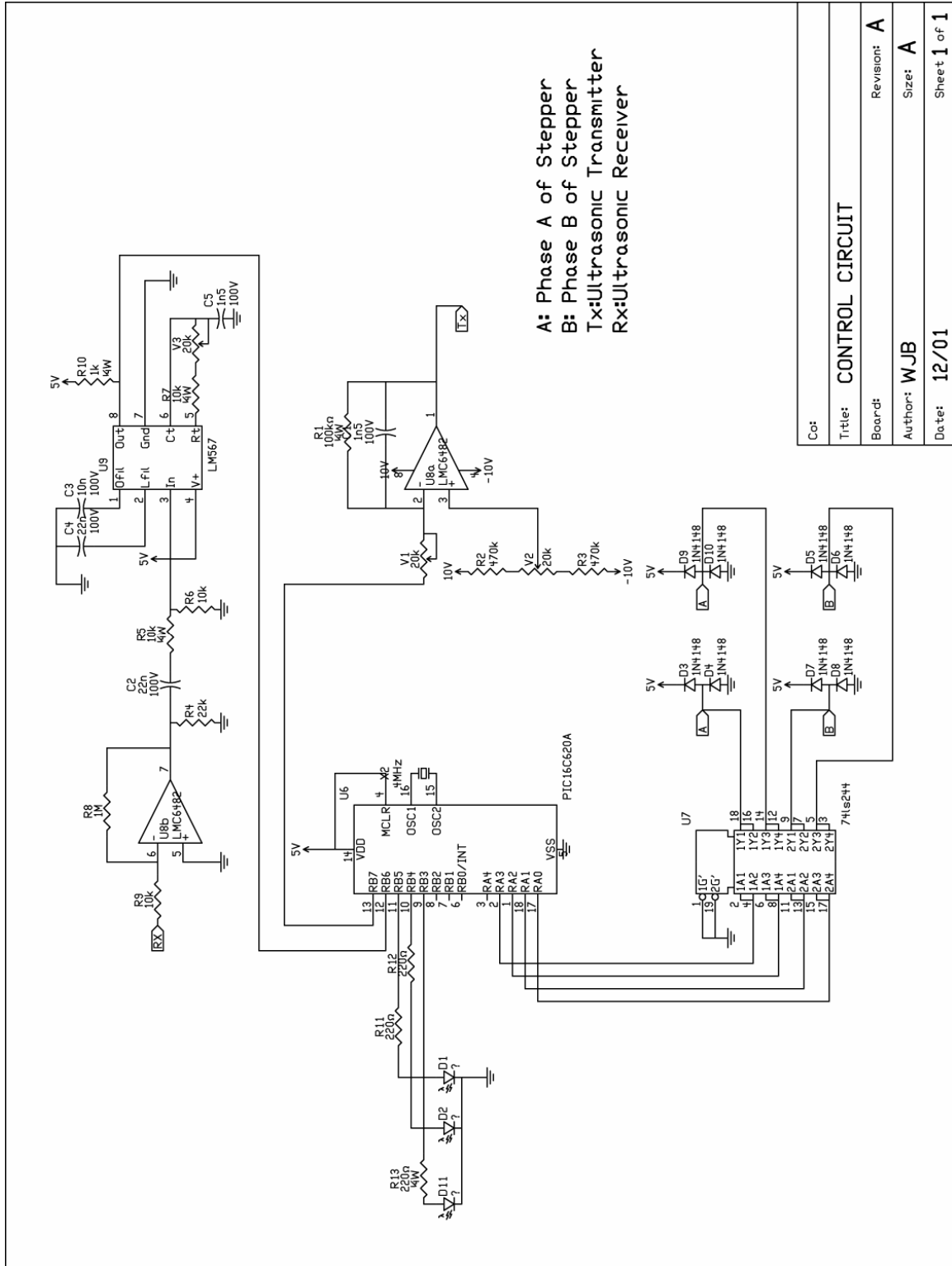
