

<u>AN597</u>

Implementing Ultrasonic Ranging

Author:

Robert Schreiber Logic Products Division

INTRODUCTION

Object ranging is essential in many types of systems. One of the most popular ranging techniques is ultrasonic ranging. Ultrasonic ranging is used in a wide variety of applications including:

- Auto focus cameras
- Motion detection
- Robotics guidance
- · Proximity sensing
- · Object ranging

This application note describes a method of interfacing PIC16CXX microcontrollers to the Polaroid 6500 Ranging Module. This implementation uses a minimum of microcontroller resources, a CCP module and two I/O pins. The two major components of the system are:

- Microcontroller
- Polaroid 6500 Ranging Module

The microcontroller performs the intelligence and arithmetic functions for ultrasonic ranging, while the Polaroid 6500 Ranging Module performs the ultrasonic signal transmissions and echo detection.

FIGURE 1: RANGING MODULE INTERFACE

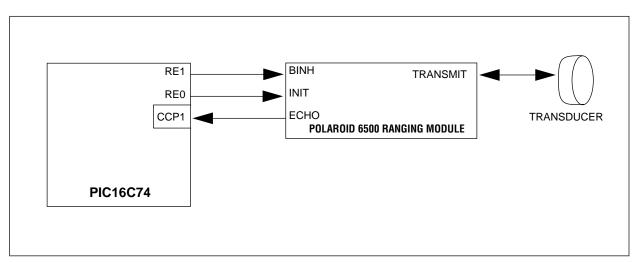
THEORY OF OPERATION

Ultrasonic ranging entails transmitting a sound wave and measuring the time that it takes for the sound wave to reflect off of an object and back to the origin. The reflection time is proportional to the distance that the object is from the source. In this implementation, the sound wave is transmitted and received from the same transducer. Therefore, a blanking interval is required between signal transmission and reception to eliminate false echoes (i.e., a transmitted signal being detected as its own echo).

CIRCUIT CONFIGURATION

In this implementation, a PIC16C74 is connected to the ranging module as shown in Figure 1. The RE0 and RE1 I/O pins are configured as digital outputs and are tied to INIT and BINH, respectively. The CCP1 pin is configured as a digital input and is tied to ECHO through a pull-up resistor. The pull-up resistor is needed since the ECHO signal is an open-collector output. The CCP1 pin is configured for capture mode (CCP1CON). Figure 2 shows the timing relationship for VDD and the three signal lines (INIT, BINH, and ECHO).

Note: The ranging module requires 5.0 milliseconds to stabilize during power-up.



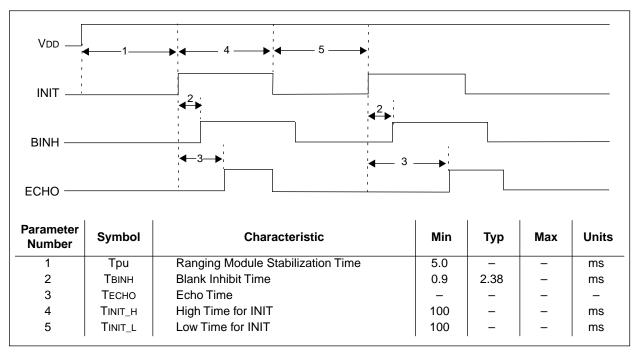


FIGURE 2: TIMING DIAGRAM OF RANGING MODULE CONTROL LINES

The PIC16C74 is configured to use one of its internal timers, Timer1, in capture mode to measure the time between signal transmission and echo detection. The resolution of the timer is determined by the microcontroller clock frequency. For this application, a 4 MHz external oscillator was used, giving a resolution of 1 ms per bit. The PIC16C74 initiates a ranging cycle by first clearing Timer1. Timer1 is then enabled and INIT is immediately asserted on the ranging module. When INIT is asserted, the ranging module transmits a series of 16 pulses on the transducer at 49.4 kHz. The transmitted pulses reflect off the object and are received back at the transducer.

The transducer is used for both transmitting and receiving sound waves. A blanking interval is needed to ensure that the transmitted signal has decayed on the transducer, in order not to receive false echoes. In normal operation, the ranging module has a blanking interval of 2.38 milliseconds, which corresponds to a minimum detection distance of approximately 17 inches. However, the BINH (blank inhibit) signal can be manipulated to reduce the blanking time on the transducer to allow for object ranging as close as 6 inches.

In this implementation, the PIC16C74 asserts the BINH signal approximately 0.9 milliseconds after signal transmission. This enables the transducer to receive reflections off objects at a distance of 6 inches. The ranging module asserts the ECHO signal when a valid reflection has been detected. The PIC16C74 uses the ECHO signal to trigger a capture of the Timer1 value. The capture register contains the 16-bit value

representing the elapsed time between signal transmission and echo detection. The PIC16C74 then calculates object distance based on the Timer1 value, microcontroller clock speed, and the velocity of sound in the atmosphere. The basic equation for calculating distance is given below:

Distance (inches) = TECHO time / 147.9 microseconds

Note:	The minimum high and low time for INIT is
	100 milliseconds, as seen in Figure 2.

DESIGN CONSIDERATIONS

There are several design considerations which must be taken into account and are listed below.

The absolute measuring distance supported by the ranging module is 6 inches to 35 feet with an accuracy of +/-1%.

The distance output from the ranging module can be averaged over time to filter distance calculations.

In some applications, the gain of the receiver amplifier may be too low or too high and may need to be adjusted. For example, if the transducer is mounted in a cylinder, the gain may need to be lowered to reduce false echoes within the cylinder. In this case, R1 (refer to the Polariod Ultrasonic Ranging System manual) may be replaced with a 20 k Ω potentiometer to tweak the gain of the receiver amplifier to reduce false echoes.

In order for the Polaroid 6500 ranging module to operate properly, the power supply must be capable of handling high current transients (2.5 A) during the

transmit pulse. The instantaneous drain on the power supply can be mitigated by installing a storage capacitor across the power lines at the ranging module. A value of 500 microfarads is recommended.

A 200 millisecond interval is recommended between ranging cycles (Figure 2) to allow the transducer to clear.

The ECHO line requires a pull-up resistor (4.7 k Ω was used in this application).

There must be a common ground between the PIC16C74 circuitry and the ranging module.

Some applications may not need the resources of the higher end PIC16CXX devices. It is still possible to do this application using a device that does not contain a CCP module (for ECHO timing). The capture function can be implemented in firmware. The effect of a firmware implementation is that the resolution of the ECHO time would be 3 Tcy cycles versus 1 Tcy cycle for the CCP module. Also, the firmware implementation would not allow other tasks to be performed while the capture function was occurring.

Refer to Appendix A for general ranging module specifications.

APPENDIX A: POLAROID MODULE SPECIFICATIONS

Note: This appendix contains general specifications from the Polaroid Ultrasonic Ranging System Manual. Please refer to the current Polaroid Ultrasonic Ranging System Manual for current information regarding ranging module design considerations.

DESIGN CONSIDERATIONS IN ULTRASONICS

Range: (with user custom designed processing electronics)

Farther

- a) Use an acoustic horn to "focus" the sound (narrowing the beamwidth).
- b) Use two transducers 1 receiver and 1 transmitter facing each other.
- c) Lower the transmitting frequency (which will decrease the attenuation in air).

