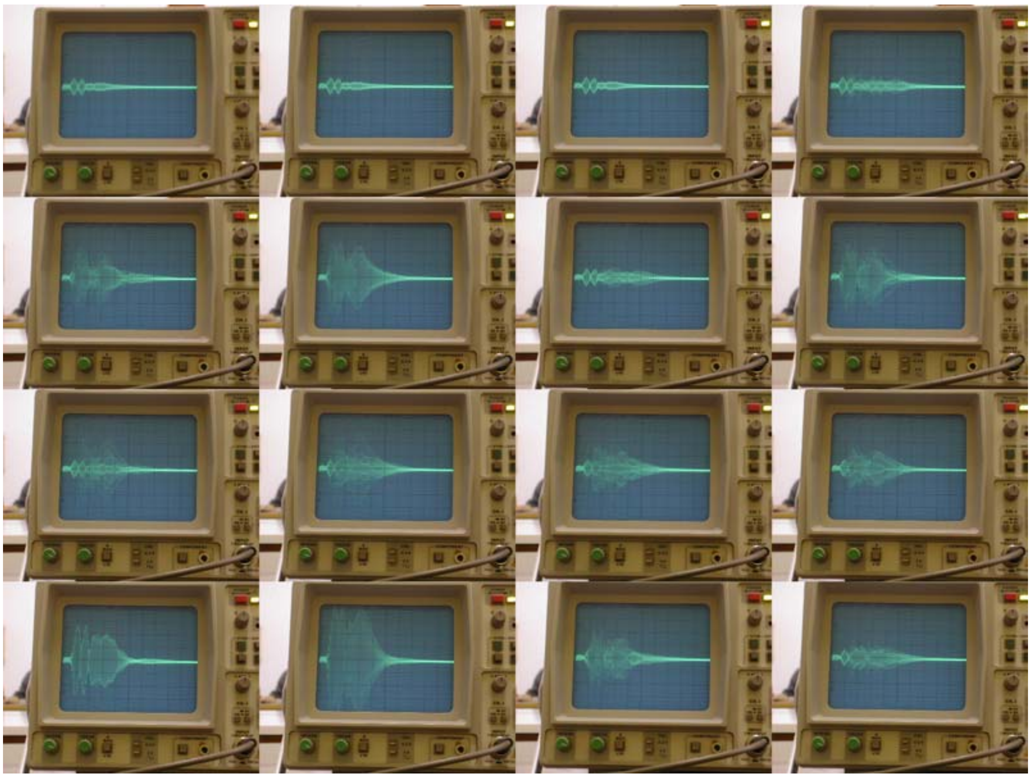
The output signal of the second LM 741 Opamp



The output of the detection circuit
(at the output shottky diodes)



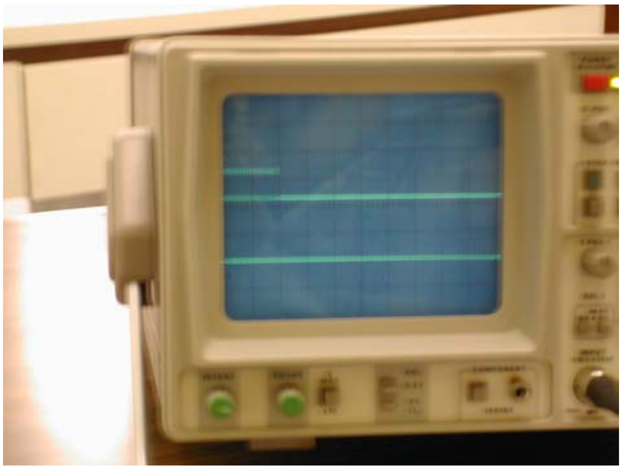
The output signal of the detection circuit.
(at the output shottky diodes)



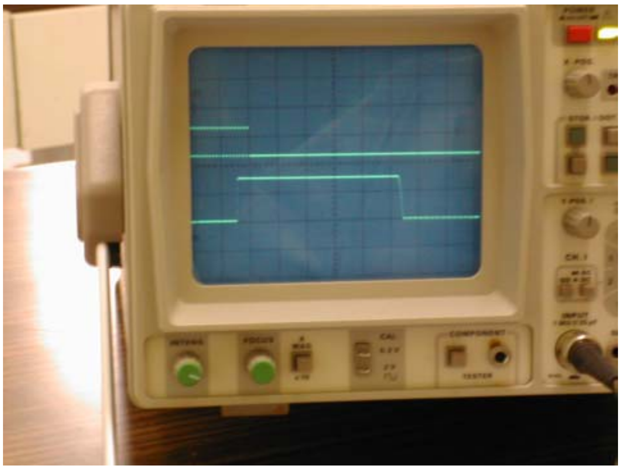
The output of the second LM 741 Opamp

This signal changes as an object gets closer.