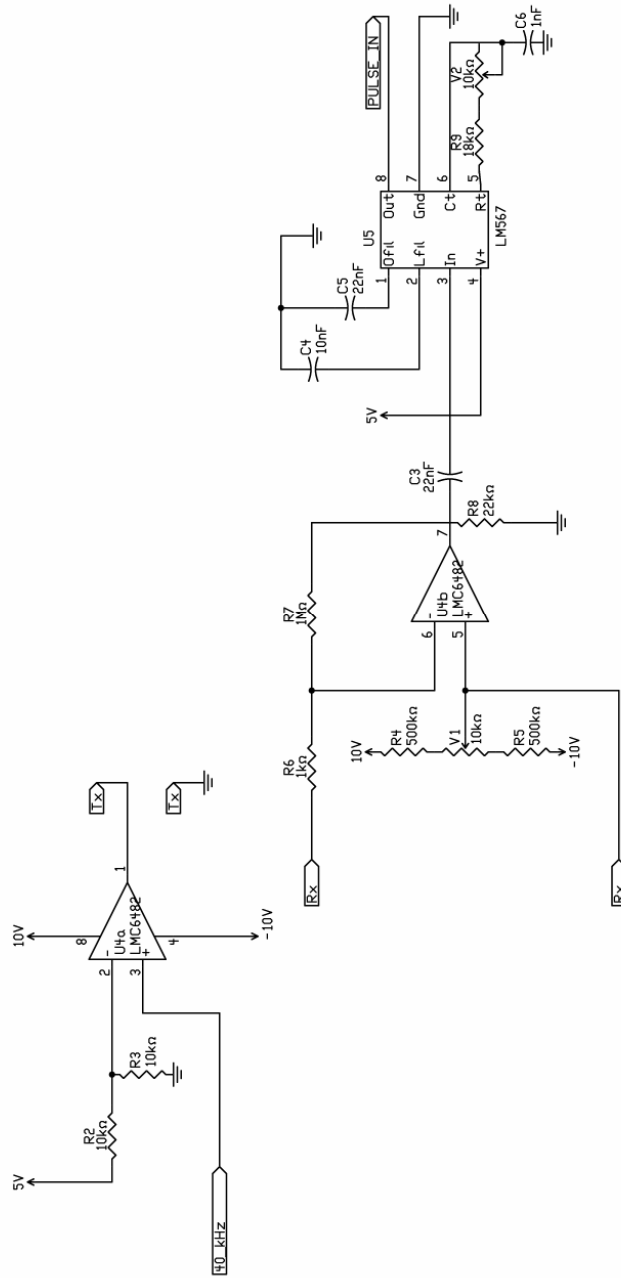


