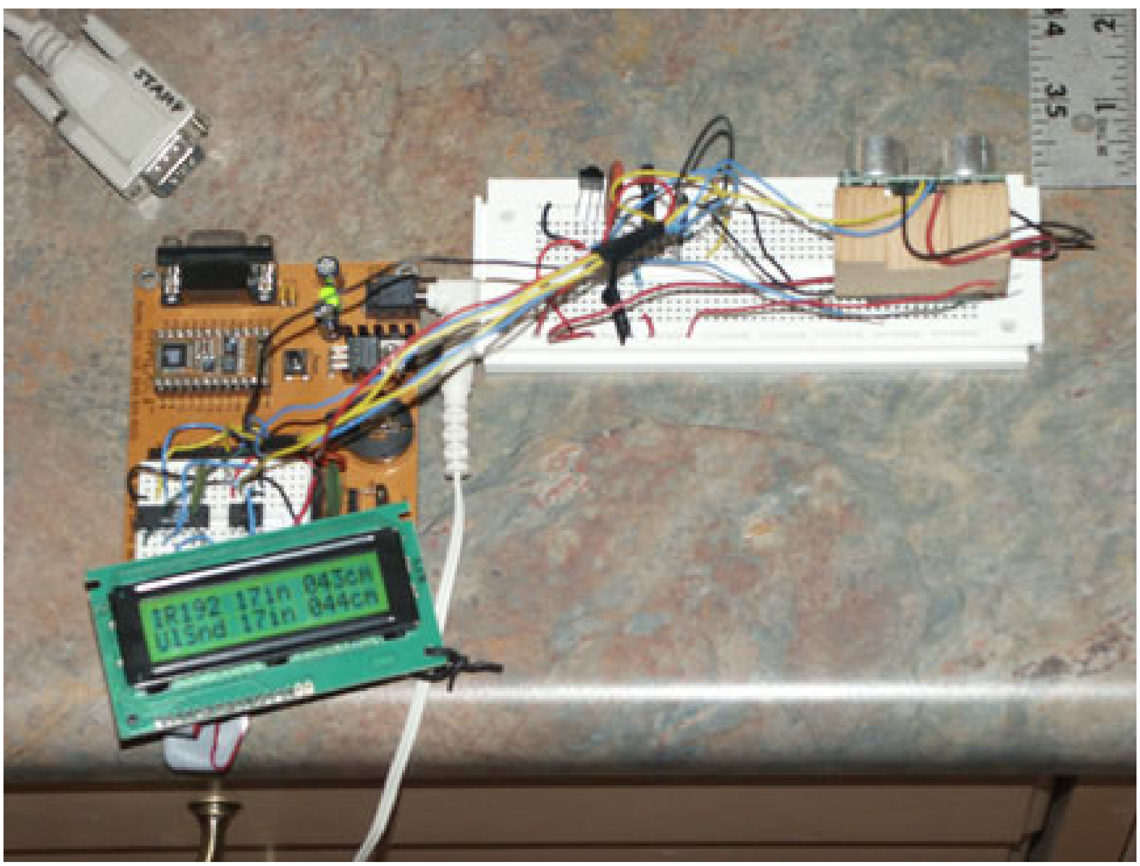
Robot Wall Sensors, Sound vs. Light

By Will Plachno
1/7/03
Science per. 2
C.T.English



1. Table of Contents

2. List of Figures

3. Purpose

The purpose of this experiment is to find which sensor (Infrared light or Ultrasonic sound), detects walls more accurately.

4. Background Research

The physical nature between light and sound will be the major cause for which one is more dependable for measuring distances.

4.A What Is Sound?

When I strike a string of my guitar, it vibrates back and forth. But when it gets to one side it pushes the air in front of it together while leaving less air in back of it. This situation of high-pressure and low-pressure is unstable. The air in the high-pressure will try to move to the areas of low-pressure. This over compensates and the areas of low-pressure become areas of high-pressure. This wave also moves very well and very fast through the air. The speed of sound is a constant. Since the air doesn't have to move very far, just from one area of high-pressure to an adjacent area of low-pressure, the sound wave propagates very fast. Lord Walter Rayleigh published a comprehensive book called *The Theory of Sound*[1] in 1877.