<u>Closer</u>

- a) Use a shorter transmit signal (such as four cycles).
- a) Use two transducers one to transmit, one to receive (eliminates waiting for damping time).

<u>Resolution</u>

- a) Above all, know the target and range well, and design a system with them in mind.
- b) Use a higher transmit frequency.
- c) Look at phase differences of a given cycle of the transmitted signal and received echo (as opposed to using and integration technique).
- d) Increase the clock frequency of the timer.
- Accuracy: (again, you must have a well defined target)

Temperature Compensate

- a) Use a second small target, as a reference, at a known distance in the ranging path (such as a 1/4" rod several feet away), process both echoes, then normalize the second distance with respect to the first, since t1/d1 = t2/d2.
- b) Incorporate a temperature sensing integrated circuit to drive a VCO to do the distance interval clocking.
- c) To increase sensitivity of detection circuit change the value of C4 from 3300 pF to 1000 pF on the 6500 Series Ranging Module.

Beam Width:

<u>Increase</u>

- a) Use an acoustic lens (to disperse the signal).
- b) Decrease the transmitting frequency.
- c) Use several transducers to span an area.

Decrease

- a) Use an acoustic horn (to focus the sound).
- b) Increase the transmitting frequency.

TABLE 1: RECOMMENDED OPERATING CONDITIONS

		Min.	Max.	Unit
Supply Voltage, Vcc		4.5	6.8	V
High-level input voltage, VIH	BINH, INIT	2.1		V
Low-level input voltage, VIL	BINH, INIT		0.6	V
ECHO and OSC output voltage		6.8	V	
Delay time, power up to INIT high	5		ms	
Recycle period	80		ms	
Operating free-air temperature, TA	0	40	°C	

TABLE 2:ELECTRICAL CHARACTERISTICS OVER RECOMMENDED RANGES OF SUPPLY
VOLTAGE AND OPERATING FREE-AIR TEMPERATURE (UNLESS OTHERWISE
NOTED)

Parameter	Test Conditions	Min.	Тур.	Max.	Unit		
Input current	BINH, INIT	V1 = 2.1V			1	mA	
High-level output current, IOH	ECHO, OSC	Vон = 5.5V			100	μΑ	
Low-level output voltage, VoL	ECHO, OSC	IOL = 1.6 mA			0.4	V	
Transducer bias voltage	TA = 25°C		200		V		
Transducer output voltage (peak-to-	TA = 25°C		400		V		
Number of cycles for XDCR output	C= 500 pF			7			
Internal blanking interval			2.38*		ms		
Frequency during 16-pulse trans-	OSC output			49.4*		– kHz	
mit period	XMIT output			49.4*			
Frequency after 16 pulse transmit	OSC output			93.3*			
period	XMIT output			0			
Querale summer las	During transmit period				2000	mA	
Supply current, Icc	After transmit period				100		

* These typical values apply for a 420 kHz ceramic resonator.

Please check the Microchip BBS for the latest version of the source code. For BBS access information, see Section 6, Microchip Bulletin Board Service information, page 6-3.

APPENDIX B: FIRMWARE LISTING

```
MPASM 01.02 Released
                        XDCR.ASM 11-14-1994 9:29:15
                                                                      PAGE 1
LOC OBJECT CODE LINE SOURCE TEXT
 VALUE
          0001 ; XDCR.ASM
          0002 ;
          0003 ; This routine continually executes ranging cycles in the
          0004 ; following order:
          0005 ;
          0006 ;
                     1) Timers and Flags are cleared
          0007 ;
                     2) Ranging Cycle Executes
          ; 8000
                     3) Distance is Calculated (to 0.5 inch)
          0009;
                    4) HW is re-initialized for next cycle
          0010 ;
          0011 ; The processor uses a 4MHz oscillator, so all timing
          0012 ; calculations are referenced to that. The calculated
          0013 ; distance is a 16-bit result in the ACCbHI:ACCbLO registers.
          0014 ;
          0015
          0016
                      LIST P=16C74, F=INHX8M
          0017 ;
          0029
          0030 ;***************
          0031 ; Bank 0 Registers
          0032 ;**************
          0033 ;
          0034 ; TMR1 is off, Prescaler is 1 for a capture timeout of 65 msec
0000 0190 0035 clrf
                       T1CON
          0036 ; Set to capture on every rising edge
0001 3005 0037 movlw 0x05
0002 0097 0038 movwf CCP1CON
          0039 ; Clear the Ports
0003 0185 0040 clrf PORT_A
0004 0186 0041
                clrf
                         PORT B
                clrf
0005 0187
          0042
                         PORT C
0006 0188
          0043
                 clrf
                         PORT D
0007 0189
          0044
                 clrf
                         PORT_E
          0045 ;
          0046 ;**************
          0047 ; Bank 1 Registers
          0048 ;**************
          0049 ;
0008 1683 0050 bsf
                       STATUS, RPO
                                      ; Set RPO
          0051 ; Port A is Digital, Port E is Digital
0009 3007 0052 movlw 0x07
                 movwf ADCON1
000A 009F 0053
          0054 ; Configure CCP1 (RC2) as an input, and all other ports
          0055 ; as Outputs, (RE0 = INIT, RE1 = BINH)
000B 0185 0056 clrf TRIS_A
000C 0186 0057
                 clrf
                         TRIS B
000D 3004
          0058
                 movlw
                         0x04
000E 0087
          0059
                 movwf
                         TRIS C
000F 0188
          0060
                 clrf
                         TRIS_D
0010 0189
          0061
                 clrf
                         TRIS E
0011 1283 0062
                 bcf
                        STATUS, RPO
                                       ; Clear RPO
0012
          0063 Xdcr
          0064 ;
          0065 ; Initialize Timers and Flags
          0066 ;
0012 1010
          0067
                 bcf
                         T1CON,0
                                         ; Disable TMR1
0013 018C 0068
                 clrf
                         PIR1
                                         ; Clear Timer1 Overflow Flag & Timer1 Capture Flag
```