CIRCUIT DIAGRAM A



Cot:	
Title:	CONTROL CIRCUIT
Board:	
Author:	WJB
Date:	12/01
Revision:	A
Size:	A
Sheet	1 of 1

CIRCUIT DIAGRAM B



Co:	
Title:	FIGURE 14:CONTROL CIRCUIT PART B
Board:	Revision: A
Author:	WJB Size: A
Date:	12/01 Sheet 1 of 1

ASSEMBLY CODE for PIC16x

```
;;;;;;;;;;;;;Object Ranger;;;;;;;;;;;;;
;
;Toggle PORTA,0 , through an op amp
;drives ultrasonic transducer. After sufficient dead time for ringing
;a 1:32 scaled TIMER0 counts. When an interrupt is received, a table
;look up and output gives appreciation for distance. Alternatively
;TIMER0 may overflow indicating that there is no object detected and
;then a STEP signal is sent. When ENABLE is received, it's ok to gate
;the oscillator and the process is repeated.
;
;When writing to table, two error detection implementations exist:
;*current value must be lower than last as object is approaching.
;*current value cannot differ from last by more than 1cm, which
;if it were would correspond to the object moving at more than
;5km per hour
;
;WJB 12/01
;;;;;;;;;;;;;

list P=PIC16C620A, C=160, N=80, ST=OFF, MM=OFF, R=DEC
include "C:\PROGRA~1\MPLAB\P16C620A.INC"
__config (_CP_OFF & _PWRTE_ON & _XT_OSC & _WDT_OFF & _BODEN_OFF)
errorlevel -302

;;;;;;;;;;;;;variables;;;;;;;;;;;;;

cblock      0x20
TEST
W_TEMP
STATUS_TEMP
CNT0
CNT
CNT1
endc

;;;;;;;;;;;;;vectors;;;;;;;;;;;;;

org 0
goto Mainline
org 4
goto IntService

;;;;;;;;;;;;;table;;;;;;;;;;;;;

ConvTable
movf CNT0,W
addwf PCL,F
retlw 36      ;Based on counts, returns value for
retlw 36      ;distance;
retlw 36      ;Convention
retlw 36      ;bit|7 6 5 4 3 2 1
```



```

retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16
retlw 16

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;Mainline;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Mainline
    call  Initial
    call  Gate

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;Gate;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Gate

P    btfss PORTA,1
     goto P           ;wait for ENABLE
     bcf   PORTA,0     ;turn off step request
       btfsc   PORTA,3
     call  LoL
     btfss PORTA,3
     call  ShL

     call  Delay       ;Now allow for ringing
     clrf  TMR0
     movlw B'10100000'
     movwf INTCON      ;set up global, RB0, TMR0 interrupt
Q    btfsc PORTB,0
     goto Q           ;twiddle thumbs
     call  Conv
     goto  Gate

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;Initialize;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

Initial
     movlw 0xFF
     movwf TEST

     clrf  PORTA
     movlw B'11111111'
     movwf PORTB
     movlw 0x07
     movwf CMCON
     bsf  STATUS,RP0

```

```

movlw B'10000100'
movwf OPTION_REG
movlw B'11010'      ;set up PORTA as mostly inputs
movwf TRISA
movlw B'00000001'
movwf TRISB        ;set up PORTB as mostly outputs
bcf  STATUS,RP0

btfsc PORTA,4      ;Test mode
goto  Test

return

;;;;;;;;;;;;;Interrupts;;;;;;;;;;;;;

IntService
movwf W_TEMP
swapf STATUS,W
movwf STATUS_TEMP

btfsc INTCON,2
call  Step

swapf STATUS_TEMP,W
movwf STATUS
swapf W_TEMP,F
swapf W_TEMP,W

goto  Gate

;;;;;;;;;;;;;Conversion;;;;;;;;;;;;;

Conv
clrf  INTCON
movf  TMR0,0
movwf CNT0
bcf   STATUS,0      ;clear carry bit
movf  CNT0,0        ;test to see if value is
subwf TEST,0        ;less than previous
btfss STATUS,0
call  Eror
btfss STATUS,0
return

bcf   STATUS,0      ;make sure value for counts is
movf  CNT0,0        ;0<CNT0<247 so we can read table
sublw 0xF7
btfss STATUS,0
movlw 0xF7
btfss STATUS,0
movwf CNT0
call  ConvTable     ;Convert counter value to distance
movwf PORTB        ;using table look-up.

```

```

    movf  CNT0,0
    movwf TEST
    clrf  CNT0
    movlw 120
    movwf CNT

J    clrf  TMR0      ;time ~ 200us
    bcf   INTCON,2
JJ   btfss INTCON,2
    goto JJ
    decfsz CNT
    goto J

    return

;;;;;;;;;;;;;Error;;;;;;;;;;;;;

Error
    movlw 36      ;reading not available.
    movwf PORTB
    movf  CNT0,0
    movwf TEST
    clrf  CNT0

    return

;;;;;;;;;;;;;Delay;;;;;;;;;;;;;

Delay
    movlw 0xFD
    movwf TMR0
    bcf   INTCON,2
R    btfss INTCON,2
    goto R

    return

;;;;;;;;;;;;;Step;;;;;;;;;;;;;

Step
    clrf  INTCON
    bsf   PORTA,0
    call Error
    return

;;;;;;;;;;;;;Tau;;;;;;;;;;;;;

LL   nop
AL   nop
BL   nop
CL   nop
DL   nop
EL   nop
FL   nop

    return