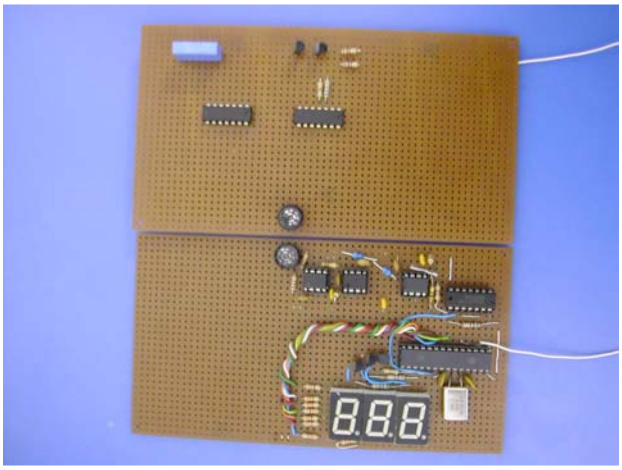
These pictures are taken after the soldering process.



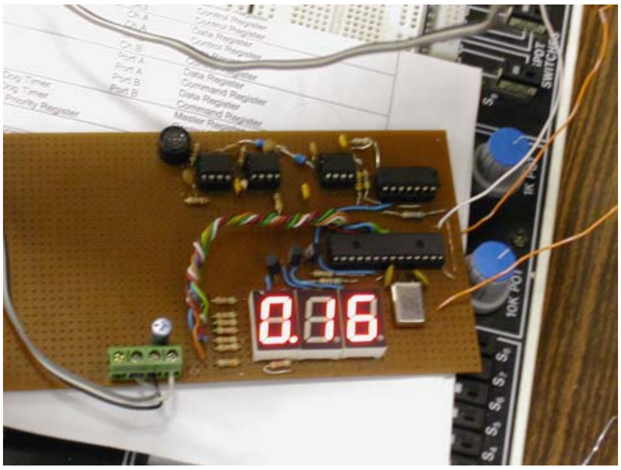
The transmitted signal.
(at the output of the transmitter)



The output signal of the detection circuit.
(at the output LM 358)



Ultrasonic Range Meter
(The card above is the sender part. The below card is reciever part)



Ultrasonic Range Meter
(Working!)

The circuit design we used as reference was designed to measure the long distances, but we could not measure the long distances because of the transducer's characteristics. To modify the circuit for short distances, we edited the code for the PIC16F873. In the original code, the RS flipflop was setting so later that the signal was transmitted and reflected back before the RS flipflop was enabled. This problem disables the calculation process start. To solve this problem, we decreased the capture guard timer counter 124x2 to 10x2 and the pulse count 20 to 10. By this way, the range of measured distance was 7 cm to 26 cm.

In order to use this range meter effectively,

- To measure long distances, different transducers can be used.
- To reduce the noise and prevent problems caused by short circuit, printed circuit board (pcb) should be used.
- To change the distance range, in the code the capture guard timer count and the pulse count can be modified.
- To calibrate the displays, a 1K potentiometer cuold be used in the PIC16F873's RA0 input.

are advised.