Shown in figure 1 are the frequencies for each musical note and the common range for each instrument. The best a human being can hear is between 20 Hz and 20 KHz. The Ultrasonic sensor I will use is at 40 KHz. This is above what humans can hear. As you can see middle C is 261.63 Hz. That means the wave vibrates 261.63 times a second. This is the frequency of the sound wave. Another way of looking at the same thing would be the wavelength. While the frequency is defined as how many times it vibrates a second, the wave length is defined as the distance the wave propagates to complete one vibration or cycle. Waves with higher (larger) frequencies have a shorter wavelength.

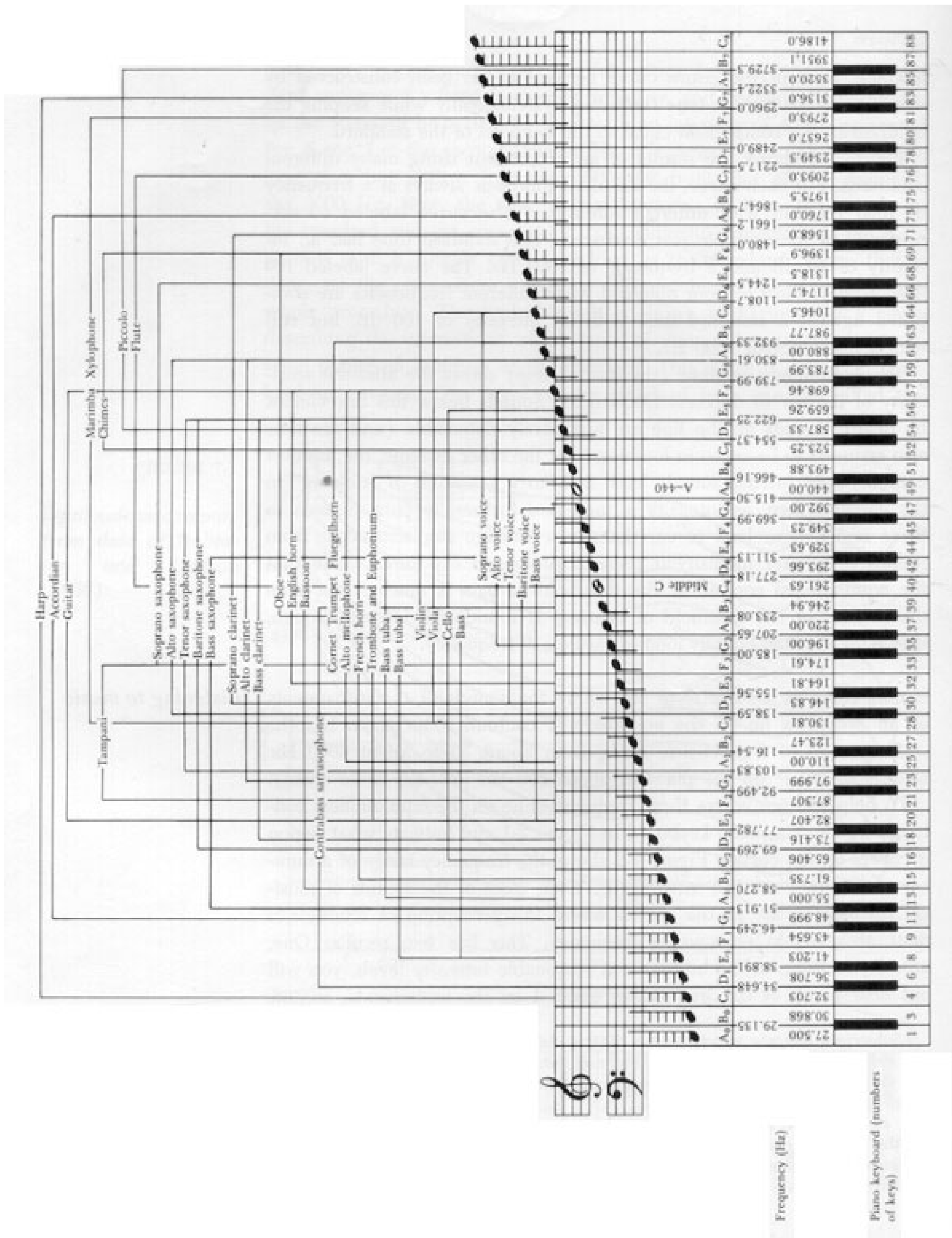
4.B Ultrasound Range Detection Technique