0014 018E 0069 clrf TMRL ; Clear TMRLH 0015 018F 0070 clrf TMRLH ; Clear CCPRIH 0016 0195 0071 clrf CCPRIH ; Clear CCPRIH 0018 1409 0073 baf PORT_K,0 ; Set INIT Migh on Ranging Module 0019 1410 0074 baf PORT_K,0 ; Set INIT Migh on Ranging Module 0019 1410 0074 baf PORT_K,0 ; Set INIT Migh on Ranging Module 0019 1410 0074 baf PORT_K,1 ; Enable TMRL 0010 1907 chk.tl 0011 1409 0078 bifso FIRI.2 ; Check for Capture 0010 1907 0018 bifso FIRI.2 ; Check for CAPLURE 0012 1907 0018 bifso FIRI.2 ; Check for TMRL Overflow 0011 1000 0020 bof TlcON,0 ; Turn off TMRL 0022 0079 goto chk.tl 0021 0010 0020 bof TlcON,0 ; Turn off TMRL 0022 0014 chk.done 0086 ; Calculate distance to 0.5 inch resolution 0087 ; 0086 ; Calculate distance to 0.5 inch resolution 0087 ; 0086 ; Calculate distance to 0.5 inch resolution 0087 in 0087 bof TlCON,0 ; Turn off TMRL 0021 0010 0088 bof TlCON,0 ; Turn off TMRL 0022 0010 0088 bof TlCON,0 ; Turn off TMRL 0023 0010 0088 bof TlCON,0 ; Turn off TMRL 0024 0002 0090 movef ACChLO ; MOVer LSB into MC 0025 0018 0091 movef ACChLI ; MOVE MSB into ACChLI 0027 0040 0093 moviw 0x4A ; MOVE MSB into ACCALI 0028 00A0 0993 moviw 0x4A ; MOVE MSB into ACCALI 0029 01A1 0995 clrf ACCALI ; MOVE MSB into ACCALI 0027 0098 0094 movef ACCALO ; MOVE LSB into ACCALI 0029 01A1 0995 clrf ACCALI ; MOVE MSB into ACCALI 0020 0031 0049 movef ACCALO ; MOVE LSB into ACCALI 0032 0031 0041 movef ACCALI ; MOVE MSB into ACCALI 0032 0031 0031 0044 movef ACCALI ; Clear MSB (ACCALI) 0032 0031 0031 013 bifso STATUS,CALV ; If Memainder < (0.5 * Divisor), skip 0032 0131 0095 movef ACCALI ; MOVE MSB into ACCALI 0032 0131 0035 0140 off ACCALI ; Clear MSB (ACCALI) 0030 0130 0131 bifso STATUS,CALV ; If Memainder < (0.5 * Divisor), skip 0031 0130 0141 bifso STATUS,CALV ; If Move LSB INT 0032 0131 0131 bifso STATUS,CALV ; If Move LSB INT 0035 0100 0112 bifs STATUS,CALV ; Mait 100 mace before clearing HW. 0035 0110 ding 0114 ACCHI 0035 0110 0114 csl1 DML_100 ; Wait 100 mace before enabling HW. 0035 0110 011					
0016 0195 0071 clrf CCPRLI ; Clear CCPRLH 0017 0196 0072 off CCPRL ;; Clear CCPRLH 0018 1409 0073 bef PORP.E.O. ; Set INIT High on Ranging Module 0018 21P3 0075 call DBL_9 ; Delay 0.9 macc for transducer to stabilize 0018 1489 0076 call DBL_9 ; Delay 0.9 macc for transducer to keceive (BINN) 00112 022 0079 got chk_chone ; Jump if Capture 0012 022 0079 oot chk_chone ; Jump if Capture 0012 023 0081 got ovr_flo ; Capture event did not occur 0022 0030 ovr_flo ; Turn off TRRI 0022 0030 movef ACCDEL, ; Move LSB into M 0023 0815 089 movef ACCDEL, ; Move KSB into W 0024 0032 0816 0091 movef ACCDEL, ; Move KSB into ACCDHI 0024 0032 0816 0091 movef ACCDEL, ; Move KSB into ACCDHI 0027 0034 movef ACCDHI ; Move KSB into ACCDHI ; Move KSB into ACCDHI 0028 00A0 0094 movef ACCALO ; Move KSB into ACCDHI ; Move KSB into ACCDHI 0029 01A1 0095 off ACCAHI ; Clear KMSA (ACCAHI) 0028 02A0 0094 movef ACCALO ; Wove KSB into ACCDHI ; Move KSB into ACCDHI 0028 02A1 0095	0014 018E 006	9 clrf	TMR1L	;	Clear TMR1L
0017 0196 0073 baf PORT_F,0 ; Set INT High on Ranging Module 0018 1400 0074 baf PORT_F,0 ; Set INT High on Ranging Module 0018 1409 0075 call DEL_9 ; Pelay 0.9 msec for transducer to stabilize 0018 1409 0077 chk_din ; Check for Capture 0011 0100 0077 chk_din ; Jump if Capture 0012 1000 0080 bifse PIRI,0 ; Check for TMMI Overflow 0018 1000 0080 pito or chk_t1 ; Loop if nothing happened 0021 010 0082 pito or chk_t1 ; Loop if nothing happened 0022 0101 0082 calculate distance to 0.5 inch resolution 0085 ; Calculate distance to 0.5 inch resolution 0086 ; Calculate distance to 0.5 inch resolution 0022 0101 0088 bcf TiCON,0 ; Turn off TMRI 0023 0101 0088 bcf TiCON,0 ; Turn off TMRI 0024 00A2 0090 mover ACChLO ; Move LSB into ACChLO 0024 00A2 0090 mover ACCHLO ; Move CBB into ACChLO 0025 0101 mover ACCALO ; Move CBB <td< td=""><td>0015 018F 007</td><td>0 clrf</td><td>TMR1H</td><td>;</td><td>Clear TMR1H</td></td<>	0015 018F 007	0 clrf	TMR1H	;	Clear TMR1H
0019 1409 0074 bsf PORT_F.0 ; Set INIT High on Ranging Module 0019 1410 0074 bsf PORT_F.1 ; Enable Transducer to Receive (BINH) 0012 0077 cht_1 ; Enable Transducer to Receive (BINH) 0012 0077 cht_1 ; Enable Transducer to Receive (BINH) 0012 0077 cht_1 ; Loop if nothins happened 0012 0081 oto cht_1 ; Loop if nothins happened 0021 0083 oto cott_1 ; Capture event did not occur 0022 0084 cht_done ; Jump if Capture 0021 0233 083 goto ort.1 ; Capture event did not occur 0022 0084 cht_done ; Move LSB into ACCLO 0023 0815 ooff attance to 0.5 into free 0023 0815 ooff attance ; Move LSB into ACCLO 0023 0815 ooff attance ; Move MSB into ACCLO 0024 0040 movef ACCLO ; Move LSB into ACCLO 0028 0040	0016 0195 007	1 clrf	CCPR1L	;	Clear CCPR1L
0019 1410 0074 ksf TICON,0 ; Enable TWRL 0018 218 0075 call DRL) Delay 0.9 smcd for transducer to Receive (BINE) 0011 1007 cht_t1 ; Check for Capture (BINE) 0011 1007 cht_c1 ; Check for Capture (BINE) 0011 1007 cost FIL1.2 ; Check for TMRI Overflow 0011 1007 0080 bft.se FIL1.0 ; Check for TMRI Overflow 0012 1010 0082 bft. TLON,0 ; Turn off TWR1 0021 01010 0083 cft.ad.cone (0085 ; 0023 0110 0088 bft. TLON,0 ; Turn off TWR1 0024 0004 0084 cht_chane (0087 ; 0023 0815 0080 movf CCAPLL,N ; Move LSB into % 0024 0004 0093 movf CCAPLL,N ; Move CAPLL 0024 004004	0017 0196 007	2 clrf	CCPR1H	;	Clear CCPR1H
001A 21F3 0075 call DEL_9' ; Delay 0.9 msec for transducer to stabilize 001B 1489 0076 bdf PORT_E.1 ; Enable Transducer to Receive (BINH) 001C 0007 chk_t1 ; Check for TMSI Overflow 001D 2822 0009 goto chk_tan ; Jump if Capture 001E 2812 0081 goto chk_tan ; Jump if Capture 001E 2812 0081 goto chk_tan ; Jump if Capture 0021 2833 0083 goto chk_tan ; Loop if nothing happened 0022 0084 chk_dome ; Turn off TMRI 0023 0085 ; Calculate distance to 0.5 inch resolution 0085 ; 0024 0086 bcf TICON,0 ; Turn off TMRI 0023 0086 bcf TICON,0 ; Move LSB into W 0024 0080 0099 movf CCPRIH,W 0024 0080 0090 movf ACCBAI 0025 016 ACCAH ; Move LSB into ACCAH 0026 0141 0095 clrf ACCAH	0018 1409 007	3 bsf	port_e,0	;	Set INIT High on Ranging Module