The Code

```
;********************************************************
;
;                 Ultrasonic Range Meter
;
;                                 Device : PIC16F873
;
;********************************************************

        list            p=pic16f873
        include         p16f873.inc
       __config _hs_osc & _wdt_off & _pwrte_on & _lvp_off

;****************  Label Definition  ********************
        cblock  h'20'
s_count                            ;Send-out pulse count adr
s_adj                              ;Adjustment data address
s_adj_count                        ;Rotate value save adr
s_digit                            ;Digit cont work address
g_time1                            ;Guard timer address 1
g_time2                            ;Guard timer address 2
p_countl                           ;Propagation L cnt adr
p_counth                           ;Propagation H cnt adr
digit_cnt                          ;Digit counter head adr
disp_ha                            ;Digit head address
disp_u                             ;1st digit address
disp_t                             ;10th digit address
disp_h                             ;100th digit address
seg7_ha                            ;7 segLED table head adr
seg70                              ;Pattern 0 set adr
seg71                              ;Pattern 1 set adr
seg72                              ;Pattern 2 set adr
seg73                              ;Pattern 3 set adr
seg74                              ;Pattern 4 set adr
seg75                              ;Pattern 5 set adr
seg76                              ;Pattern 6 set adr
seg77                              ;Pattern 7 set adr
seg78                              ;Pattern 8 set adr
seg79                              ;Pattern 9 set adr
seg7a                              ;Pattern A set adr
seg7b                              ;Pattern B set adr
        endc

ra1     equ     h'01'              ;RA1 port designation
ra2     equ     h'02'              ;RA2 port designation
ra3     equ     h'03'              ;RA3 port designation
ra5     equ     h'05'              ;RA5 port designation

ccp1    equ     h'02'              ;CCP1(RC2) designation

seg7_0  equ     b'01000000'        ;-gfedcba Pattern 0
seg7_1  equ     b'01111001'        ;         Pattern 1
seg7_2  equ     b'00100100'        ;         Pattern 2
seg7_3  equ     b'00110000'        ;         Pattern 3
seg7_4  equ     b'00011001'        ;         Pattern 4
seg7_5  equ     b'00010010'        ;         Pattern 5
seg7_6  equ     b'00000010'        ;         Pattern 6
seg7_7  equ     b'01111000'        ;         Pattern 7
seg7_8  equ     b'00000000'        ;         Pattern 8
seg7_9  equ     b'00010000'        ;         Pattern 9
```

```
seg7_a  equ     b'01111111'     ;         Detect error
seg7_b  equ     b'00100011'     ;         Illegal int

;****************  Program Start  ***********************
        org     0               ;Reset Vector
        goto    init
        org     4               ;Interrupt Vector
        goto    int

;****************  Initial Process  *********************
init

;*** Port initialization
        bsf     status,rp0      ;Change to Bank1
        movlw   b'00000001'     ;AN0 to input mode
        movwf   trisa           ;Set TRISA register
        clrf    trisb           ;RB port to output mode
        movlw   b'00000100'     ;RC2/CCP1 to input mode
        movwf   trisc           ;Set TRISC register

;*** Ultrasonic sending period initialization (Timer0)
        movlw   b'11010111'     ;T0CS=0,PSA=0,PS=1:256
        movwf   option_reg      ;Set OPTION_REG register
        bcf     status,rp0      ;Change to Bank0
        clrf    tmr0            ;Clear TMR0 register

;*** Capture mode initialization (Timer1)
        movlw   b'00000001'     ;Pre=1:1 TMR1=Int TMR1=ON
        movwf   t1con           ;Set T1CON register
        clrf    ccp1con         ;CCP1 off