```

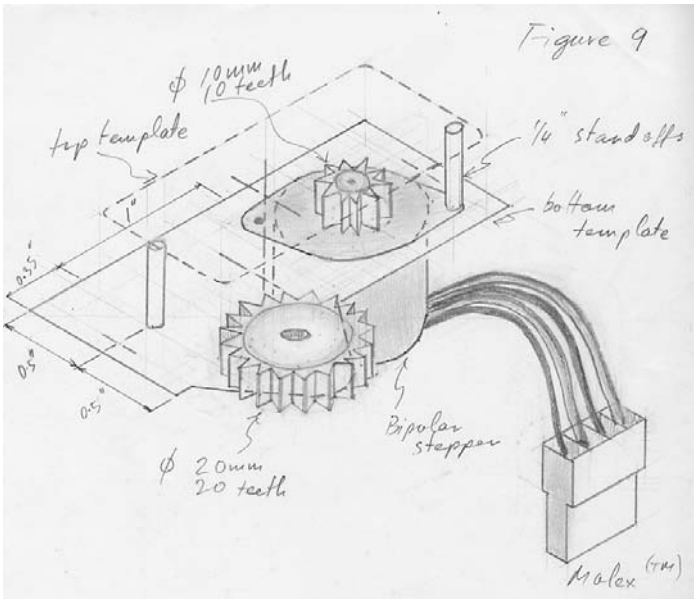



Fig1: stepper motor and gear

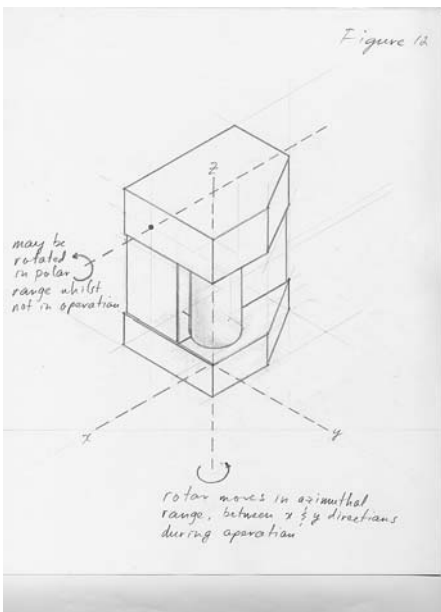


Fig2: Isometric view

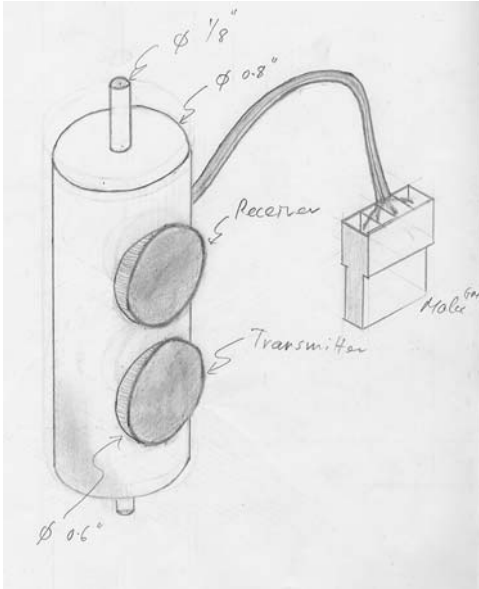


Fig3: Isometric view sensor housing

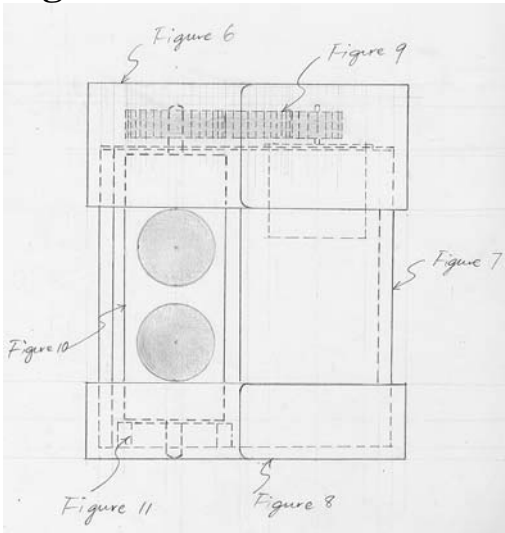


Fig4: Plan View