0011 0077 bbf PORT_E.1 ; Enable Transducer to Receive (BINH) 0011 0077 btfsc PIE1.2 ; Check for Capture 0011 1002 0078 btfsc PIE1.2 ; Check for Capture 0011 1002 0080 btfss PIE1.0 ; Check for TMRI Overflow 0012 1010 0082 btfss PIE1.0 ; Check for TMRI Overflow 0022 1008 std.dome ; Turn off TMRI 0022 0084 chk.dome ; Outro off TMRI 0023 0081 colo 0.082 bef TICON.0 ; Turn off TMRI 0024 0010 0088 bef TICON.0 ; Turn off TMRI 0022 0081 movef ACCPRIL, W ; Move LSB into W 0023 0815 0099 movef ACCPRIL, W ; Move LSB into W 0024 0002 0034 movef ACCPRIL, W ; Move LSB into W 0025 0810 091 movef ACCPRIL, W ; Move LSB into W 0024 0040 092	0019 1410 007	4 bsf	T1CON,0	;	Enable TMR1
001C 0077 chk_t1 001C 0079 goto chk_done ; Jump if Capture 001D 2822 0079 goto chk_t1 ; Loop if Capture 001F 281C 0081 goto chk_t1 ; Loop if nothing happened 0021 0083 goto ork_t1 ; Loop if nothing happened 0022 0084 chk_done ; Capture event did not occur 0022 0084 chk_done ; Turn off TMR1 0021 0083 pot rtron off TMR1 0022 0084 chk_done ; Move LSB into W 0023 0010 movf CCPRIL,W ; Move LSB into ACCLO 0024 0042 0043 movf CCPRIL,W ; Move LSB into ACCLO 0025 0015 movf CCELO ; Move ACCLO ; Move CACLO 0026 0043 movf ACCALO ; Wore 75usec/0.50in into W 0027 clear MSB inco ACCALO ; Move ACCALO 0028 0040 <td< td=""><td>001A 21F3 007</td><td>5 call</td><td>DEL_9</td><td>;</td><td>Delay 0.9 msec for transducer to stabilize</td></td<>	001A 21F3 007	5 call	DEL_9	;	Delay 0.9 msec for transducer to stabilize
<pre>0010 190C 0078 btfsc PIR1,2 ; Check for Capture 001D 2822 0079 goto chk_done ; Jump if Capture 001E 100C 0080 btfss PIR1,0 ; Check for TMEL Overflow 001E 120C 0081 goto chk_t1 ; Loop if nothing happened 0021 0101 0082 bef TICON,0 ; Turn off TMR1 0022 0084 cfk_done 0085; 0086 ; Calculate distance to 0.5 inch resolution 0087; 0022 01010 0088 bef TICON,0 ; Turn off TMR1 0023 0815 0089 movrf ACCELO ; Nove LSB into W 0024 0000 0094 movrf ACCELO ; Nove LSB into ACCELO 0025 0816 0091 movrf ACCELO ; Nove LSB into ACCELO 0025 0816 0091 movrf ACCELI ; Nove MSB into ACCELO 0026 0033 0092 movrf ACCELI ; Nove MSB into ACCELO 0026 0030 0094 movrf ACCELI ; Nove MSB into ACCELI 0027 0040 0094 movrf ACCELO ; Nove LSB into ACCELI 0028 0040 0094 movrf ACCELO ; Nove LSB into ACCELI 0029 01A1 0095 clrf ACCELI ; Nove MSB into ACCELI 0097 ; Call 16-bit/8-bit routine 0097 ; Call 16-bit/8-bit routine 0097 ; Which is described in 0098 ; Application Note 544 002D 1803 0101 btfsc STATUS,CARRY ; If Remainder to see if 0027 1003 0103 btfsc STATUS,CARRY ; If Remainder < (0.5 * Divisor), skip 002E 1003 0101 btfsc STATUS,Z ; Check remainder to wrap around 0031 1003 0101 btfsc STATUS,Z ; Check low byte for wrap around 0031 0103 0101 btfsc STATUS,Z ; Check low byte for wrap around 0031 0103 0105 btfss STATUS,Z ; Check low byte for wrap around 0031 0103 0105 btfss STATUS,Z ; Check low byte for wrap around 0032 0124 incf ACCELI,F ; If LSB wrapped, increment high byte 0035 0100 done ; High byte didn't wrap 0036 0109 olig ACCELI ACCELI 0036 0109 0112 btf PORT_F,0 ; Disable INIT 0037 1089 0113 bcf PORT_F,1 ; Disable INIT 0037 1089 0113 bcf PORT_F,1 ; Disable INIT 0037 1089 0113 bcf PORT_F,1 ; Disable INIT 0038 21PD 0114 ccall DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdor 0116 0116 0116 0116 0116 0116 0116 011</pre>	001B 1489 007	6 bsf	PORT_E,1	;	Enable Transducer to Receive (BINH)
001D 1222 0079 goto chk_done ; Jupp if Capture 001F 201C 0080 bifs PIRL0 ; Check for TMRL Overflow 0012 2030 0081 goto chk_t1 ; Loop if nothing happened 0021 2030 0083 bof TLCON.0 ; Turn off TMRL 0022 0084 chk_done ; Capture event did not occur 0023 0085 ; Calculate distance to 0.5 inch resolution 0087 : 0088 bof TLCON.0 ; Turn off TMRL 0022 0084 chk_done ; Move LSB into N 0022 0085 outs for CCPRLN.W ; Move LSB into ACCLO 0024 0020 0090 movef CCPRLN.W ; Move MSB into ACCLO 0026 00A3 0092 movef ACCLO ; Move LSB into ACCLO 0027 00A1 0093 movef ACCLO ; Move TSB into ACCLO 0028 00A0 0094 movef ACCLO ; Move TSB into ACCLO 0028 00A0 0094 movef ACCLO ; Watch is described in 0028 0026 0039 clf ACCALO ; Which is described in 0028 0020 0099 movef ACCLO,W ; Watch is described in 0028 0101	001C 007	7 chk_t1			
001E ICCC 0080 btfss FIR1.0 ; Check for TMR1 Overflow 001F 281C 0081 goto chk_l1 ; Loop if nothing happened 0021 101 0082 bcf TICON.0 ; Turn off TMR1 0022 0084 chk_done capture event did not occur 0085 ; colstance colstance colstance 0085 ; Calculate distance to 0.5 inch resolution colstance 0087 ; Curn off TMR1 colstance 0023 0815 0089 movf CCPRIL,W ; Move LSB into ACCDLO 0024 0004 0034 movef CCPRIL,W ; Move MSB into W 0025 0816 0091 movef ACCDHI ; Move Sistico ACCDHI 0026 0031 0092 movef ACCALH ; Move Sistico ACCAHI 0027 304A 0093 movef ACCALO ; Move LSB into ACCAHI 0028 0101 0095 clrf ACCALH ; Clear MSB (ACCAHI) 0028 0028 0280 0096 call D_divF ; Clait 16-bit/8-bit routine ; which is described in 0027 1042 0098 movef ACCALO,W ; we hould round up 00202 02022 0100 movef AC	001C 190C 007	8 btfsc	PIR1,2	;	Check for Capture
001F 281C 0081 goto chk_t1 ; Loop if nothing happened 0020 1010 0082 bcf TICON,0 ; Turn off TMR1 0021 2833 0083 goto ovr_flo ; Capture event did not occur 0022 0084 chk_done ; 0085; ; 0085 ; 0086 ; Calculate distance to 0.5 inch resolution ; 0022 1010 0088 bcf TICON,0 ; Turn off TMR1 0023 0815 0809 movf CCPRLL,W ; Move LSB into ACCDLO 0024 0032 0033 movf ACCAL ; Move TSB into ACCDH 0025 0043 0094 movf ACCAL ; Move LSB into ACCALO 0028 0040 0094 movf ACCALO ; Move LSB into ACCALO 0028 0040 0094 movf ACCALO ; Move LSB into ACCALO 0028 0045 0097 ; which is described in ; hope in othe Scalo 0029 0110 </td <td>001D 2822 007</td> <td>9 goto</td> <td>chk_done</td> <td>;</td> <td>Jump if Capture</td>	001D 2822 007	9 goto	chk_done	;	Jump if Capture
0020 1010 0082 bcf TLCN,0 ; Turn off TMRI 0021 2833 0083 goto ovr_flo ; Capture event did not occur 0022 0084 chk_done 0085; 0086 ; Calculate distance to 0.5 inch resolution 0086; 0023 0815 0088 bcf TLCON,0 ; Turn off TMRI 0023 0815 0089 movf CCPRIL,W ; Move LSB into ACCDLO 0024 0002 0090 mov# ACCLO ; Move MSB into W 0026 0035 0025 0816 0091 mov# ACCLI ; Move MSB into ACCDH 0027 304A 0028 0000 0094 mov# ACCLIC ; Move LSB into ACCALO 0028 0040 0028 0010 0094 mov# ACCLIC ; Wove LSB into ACCALO 0028 0020 0028 0026 0095 clrf ACCLIC ; Which is described in 0028 0026 0099 movW 0245 ; Check remainder to see if 0020 1803 0101 btfsc STATUS,Z ; Check low bte for wrap around 0021 103 0105 btfsc STATUS,Z ; C	001E 1COC 008	0 btfss	PIR1,0	;	Check for TMR1 Overflow