;*** A/D converter initialization
        movlw   b'01000001'     ;ADCS=01 CHS=AN0 ADON=ON
        movwf   adcon0          ;Set ADCON0 register
        bsf     status,rp0      ;Change to Bank1
        movlw   b'00001110'     ;ADFM=0 PCFG=1110
        movwf   adcon1          ;Set ADCON1 register
        bcf     status,rp0      ;Change to Bank0

;*** Display initialization (Timer2)
        movlw   disp_u          ;Set digit head address
        movwf   disp_ha         ;Save digit head sddress
        movlw   h'0a'           ;"Detect error" data
        movwf   disp_u          ;Set 1st digit
        movwf   disp_t          ;Set 10th digit
        movwf   disp_h          ;Set 100th digit
        movlw   d'3'            ;Digit counter
        movwf   digit_cnt       ;Set digit counter
        movlw   seg70           ;Set 7seg head address
        movwf   seg7_ha         ;Save 7seg head address
        movlw   seg7_0          ;Set 7segment pattern 0
        movwf   seg70           ;Save pattern 0
        movlw   seg7_1          ;Set 7segment pattern 1
        movwf   seg71           ;Save pattern 1
        movlw   seg7_2          ;Set 7segment pattern 2
        movwf   seg72           ;Save pattern 2
        movlw   seg7_3          ;Set 7segment pattern 3
        movwf   seg73           ;Save pattern 3
        movlw   seg7_4          ;Set 7segment pattern 4
        movwf   seg74           ;Save pattern 4
        movlw   seg7_5          ;Set 7segment pattern 5
```

```
        movwf   seg75           ;Save pattern 5
        movlw   seg7_6          ;Set 7segment pattern 6
        movwf   seg76           ;Save pattern 6
        movlw   seg7_7          ;Set 7segment pattern 7
        movwf   seg77           ;Save pattern 7
        movlw   seg7_8          ;Set 7segment pattern 8
        movwf   seg78           ;Save pattern 8
        movlw   seg7_9          ;Set 7segment pattern 9
        movwf   seg79           ;Save pattern 9
        movlw   seg7_a          ;Set 7segment pattern A
        movwf   seg7a           ;Save pattern A
        movlw   seg7_b          ;Set 7segment pattern B
        movwf   seg7b           ;Save pattern B
        movlw   b'00011110'     ;OPS=1:4,T2=ON,EPS=1:16
        movwf   t2con           ;Set T2CON register
        bsf     status,rp0      ;Change to Bank1
        movlw   d'157'          ;157x64=10048usec
        movwf   pr2             ;Set PR2 register
        bsf     pie1,tmr2ie     ;TMR2IE=ON
        bcf     status,rp0      ;Change to Bank0


;*** Interruption control
        movlw   b'11100000'     ;GIE=ON,PEIE=ON,T0IE=ON
        movwf   intcon          ;Set INTCON register

wait
        goto    $               ;Interruption wait

;***************  Interruption Process  *****************
int
        movfw   pir1            ;Read PIR1 register
        btfsc   pir1,ccp1if     ;Capture occurred ?
        goto    capture         ;Yes. "Capture"
        btfsc   pir1,tmr2if     ;TMR2 time out ?
        goto    led_cont        ;Yes. "LED display"
        movfw   intcon          ;Read INTCON register
        btfsc   intcon,t0if     ;TMR0 time out ?
        goto    send            ;Yes. "Pulse send"

;***************  Illegal interruption  *****************
illegal
        movlw   h'0b'           ;Set Illegal disp digit
        addwf   seg7_ha,w       ;Seg7 H.Adr + digit
        movwf   fsr             ;Set FSR register
        movfw   indf            ;Read seg7 data
        movwf   portb           ;Write LED data
        bcf     porta,ra1       ;RA1=ON
        bcf     porta,ra2       ;RA2=ON
        bcf     porta,ra3       ;RA3=ON
        goto    $               ;Stop

;************  END of Interruption Process **************
int_end
        retfie

;***************  Pulse send-out Process ****************
send
        bcf     intcon,t0if     ;Clear TMR0 int flag
        clrf    tmr0            ;Timer0 clear