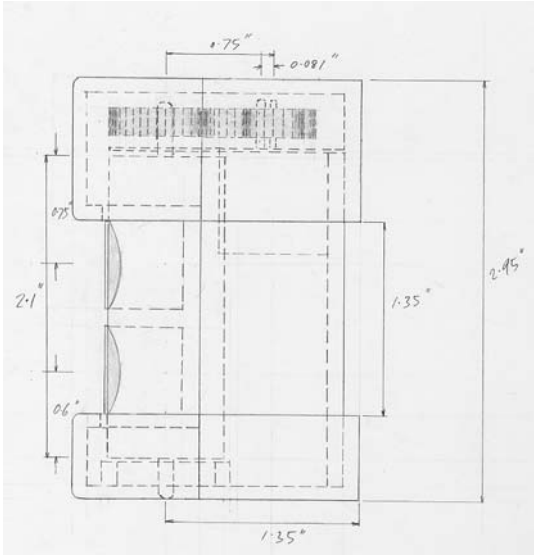


Fig5: End view

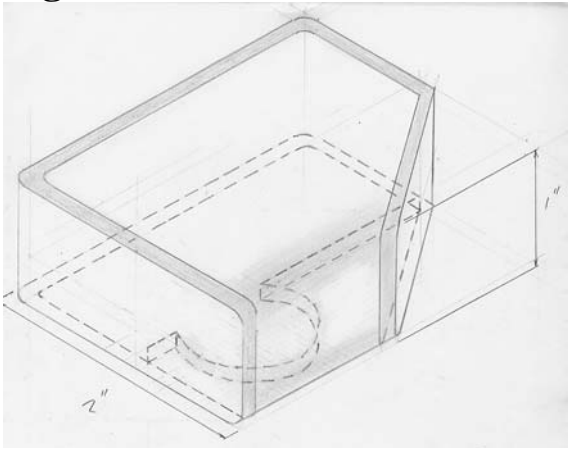


Fig6: Isometric view, top

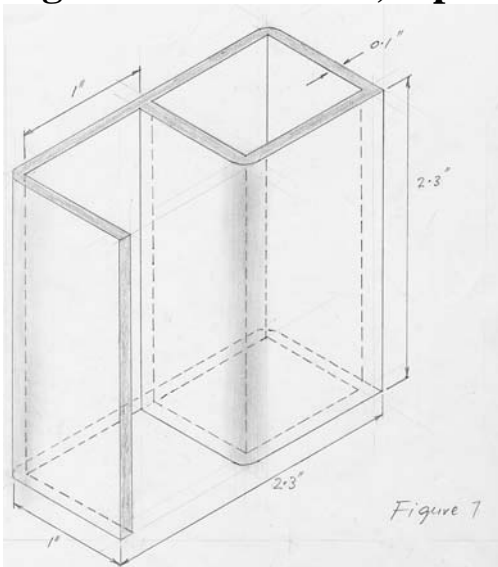


Fig7: Isometric View, body

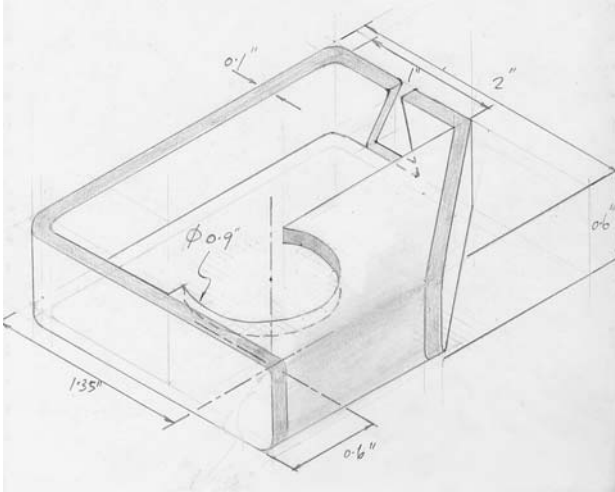


Fig8: Isometric view, bottom