0021 2833 0083 goto ovr_flo ; Capture event did not occur 0022 0084 chk_done 0085; 0086; Calculate distance to 0.5 inch resolution 0087; 0022 1010 0088 bcf TICON,0 ; Turn off TMR1 0023 0815 0089 movf CCPR1L,W ; Move LSB into ACCDLO 0024 0022 0090 movf ACCDLO ; Move LSB into ACCDLO 0025 0816 0091 movf CCPR1H,W ; Move MSB into ACCDHI 0027 0304 0093 movlw 0x4A ; Move SB into ACCDHI 0027 0040 0094 movf ACCLO ; Move LSB into ACCDHI 0027 0040 0094 movf ACCLO ; Move LSB into ACCDHI 0028 0080 0094 movf ACCLO ; Move LSB into ACCDHI 0029 01A1 0095 clrf ACCAHI ; Clear MSB AcCCAIO 0029 01A1 0095 clrf ACCAHI ; Clear MSB AcCCAIO 0029 01A1 0095 clrf ACCAHI ; Clear MSB (ACCAHI) 0027 0098 ; Actor LSB AcCCAIO 0028 0080 0099 movlw 0x25 ; Check remainder to see if 0020 2024 0100 subwf ACCLO,F ; Round up 0027 1903 0101 btfsc STATUS,CARRY ; If Remainder < (0.5 * Divisor), skip 0027 0030 0A3 0104 incf ACCDLO,F ; Round up 0027 1903 0103 btfsc STATUS,Z ; Check low byte for wrap around 0030 0A3 0104 incf ACCDLO,F ; If LSB wrapped, increment high byte 0031 0A3 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 0106 goto done ; High byte didn't wrap 0033 0107 ovr_flo 0033 0108 clrf ACCDLO 0033 0110 done 0035 0110 done 0035 0110 done 0035 0110 done 0036 1099 0112 bcf PORT_E,0 ; Disable BINH 0036 113 bcf PORT_E,1 ; Disable BINH 0038 212P 0114 call DEL_100 ; Wait 100 msec before clearing HW. 0039 2812 0115 goto Xdcr 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	001F 281C 008	1 goto	chk_t1	;	Loop if nothing happened
0022 0084 chk_done 0085; Calculate distance to 0.5 inch resolution 0086; Calculate distance to 0.5 inch resolution 0022 0088 bcf TICON,0 ; Turn off TMR1 0023 0815 0089 movf CCPRLL,W ; Move LSB into W 0024 0080 movef ACCDLO ; Move MSB into ACCDA 0026 0816 0091 movef ACCDHI ; Move MSB into ACCDHI 0027 304A 0093 movef ACCAHI ; Move MSB into ACCDHI 0028 0040 0094 movef ACCAHI ; Clear MSB (ACCAHI) 0028 0040 0094 movef ACCAHI ; Clear MSB (ACCAHI) 0028 0040 0094 movef ACCALO ; Which is described in 0028 0095 clrf ACCALO,W ; We should round up 0020 1003 btfsc STATUS, CARRY ; If Remainder < (0.5* bivisor), skip	0020 1010 008	2 bcf	T1CON,0	;	Turn off TMR1
0085 ; 0086 ; Calculate distance to 0.5 inch resolution 0087 ; 0022 1010 0088 bcf TLCON,0 ; Turn off TMR1 0023 0815 0089 movf CCPRLL,W ; Move LSB into W 0024 00A2 0090 movwf ACCbLO ; Move LSB into ACCbLO 0025 0816 0091 movf CCPRLH,W ; Move MSB into ACCHI 0027 304A 0093 movwf ACCbHI ; Move MSB into ACCCHI 0027 304A 0093 movwf ACCaLO ; Move LSB into ACCALO 0028 00A0 0094 movwf ACCaLO ; Move LSB into ACCALO 0029 01A1 0095 clrf ACCaHI ; Clear MSB (ACCAHI) 0028 00A0 0099 movW MXA ; Clear MSB (ACCAHI) 0028 00A0 0099 movW MX25 ; Clear MSB (ACCAHI) 0028 0099 movW MX25 ; Check remainder to see if 0097 ; Which is described in 0098 ; Application Note 544 0020 2022 0099 movW 0X25 ; Check low byte for wrap around 0020 1010 btfsc STATUS,CARRY ; If Remainder < (0.5 * Divisor), skip 0020 103 0101 btfsc STATUS,2 ; Check low byte for wrap around 0030 0AA3 0104 incf ACCbHJ,F ; If LSB wrapped, increment high byte 0033 0107 ovr_flo 0033 0107 ovr_flo 0033 0107 ovr_flo 0033 0107 ovr_flo 0035 21FD 0111 call DEL_100 ; Wait 100 msec before clearing HW. 0036 1009 0112 bcf PORT_E,0 ; Disable INIH 0038 21FD 0113 bcf PORT_E,0 ; Disable INIH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before clearing HW. 0039 2812 0115 goto Xdcr 0116 0130 015 btfsv XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	0021 2833 008	3 goto	ovr_flo	;	Capture event did not occur
0086 ; Calculate distance to 0.5 inch resolution 0021 1010 0088 0022 1010 0088 0023 0015 0089 0024 0024 0090 0025 0816 0091 0027 3040 0092 0027 3040 0093 0027 3040 0093 0028 00A0 0094 0027 3040 0093 0028 00A0 0094 0028 011 0055 0028 011 0056 0029 0111 0056 0029 0111 0056 0020 0096 0021 0096 0021 0096 0022 0096 0021 0031 0022 0099 0024 0009 0025 0099 0026 0101 0027 0103 0028 0101 0029 0103 0102 166 0103 1016 0102 167 <td>0022 008</td> <td>4 chk_doi</td> <td>ne</td> <td></td> <td></td>	0022 008	4 chk_doi	ne		
0027 ; 0021 000 0088 bcf TlCON,0 ; Turn off TMR1 0023 0815 0088 movf CCPRIL,W ; Move LSB into W 0024 00A2 0090 movwf ACCbL0 ; Move KSB into ACCbL0 0025 0816 0091 movf CCPRIH,W ; Move MSB into ACCbHI 0026 00A3 0092 movwf ACCbHI ; Move MSB into ACCbHI 0027 304A 0093 moviw 0x4A ; Move TSB into ACCaL0 0028 00A0 0094 movwf ACCLO ; Move LSB into ACCaL0 0028 00A0 0094 movwf ACCLO ; Move LSB into ACCaL0 0028 00A0 0095 clrf ACCAHI ; Clear MSB (ACCAHI) 0029 01A1 0095 clrf ACCAHI ; Clear MSB (ACCAH) 0028 0040 0096 call D_divF ; Call 16-bit/8-bit routine 0097 ; which is described in 0098 ; Application Note 544 00202 0224 0100 subwf ACCCLO,W ; we should round up 0021 803 0101 btfss STATUS,Z ; Check low byte for wrap around 0030 0AA3 0104 incf ACCLOF; F ; Round up 0031 1003 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 2835 0106 goto done ; High byte didn't wrap 0033 01A2 0108 clrf ACCBHI 0034 01A3 0109 clrf ACCBHI 0035 1010 done 0036 1019 btfs goto Xdcr 0037 1089 0113 bcf PORT_E,0 ; Disable INIT 0038 21FD 0114 call DEL_100 ; Wait 100 msec before clearing HW. 0039 2812 0115 goto Xdcr <	008	5;			
0022 1010 0088 bcf TICON,0 ; Turn off TMR1 0023 00815 0089 movf CCPRIL,W ; Move LSB into ACCDLO 0025 0016 0091 movf CCPRIL,W ; Move KSB into ACCDLO 0026 00A3 0092 movmf ACCDHI ; Move MSB into ACCDHI 0027 004A 0093 movlw 0x4A ; Move 75uec/0.50in into W 0028 00A0 0094 movmf ACCALO ; Move LSB into ACCALO 0029 01A1 0095 clrf ACCAHI ; Clear MSB (ACCAHI) 0028 0086 0096 call D_divF ; Call 16-bit/8-bit routine 0097 ; Which is described in 0098 (ACCALO); Move TSUEC/0.50in Note 544 0028 0020 0109 movlw 0x25 ; Check remainder to see if 0020 1010 subwf ACCCLO,W ; We should round up 0021 0103 0101 btfsc STATUS,CRRY ; If Remainder < (0.5 * Divisor), skip 0022 0AA2 0102 incf ACCLO,F ; Round up 0021 903 0103 btfsc STATUS,Z ; Check low byte for wrap around 0030 0AA3 0104 incf ACCHI,F ; If LSB wrapped, increment high byte 0031 0103 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 0A33 0107 ovr_flo 0033 0107 ovr_flo 0034 0113 0109 0112 bcf PORT_E,1 ; Disable INIT 0035 0110 done 0035 0110 done 0150 end 0150 016 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	008	6 ; Calcula	te distance to 0.5	in	ch resolution