In *Basic Stamp Application Notes, Chapter 12, Sonar Rangefinding*[2], it explains the technique I will be using. This technique is easy. I have a speaker that puts out a sound ping into 40 KHz then I measure the time it takes to come back and to be heard by a microphone. I will take that time and multiply by the speed of sound and then divide by 2 because of the echo, which calculates the distance to the object.

4.C What Is Light?

Early scientists had a hard time determining if light was a wave or a stream of particles[3]. Light sometimes looks like a wave because it showed the same interference patterns that a wave of sound or water would show. Faraday[4] (1821-1860) was interested in experimenting with how electricity could create a magnet and the reverse of how a magnet can be used to create electricity. Maxwell wrote mathematical equations based on Faraday's work and published them in: *A Treatise of Electricity and Magnetism*[5].

Figure 1. Sound Frequencies[6]



Maxwell wondered what would happen if an electric field generated a magnetic field, and that magnetic field generated the original electric field thus sustaining itself. Using his equations, he calculated how fast that wave would have to be, and that number turned out to be the measured speed of light. Show in figure 2 is Maxwell's electromagnetic wave.

Shown in figure 3 is the Electromagnetic Spectrum. Light is a very small part of the electromagnetic spectrum. The different frequencies correspond to what we see at different colors. We see from red, the lowest frequency, to violet, the highest frequency. Infrared is below, in frequency, what we can see.

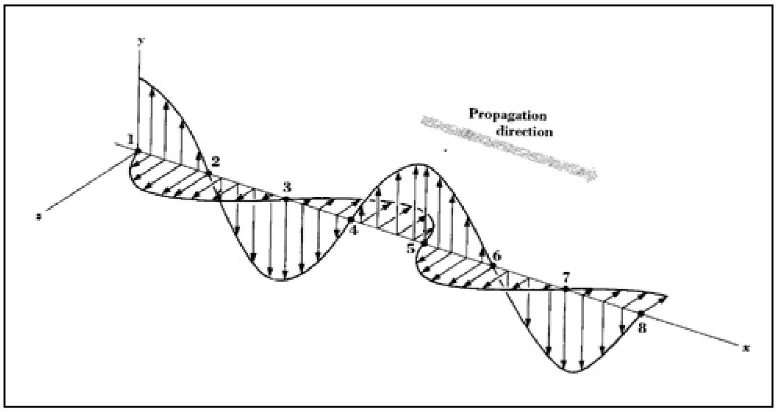


Figure 2 Maxwell's electromagnetic wave[7].