```

```
;*** Received Pulse detection check
        movfw   portc           ;Read PORTC register
        btfsc   portc,ccp1      ;Detected ?
        goto    detect_off      ;Yes. Detected
        movlw   h'0a'           ;"Detect error" data
        movwf   disp_u          ;Set 1st digit
        movwf   disp_t          ;Set 10th digit
        movwf   disp_h          ;Set 100th digit

;*** Receive pulse detector off
detect_off
        bcf     porta,ra5       ;Set detector OFF

;*** Capture start
        clrf    tmr1h           ;Clear TMR1H register
        clrf    tmr1l           ;Clear TMR1L register
        clrf    ccpr1h          ;Clear CCPR1H register
        clrf    ccpr1l          ;Clear CCPR1L register
        movlw   b'00000101'     ;CCP1M=0101(Capture)
        movwf   ccp1con         ;Set CCP1CON register
        bsf     status,rp0      ;Change to Bank1
        bsf     pie1,ccp1ie     ;CCP1 interruptin enable
        bcf     status,rp0      ;Change to Bank0
        bcf     pir1,ccp1if     ;Clear CCP1 int flag

;*** 40KHz pulse send ( 0.5 msec )
        movlw   d'10'           ;Send-out pulse count
        movwf   s_count         ;Set count
s_loop
        call    pulse           ;Call pulse send sub
        decfsz  s_count,f       ;End ?
        goto    s_loop          ;No. Continue

;*** Get adjustment data
        bsf     adcon0,go       ;Start A/D convert
ad_check
        btfsc   adcon0,go       ;A/D convert end ?
        goto    ad_check        ;No. Again
        movfw   adresh          ;Read ADRESH register
        movwf   s_adj           ;Save converted data

        movlw   d'5'            ;Set rotate value
        movwf   s_adj_count     ;Save rotate value
ad_rotate
        rrf     s_adj,f         ;Rotate right 1 bit
        decfsz  s_adj_count,f   ;End ?
        goto    ad_rotate       ;No. Continue
        movfw   s_adj           ;Read rotated value
        andlw   b'00000111'     ;Pick-up 3 bits
        addlw   d'54'           ;(0 to 7) + 54 = 54 to 61
        movwf   s_adj           ;Save adjustment data

;*** Capture guard timer ( 1 milisecound )
        movlw   d'2'            ;Set loop counter1
        movwf   g_time1         ;Save loop counter1
g_loop1 movlw   d'10'          ;Set loop counter2
        movwf   g_time2         ;Save loop counter2
g_loop2 nop                     ;Time adjust
        decfsz  g_time2,f       ;g_time2 - 1 = 0 ?
        goto    g_loop2         ;No. Continue
        decfsz  g_time1,f       ;g_time1 - 1 = 0 ?
```

```
        goto    g_loop1         ;No. Continue

;*** Receive pulse detector on
        bsf     porta,ra5       ;Set detector ON

        goto    int_end

;***************  Pulse send-out Process ***************
pulse
        movlw   b'00010000'     ;RC4=ON
        movwf   portc           ;Set PORTC register
        call    t12us           ;Call 12usec timer
        clrf    portc           ;RC4=OFF
        goto    $+1
        goto    $+1
        nop
        return

;***************  12 microseconds timer ****************
t12us
        goto   $+1
        goto   $+1
        goto   $+1
        goto   $+1
        nop
        return

;******************  Capture Process *******************
capture
        bcf     pir1,ccp1if     ;Clear CCP1 int flag

        clrf    p_countl        ;Clear L count
        clrf    p_counth        ;Clear H count
        clrf    ccp1con         ;CCP1 off

division
        movfw   s_adj           ;Read adjustment data
        subwf   ccpr1l,f        ;Capture - adjust
        btfsc   status,z        ;Result = 0 ?
        goto    division2       ;Yes. "R = 0"
        btfsc   status,c        ;Result < 0 ?
        goto    division1       ;No. "R > 0"
        goto    division3       ;Yes."R < 0"

division1                       ;( R > 0 )
        movlw   d'1'            ;Set increment value
        addwf   p_countl,f      ;Increment L count
        btfss   status,c        ;Overflow ?
        goto    division        ;No. Continue
        incf    p_counth,f      ;Increment H count
        goto    division        ;Jump next

division2                       ;( R = 0 )
        movfw   ccpr1h          ;Read CCPR1H
        btfss   status,z        ;CCPR1H = 0 ?
        goto    division1       ;No. Next
        movlw   d'1'            ;Set increment value
        addwf   p_countl,f      ;Increment L count
        btfss   status,c        ;Overflow ?
        goto    digit_set       ;Jump to digit set
        incf    p_counth,f      ;Increment H count
```

```
        goto    digit_set       ;Jump to digit set

division3                       ;( R < 0 )
        movfw   ccpr1h          ;Read CCPR1H
        btfss   status,z        ;CCPR1H = 0 ?
        goto    division4       ;No. Borrow process
        goto    digit_set       ;Jump to digit set

division4
        decf    ccpr1h,f        ;CCPR1H - 1
        movlw   d'255'          ;Borrow value
        addwf   ccpr1l,f        ;CCPR1L + 255
        incf    ccpr1l,f        ;CCPR1L + 1
        goto    division1       ;Next

;****************  Digit Set Process *******************
digit_set
        clrf    disp_u          ;Clear 1st digit