0023 0815 0089 movf CCPRLL,W ; Move LSB into W 0024 00A2 0090 movf ACCbLO ; Move LSB into ACCbLO 0025 0816 0091 movf ACCbHI ; Move MSB into ACCbHI 0027 304A 0093 movlw 0x4A ; Move MSB into ACCaLO 0028 00A0 0094 movmf ACCaLO ; Move LSB into ACCaLO 0029 01A1 0095 clrf ACCaHI ; Clear MSB (ACCaHI) 002A 208F 0096 call D_divF ; Call 16-bit/8-bit routine 0097 ; which is described in 0098 ; Application Note 544 002D 1003 0101 btfsc STATUS,Z ; Check remainder to see if 002E 1002 1010 subwf ACCALO,W ; We should round up 002D 1030 0101 btfsc STATUS,Z ; Check low byte for wrap around 0030 0AA3 0104 inef ACCALI,F ; If LSB wrapped, increment high byte 0031 1003 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 2035 0110 done ; High byte didn't wrap 0033 0107 ovr_flo 0034 01A3 0109 clrf ACCALI 0035 0110 done ; High byte didn't wrap 0035 0110 done 0035 0110 done 0035 0110 done 0035 21FD 0111 call DEL_100 ; Wait 100 msec before clearing HW. 0036 1009 0112 bcf PORT_E,1 ; Disable INIT 0037 1089 0113 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2160 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2160 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2160 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2160 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2160 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2160 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2160 end 0150 end 0150 end 0151 H MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	008	7;			
0024 0042 0090 movuf ACCbLO ; Move LSB into ACCbLO 0025 0046 0091 movuf CCPRIH,W ; Move MSB into ACCbHI 0027 304A 0093 movuf ACCbHI ; Move MSB into ACCAHI 0028 00A0 0094 movuf ACCaLO ; Move LSB into ACCAHI 0028 00A1 0095 clrf ACCaLI ; Clear MSB (ACCaHI) 0024 0096 call D_divF ; Call 16-bit/8-bit routine 0097 ; Which is described in 0097 ; Application Note 544 0022 0100 subwf ACCLO,F ; Round up 0022 0101 btfsc STATUS,Z ; Check low byte for wrap around 0030 0A3 0104 incf ACCbH,F ; High byte didn't wrap 0031 1003 0105 btfss STATUS,Z ; Check high byte for wrap around 0033 0107 ovr_flo ; Status in the sec before clearing HW. ; 0033 0109 clrf ACCbLO ; Mait 100 msec before enabling HW. <	0022 1010 008	8 bcf	T1CON,0	;	Turn off TMR1
0025 0816 0091 movf CCPRIH,W ; Move MSB into W 0026 00A3 0092 movuf ACCHI ; Move MSB into ACCHI 0027 00A4 0093 movuf ACCLAI ; Move MSB into ACCALO 0028 00A0 0094 movuf ACCALO ; Move MSB into ACCALO 0028 00A0 0094 movuf ACCALO ; Move SB into ACCALO 0028 01A1 0095 clrf ACCALI ; Clear MSB (ACCALI) 0020 01A1 0095 clrf ACCALI ; Clear MSB (ACCALI) 0021 0103 0096 ; Application Note 544 0097 0022 0224 0100 subwf ACCLO,W ; we should round up 0021 0030 0101 btfsc STATUS,Z ; Check how byte for wrap around 0021 1003 0103 btfsc STATUS,Z ; Check high byte for wrap around 0031 1003 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 2835 0106 got done ; High byte didn't wrap 0033 0107 ovr_flo 0033 0107 ovr_flo josable INIT josable INIT 0035 0110 done josable BINH josable BINH	0023 0815 008	9 movf	CCPR1L,W	;	Move LSB into W
0026 00A3 0092 movuf ACCHH ; Move MSB into ACCHI 0027 304A 0093 movuf 0x4A ; Move 75usec/0.501n into W 0028 00A0 0094 movuf ACCaLO ; Move ASB into ACCALO 0029 01A1 0095 clrf ACCaLI ; Clear MSB (ACCAHI) 0024 2087 0096 call D_divF ; Call 16-bit/8-bit routine 0097 ; Which is described in 0098 ; Application Note 544 0022 0220 0100 subwf ACCLO,F ; Round up 0021 0103 0105 btfsc STATUS,Z ; Check low byte for wrap around 0030 0A3 0104 incf ACCHI,F ; If LSB wrapped, increment high byte 0031 1003 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 2835 0106 goto done ; High byte didn't wrap 0033 01A2 0108 clrf ACCHI incs	0024 00A2 009	0 movwf	ACCbLO	;	Move LSB into ACCbLO
0027 304A 0093 movlw 0x4A ; Move 75usec/0.50in into W 0028 00A0 0094 movwf ACCaLo ; Move LSB into ACCALO 0029 01A1 0095 clrf ACCaLo ; Move 75usec/0.50in into W 0024 208F 0096 call D_divF ; Clear MSB (ACCaHI) 0028 208F 0096 call D_divF ; Call 16-bit/8-bit routine 0028 3025 0099 movlw 0x25 ; Check remainder to see if 0020 1803 0101 btfsc STATUS, CARRY ; If Remainder < (0.5 * Divisor), skip	0025 0816 009	1 movf	CCPR1H,W	;	Move MSB into W
0028 00A0 0094 movwf ACCaLO ; Move LSB into ACCaLO 0029 01A1 0095 clf ACCaHI ; Clear MSB (ACCAHI 002A 208F 0096 call D_divF ; Clear MSB (ACCAHI 0097 ; which is described in ; Application Note 544 002B 3025 0099 movlw 0x25 ; Check remainder to see if 002C 0224 0100 subwf ACCcLO,W ; we should round up 002D 1033 0101 btfsc STATUS,CARRY ; If Remainder < (0.5 * Divisor), skip	0026 00A3 009	2 movwf	ACCbHI	;	Move MSB into ACCbHI
0029 01A1 0095 clrf ACCAHI ; Clear MSB (ACCaHI) 002A 208F 0096 cal1 D_divF ; Cal1 16-bit/8-bit routine 0097 ; which is described in 0098 ; Application Note 544 002B 3025 0099 movlw 0x25 ; Check remainder to see if 002D 1030 butfsc STATUS,CARRY ; If Remainder < (0.5 * Divisor), skip	0027 304A 009	3 movlw	0x4A	;	Move 75usec/0.50in into W
002A 208F 0096 call D_divF ; Call 16-bit/8-bit routine 0097 ; which is described in 0098 ; Application Note 544 002B 3025 0099 movlw 0x25 ; Check remainder to see if 002L 0224 0100 subwf ACCLO,W ; we should round up 002D 1803 0101 btfsc STATUS,CARRY ; If Remainder < (0.5 * Divisor), skip	0028 00A0 009	4 movwf	ACCaLO	;	Move LSB into ACCaLO
0097 ; which is described in 0098 ; Application Note 544 002E 3025 0099 moviw 0x25 ; Check remainder to see if 002C 0224 0100 subwf ACCcL0,W ; we should round up 002D 1803 0101 btfsc STATUS,CARRY ; If Remainder < (0.5 * Divisor), skip	0029 01A1 009	5 clrf	ACCaHI	;	Clear MSB (ACCaHI)
0098 ; Application Note 544 002E 3025 0099 movlw 0x25 ; Check remainder to see if 002C 0224 0100 subwf ACCcLO,W ; we should round up 002D 1803 0101 btfsc STATUS,CARRY ; If Remainder < (0.5 * Divisor), skip	002A 208F 009	6 call	D_divF	;	Call 16-bit/8-bit routine