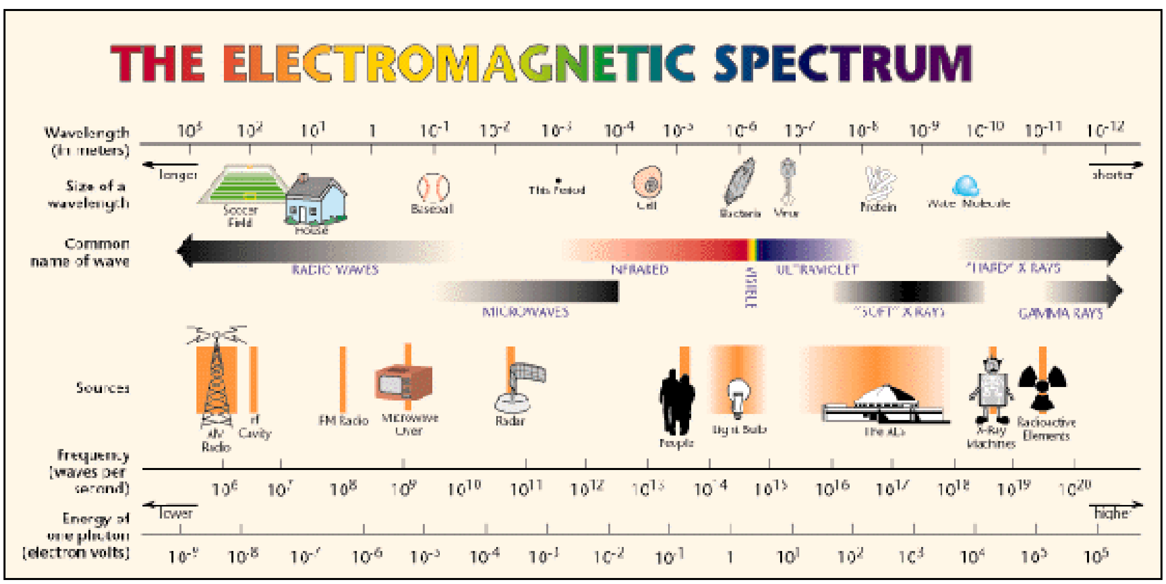


Figure 3. the Electromagnetic Spectrum[8].

4.D Infrared Range Detection Technique

It is going to be harder to use light more then sound to measure distances. The speed of light is too fast to measure using a microprocessor, so I can't use the same technique as I did for sound. I might be able to measure the strength of the light reflected off an object to determine its distance. However, I can not do this since the infrared receivers only tell you if they see an infrared light or not, not the strength of the light.

In the *Stamp Weekend Application Kit*, (see citation 9) it describes the method for distance measurement. The infrared receiver has a filter in it that tries to look at infrared light that is flashing on and off at 40,000 times per second. In this article, they try to change the frequency of the transmitted infrared light using a program on the Stamp. To have the infrared detector's filter not see the light, the transmitted frequency must be further from the center frequency of 40 KHz. The frequency that first sees the light reflection is used to determine the distance to the object. This paper's technique of generating the transmitted frequency is really not good enough for my science project. Because of their poor technique, they claim to only see 5 different distances in their measurements. I want to measure with much more accuracy so I will use a different technique for generating the transmitted frequency.

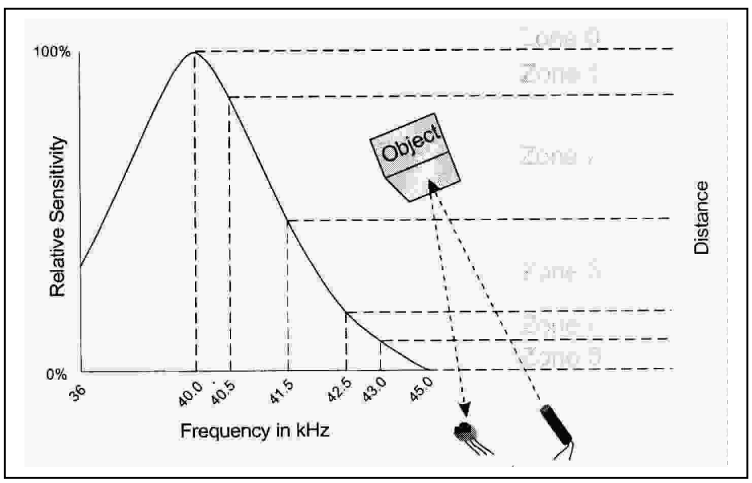


Figure 4. Infrared Measurement Ranges[9].

5. Hypothesis

If I use infrared and ultrasound sensors, than I think the ultrasound sensor will be more reliable measuring distance to different materials of different colors.

6. Materials

I'm going to build a test apparatus that will test both the ultrasonic sensor and the infrared sensor at the same time. To do this, I will make it microprocessor controlled and use the same stamp that I used for last year's science project. The Stamp is a BS2p, and I am using a BS2p demo board[10]. It has a built in breadboard, a serial-port connector (which I will be using to connect to the PC and download the programs), a 9 volt power brick, and a LCD connection (which I have already used). Last year, I created a custom PCB board, but for this year, I shall not send in for a PCB board, but prototype the circuit on the breadboard. This breadboard allows me to debug each piece as I wire the project.