        clrf    disp_t          ;Clear 10th digit
        clrf    disp_h          ;Clear 100th digit

;*** 100th digit
digit_h
        movlw   d'100'          ;Divide value
        subwf   p_countl,f      ;Digit - divide
        btfsc   status,z        ;Result = 0 ?
        goto    digit_h2        ;Yes. "R = 0"
        btfsc   status,c        ;Result < 0 ?
        goto    digit_h1        ;No. "R > 0"
        goto    digit_h3        ;Yes."R < 0"

digit_h1                        ;( R > 0 )
        incf    disp_h,f        ;Increment 100th count
        goto    digit_h         ;Jump next

digit_h2                        ;( R = 0 )
        movfw   p_counth        ;Read H counter
        btfss   status,z        ;H counter = 0 ?
        goto    digit_h1        ;No. Next
        incf    disp_h,f        ;Increment 100th count
        goto    digit_t         ;Jump to 10th digit pro

digit_h3                        ;( R < 0 )
        movfw   p_counth        ;Read H counter
        btfss   status,z        ;H counter = 0 ?
        goto    digit_h4        ;No. Borrow process
        movlw   d'100'          ;Divide value
        addwf   p_countl,f      ;Return over sub value
        goto    digit_t         ;Jump to 10th digit pro

digit_h4
        decf    p_counth,f      ;H counter - 1
        movlw   d'255'          ;Borrow value
        addwf   p_countl,f      ;L counter + 255
        incf    p_countl,f      ;L counter + 1
        goto    digit_h1        ;Next

;*** 10th digit
digit_t

;*** Range over check
```

```
        movfw   disp_h          ;Read 100th digit
        sublw   d'9'            ;9 - (100th digit)
        btfsc   status,z        ;Result = 0 ?
        goto    digit_t0        ;Yes. "R = 0"
        btfsc   status,c        ;Result < 0 ?
        goto    digit_t0        ;No. "R > 0"
        movlw   h'0a'           ;"Detect error" data
        movwf   disp_u          ;Set 1st digit
        movwf   disp_t          ;Set 10th digit
        movwf   disp_h          ;Set 100th digit
        goto    int_end

digit_t0
        movlw   d'10'           ;Divide value
        subwf   p_countl,f      ;Digit - divide
        btfsc   status,z        ;Result = 0 ?
        goto    digit_t1        ;Yes. "R = 0"
        btfsc   status,c        ;Result < 0 ?
        goto    digit_t1        ;No. "R > 0"
        goto    digit_t2        ;Yes."R < 0"

digit_t1                        ;( R >= 0 )
        incf    disp_t,f        ;Increment 10th count
        goto    digit_t         ;Jump next

digit_t2                        ;( R < 0 )
        movlw   d'10'           ;Divide value
        addwf   p_countl,f      ;Return over sub value
        goto    digit_u         ;Jump to 1st digit pro

;*** 1st digit
digit_u
        movfw   p_countl        ;Read propagetion counter
        movwf   disp_u          ;Save 1st count

        goto    int_end

;****************  LED display control  *****************
led_cont
        bcf     pir1,tmr2if     ;Clear TMR2 int flag

        movfw   digit_cnt       ;Read digit counter
        movwf   s_digit         ;Save digit counter
        decfsz  s_digit,f       ;1st digit ?
        goto    d_check1        ;No. Next
        bsf     porta,ra1       ;RA1=OFF
        bsf     porta,ra2       ;RA2=OFF
        bcf     porta,ra3       ;RA3=ON
        goto    c_digit         ;Jump to digit cont
d_check1
        decfsz  s_digit,f       ;10th digit ?
        goto    d_check2        ;No. 100th digit
        bsf     porta,ra1       ;RA1=OFF
        bcf     porta,ra2       ;RA2=ON
        bsf     porta,ra3       ;RA3=OFF
        goto    c_digit         ;Jump to digit cont
d_check2
        bcf     porta,ra1       ;RA1=ON
        bsf     porta,ra2       ;RA2=OFF
        bsf     porta,ra3       ;RA3=OFF
```

```
c_digit
        decf    digit_cnt,w      ;Digit count - 1
        addwf   disp_ha,w        ;Digit H.Adr + count
        movwf   fsr              ;Set FSR register
        movfw   indf             ;Read digit
        addwf   seg7_ha,w        ;Seg7 H.Adr + digit
        movwf   fsr              ;Set FSR register
        movfw   indf             ;Read seg7 data
        movwf   portb            ;Write LED data

        decfsz  digit_cnt,f      ;Digit count - 1
        goto    int_end          ;Jump to interrupt end
        movlw   d'3'             ;Initial value
        movwf   digit_cnt        ;Set initial value
        goto    int_end          ;Jump to interrupt end

;********************************************************
;            END of Ultrasonic Range Meter
;********************************************************

        end
```

B – DATASHEETS

Data sheets of the components that we used in the circuit is also in the DATASHEETS directory at the root directory of the CD. These files can be accessed via the links below and directly from windows explorer.

- PIC 16F87X
- muRata MA40S4R
- muRata MA40S4S
- BVY10 - Shottky Barrier Diode
- 2SA1015 – Transistor
- 2SC1815 – Transistor
- HCF4011B – NAND Gates
- HCF4069UB – Hex Inverter
- LM358 – Op Amp
- LM741 – Op Amp

C – DOCUMENTS AFTER RESEARCH

Documents that we found while searching are in the DOCUMENTS directory at the root directory of the CD