002B 3025 0099 movlw 0x25 ; Check remainder to see if 002C 0224 0100 subwf ACCcL0,W ; we should round up 002D 1803 0101 btfsc STATUS,CARRY ; If Remainder < (0.5 * Divisor), skip	009	7		;	which is described in
002C 0224 0100 subwf ACCcLO,W ; we should round up 002D 1803 0101 btfsc STATUS,CARRY ; If Remainder < (0.5 * Divisor), skip	009	8		;	Application Note 544
002D 1803 0101 btfsc STATUS,CARRY ; If Remainder < (0.5 * Divisor), skip	002B 3025 009	9 movlw	0x25	;	Check remainder to see if
002E 0AA2 0102 incf ACCbLO,F ; Round up 002F 1903 0103 btfsc STATUS,Z ; Check low byte for wrap around 0030 0AA3 0104 incf ACCbHI,F ; If LSB wrapped, increment high byte 0031 1003 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 0105 btfss STATUS,Z ; Check high byte for wrap around 0033 0107 ovr_flo ; High byte didn't wrap 0033 0107 ovr_flo ; 0034 01A3 0109 clrf ACCbHI 0035 0110 done ; Disable INIT 0036 1009 0112 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr	002C 0224 010	0 subwf	ACCcLO,W	;	we should round up
002F 1903 0103 btfsc STATUS,Z ; Check low byte for wrap around 0030 0AA3 0104 incf ACCbHI,F ; If LSB wrapped, increment high byte 0031 1D03 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 2835 0106 goto done ; High byte didn't wrap 0033 0107 ovr_flo 0033 0107 ovr_flo 0033 012 0108 clrf ACCbLO 0034 01A3 0109 clrf ACCbHI 0035 0110 done 0035 0110 done 0036 1009 0112 bcf PORT_E,0 ; Disable INIT 0038 21FD 0114 call DEL_100 ; Wait 100 msec before clearing HW. 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr 0116 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	002D 1803 010	1 btfsc	STATUS , CARRY	;	If Remainder < (0.5 * Divisor), skip
0030 0AA3 0104 incf ACCbHI,F ; If LSB wrapped, increment high byte 0031 1D03 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 2835 0106 goto done ; High byte didn't wrap 0033 0107 ovr_flo 0033 0107 ovr_flo 0033 0108 clrf ACCbLO 0034 01A3 0109 clrf ACCbHI 0035 0110 done 0035 0110 done 0036 1009 0111 call DEL_100 ; Wait 100 msec before clearing HW. 0036 1009 0112 bcf PORT_E,0 ; Disable INIT 0037 1089 0113 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr 0116 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	002E 0AA2 010	2 incf	ACCbLO,F	;	Round up
0031 1D03 0105 btfss STATUS,Z ; Check high byte for wrap around 0032 2835 0106 goto done ; High byte didn't wrap 0033 0107 ovr_flo 0033 0107 ovr_flo 0033 0107 ovr_flo 0033 0108 clrf ACCbLO 0034 01A3 0109 clrf ACCbHI 0035 0110 done ; Wait 100 msec before clearing HW. 0036 1009 0112 bcf PORT_E,0 ; Disable INIT 0037 1089 0113 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr	002F 1903 010	3 btfsc	STATUS , Z	;	Check low byte for wrap around
0032 2835 0106 goto done ; High byte didn't wrap 0033 0107 ovr_flo 0033 0102 0108 clrf ACCbLO 0034 01A3 0109 clrf ACCbHI 0035 0110 done 0035 21FD 0111 call DEL_100 ; Wait 100 msec before clearing HW. 0036 1009 0112 bcf PORT_E,0 ; Disable INIT 0037 1089 0113 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr 0116 0120 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	0030 0AA3 010	4 incf	ACCbHI,F	;	If LSB wrapped, increment high byte
0033 0107 ovr_flo 0033 01A2 0108 clrf ACCbLO 0034 01A3 0109 clrf ACCbHI 0035 0110 done 0035 21FD 0111 call DEL_100 ; Wait 100 msec before clearing HW. 0036 1009 0112 bcf PORT_E,0 ; Disable INIT 0037 1089 0113 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr 0116 0120 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	0031 1D03 010	5 btfss	STATUS , Z	;	Check high byte for wrap around
0033 01A2 0108 clrf ACCbLO 0034 01A3 0109 clrf ACCbHI 0035 0110 done 0035 21FD 0111 call DEL_100 ; Wait 100 msec before clearing HW. 0036 1009 0112 bcf PORT_E,0 ; Disable INIT 0037 1089 0113 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr 0116 0120 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	0032 2835 010	6 goto	done	;	High byte didn't wrap
0034 01A3 0109 clrf ACCbHI 0035 0110 done 0035 21FD 0111 call DEL_100 ; Wait 100 msec before clearing HW. 0036 1009 0112 bcf PORT_E,0 ; Disable INIT 0037 1089 0113 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr 0116 0116 0120 0149 0150 end 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 00000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	0033 010	7 ovr_fl	C		
0035 0110 done 0035 21FD 0111 call DEL_100 ; Wait 100 msec before clearing HW. 0036 1009 0112 bcf PORT_E,0 ; Disable INIT 0037 1089 0113 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr 0116 0120 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	0033 01A2 010	8 clrf	ACCbLO		
0035 21FD 0111 call DEL_100 ; Wait 100 msec before clearing HW. 0036 1009 0112 bcf PORT_E,0 ; Disable INIT 0037 1089 0113 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr 0116 0120 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	0034 01A3 010	9 clrf	ACCbHI		
0036 1009 0112 bcf PORT_E,0 ; Disable INIT 0037 1089 0113 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr 0116 0120 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	0035 011	0 done			
0037 1089 0113 bcf PORT_E,1 ; Disable BINH 0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr 0116 0120 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX					
0038 21FD 0114 call DEL_100 ; Wait 100 msec before enabling HW. 0039 2812 0115 goto Xdcr 0116 0120 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXX XXXXXXXXXXXXXXXXXXX			—		
0039 2812 0115 goto Xdcr 0116 0120 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX					
0116 0120 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX				;	Wait 100 msec before enabling HW.
0120 0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX	0039 2812 011	5 goto	Xdcr		
0149 0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXX XXXXXXXXXXXXXXXXXX					
0150 end 0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX	012	0			
0151 MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX	014	9			
<pre>MEMORY USAGE MAP ('X' = Used, '-' = Unused) 0000 : XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX</pre>					
0000 : XXXXXXXXXXXXXX XXXXXXXXXXXXXXXXX					
0040 : All other memory blocks unused. Errors : 0 Warnings : 0					
All other memory blocks unused. Errors : 0 Warnings : 0					
Errors : 0 Warnings : 0					
Warnings : 0		-	nused.		
-					
Messages : U	-				
	messages :	J			