6.A The SRF04, The Ultrasonic Sensor

In both hardware and software, the Ultrasonic will be much easier than the Infrared. I shall use the Devantech SRF04 Ultrasonic Range Finder[11]. It has a printed circuit board with two Ultrasonic Transducers (a speaker and a microphone, although they do both look the same.)

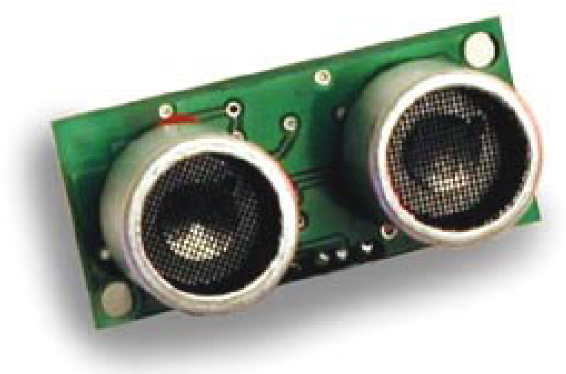
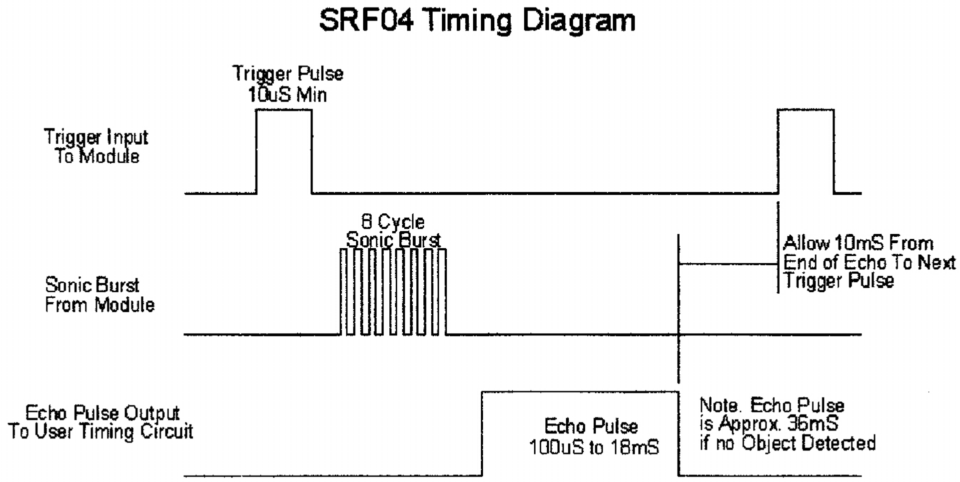


Figure 5. This is a close-up photo of the Devantech SRF04 Ultrasonic Range Finder.

The Devantech Ultrasonic Range Finder has four wires that I shall connect to the BS2p Demo Board. One of the wires is ground, while another is five volts. This leaves two wires, which relay information between the SRF04 and the BS2p. One of the wires is an input, and the other is an output. The output from the Stamp is a trigger pulse that will tell the SRF04 to start taking a measurement. This is an example of the Master-Slave protocol. This means that it will send me information only after I tell it to (in this case it happens to be after the trigger pulse.) For the project to work, I shall make the trigger pulse more then ten microseconds by changing the software.

All in all, I am sending out a ten microsecond trigger pulse and then, there will be 8 sonic bursts from The Devantech SRF04. Then, there will be an echo pulse that may be anywhere from one hundred microseconds to 36 milliseconds. After this complicated composition, ten milliseconds shall be allowed until the next trigger pulse.

Figure 6 SRF04 timing diagram



I shall measure the echo pulse to determine how many inches away from an object it is. There is a command in the stamp that measures the pulse width. This command is called pulsin. Sadly, there still are conversions I shall have to make. For example, I shall have to divide by .75 (divide by .75 because the BS2p pulsin reads .75 microseconds per unit.), multiply by the speed of sound, and then multiply by 2 (I have to multiply by 2 because of the fact that the sound has to go to the object and echo back.)