WORLDWIDE SALES & SERVICE

AMERICAS

Corporate Office

Microchip Technology Inc. 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 602 786-7200 Fax: 602 786-7277

Atlanta

Microchip Technology Inc. 500 Sugar Mill Road, Suite 200B Atlanta, GA 30350 Tel: 404 640-0034 Fax: 404 640-0307 Boston Microchip Technology Inc. Five The Mountain Road, Suite 120

Framingham, MA 01701 Tel: 508 820-3334 Fax: 508 820-4326 After July 31, 1995: Tel: 508 480-9990 Fax: 508 420-8575 Chicago

Microchip Technology Inc. 333 Pierce Road, Suite 180 Itasca. IL 60143 Tel: 708 285-0071 Fax: 708 285-0075

Dallas

Microchip Technology Inc. 14651 Dallas Parkway, Suite 816 Dallas, TX 75240-8809 Tel: 214 991-7177 Fax: 214 991-8588

Dayton

Microchip Technology Inc. 35 Rockridge Road Englewood, OH 45322 Tel: 513 832-2543 Fax: 513 832-2841

Los Angeles

Microchip Technology Inc. 18201 Von Karman, Suite 455 Irvine, CA 92715 Tel: 714 263-1888 Fax: 714 263-1338 **New York**

Microchip Technology Inc.

150 Motor Parkway, Suite 416 Hauppauge, NY 11788 Tel: 516 273-5305 Fax: 516 273-5335

AMERICAS (continued) San Jose

Microchip Technology Inc. 2107 North First Street, Suite 590 San Jose, CA 95131 Tel: 408 436-7950 Fax: 408 436-7955

ASIA/PACIFIC

Hong Kong Microchip Technology Inc. Unit No. 3002-3004, Tower 1 Metroplaza 223 Hing Fong Road Kwai Fong, N.T. Hong Kong Tel: 852 2 401 1200 Fax: 852 2 401 3431 Korea Microchip Technology Korea 168-1, Youngbo Bldg. 3 Floor Samsung-Dong, Kangnam-Ku, Seoul, Korea Tel: 82 2 554 7200 Fax: 82 2 558 5934 Singapore Microchip Technology Inc. 200 Middle Road #10-03 Prime Centre Singapore 0718 Tel: 65 334 8870 Fax: 65 334 8850 Taiwan Microchip Technology Taiwan 10F-1C 207 Tung Hua North Road Taipei, Taiwan, ROC Tel: 886 2 717 7175 Fax: 886 2 545 0139

EUROPE

United Kingdom Arizona Microchip Technology Ltd. Unit 6, The Courtyard Meadow Bank, Furlong Road Bourne End, Buckinghamshire SL8 5AJ Tel: 44 0 1628 851077 Fax: 44 0 1628 850259 France Arizona Microchip Technology SARL 2 Rue du Buisson aux Fraises 91300 Massy - France Tel: 33 1 69 53 63 20 Fax: 33 1 69 30 90 79 Germany Arizona Microchip Technology GmbH Gustav-Heinemann-Ring 125 D-81739 Muenchen, Germany Tel: 49 89 627 144 0 Fax: 49 89 627 144 44 Italy

Arizona Microchip Technology SRL Centro Direzionale Colleoni Palazzo Pegaso Ingresso No. 2 Via Paracelso 23, 20041 Agrate Brianza (MI) Italy Tel: 39 039 689 9939 Fax: 39 039 689 9883

JAPAN

Microchip Technology Intl. Inc. Benex S-1 6F 3-18-20, Shin Yokohama Kohoku-Ku, Yokohama Kanagawa 222 Japan Tel: 81 45 471 6166 Fax: 81 45 471 6122



Printed in the USA, 6/6/95 © 1995, Microchip Technology Inc.

Thormation contained in this publication regarding device applications and the like is intended by way of suggestion only. No representation of warranty is given and no fability is assumed by Microchip Technology. Inc. with respect to the accuracy or use of such information. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip The Microchip's good name are trademarks of Microchip Technology Incorporated. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.