1/0.75 (unit/µs) x 73.746 (µs/inch) x 2 (echo) = 197 (units/inch)

Therefore, the number I read from pulsin, I shall have to divide by 197 in order to achieve the distance in inches.

2.54 cm = 1 inch

1/0.75 (unit/µs) x 73.746 (µs/inch) x 1/ 2.54 x 2 (echo) = 77 (units/cm)

I cannot divide by a fraction inside the Stamp, so I must round to the nearest full integer.

6.B Infrared

The infrared will use more hardware and software than the ultrasonic range finder. Physically you have an IR transmitter and an IR detector. The transmitter shall put out light that is invisible to the human eye but will be detected by the detector. The detector has a pin that is normally high but goes low if it sees the light put out by the transmitter. The detector also requires that the light is flashing off and on at 40,000 times a second. I am going to use an IC called an NE555[12] to oscillate the transmitter LED at 40 KHz.

The scheme for the IR is a lot more involved as I shall be detuning the transmitter (making the detected signal weaker). If the transmitter is at 40 KHz, the detector will be able to sense a reflected signal farther away. However, at 43 KHz, the detector must be closer to the wall to work. After I build the apparatus, I shall calibrate the detuning to the distance to the wall. I will write a program that starts at 40 khz and increases the frequency until the sensor does not sense the wall. I will calibrate the frequency that the sensor dies at to the distance to the wall.

This is very much like trying to find how far away from a radio station you are by detuning with the tuning knob. If I'm close to that radio station, I can have it slightly off that frequency and still be able to hear it. If I'm faraway, I have to have the frequency tuned exactly to the radio station. So by playing with the tuner I can tell how faraway I am from the station. This is very similar to what I'm doing for the infrared except that I'm changing the frequency of the transmitter. The infrared receiver that I'm using has a built in filter that I have no control over.

The NE555 uses two resistors and a capacitor to set the frequency that it oscillates at. A program on the Stamp will control the frequency by changing the value of one of the resistors. To do this I will use another IC called a DS1267. The DS1267 is a digital resistor that's value can be easily changed by writing a number to it from the Stamp. This IC uses a three wire interface like the thermometers that I used last year.

Figure 7 Stamp I/O Pin Definitions

I/O Pin	Description	I/O Pin	Description
15	Not used	7	LCD D3
14	Not used	6	LCD D2
13	IR Detecter	5	LCD D1
12	DS1267 CLK	4	LCD D0
11	DS1267 RST	3	LCD Reg Select
10	DS1267 D	2	LCD R/W
9	ECHO (blue)	1	LCD Enable
8	INIT (yellow)	0	Not Used

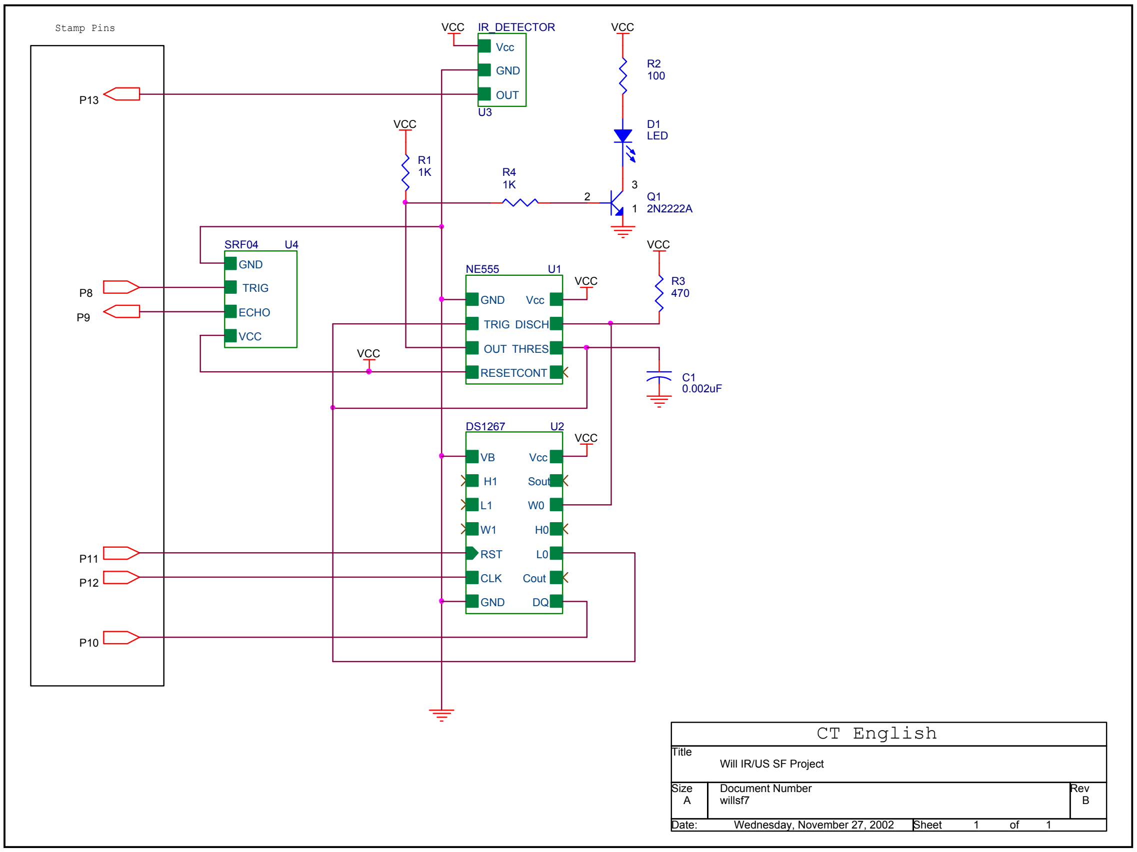


Figure 8 Circuit schematic.

The circuit schematic shows both the infrared sensor and the ultrasonic sensor. The Infrared is pretty much all of it except the Devantech SRF04, which happens to be the whole ultrasonic sensor. Yet, the infrared has only one component for detecting, the rest is for the IR transmitter. The 555 are the 40 kHz (variable) oscillator, while the DS1267 is a component that is used to change the frequency for the 555. There will be an external transistor that I shall use to make the LED brighter which therefore changes the distance that I can get a strong detection from. There are mainly three articles that I have used to learn about the DS1267, the external transistor, and the Infrared system. The names of the readings are "Stamp Weekend Application Kit-Infrared Emitting Diode and 40 kHz Infrared detector,"(see note 9) which tells about the Infrared, the "BASIC Stamp Application Notes, 11: Infrared Communication,"[13] used for the transistor, and "Need Analog Output From The Stamp? Dial It In With A Digital Potentiometer"[14] which shows the data about the DS1267.

6.C Bill of Materials

An important consideration in choosing infrared vs. ultrasonic is the cost of each component.

Description	Price
IR Transmitter	$5.20
IR Receiver	$2.60
DS1267 10K digital potentiometer	$8.32
NE555 Timer	$0.50
Devantech Ultrasonic Range Finder	$30.00

Figure 9
Bill of Materials

This table shows the main components and their prices. For my science fair project, I want more control over the apparatus, so I used more components. It's more of a fair comparison if you combine the prices of the IR transmitter and the IR receiver, but not the prices of the DS1267 and the timer. If you do this, the prices are $7.80 for the infrared system, and $30 for the ultrasonic.

Figure 10 Component purchasing websites

Description	Website
IR Trans	http://www.parallax.com/detail.asp?product_id=350-00017
IR Rec	http://www.parallax.com/detail.asp?product_id=350-00014
DS1267	http://www.digikey.com/scripts/us/dksus.dll?Detail?Ref=62364&Row=153778
NE555	http://www.digikey.com/scripts/us/dksus.dll?Detail?Ref=62468&Row=46772
Devantech	http://www.parallax.com/detail.asp?product_id=28015

This table shows the components and the websites that I used to buy them. Yet, these may not be the cheapest prices for the circuits. There are similar circuits that you could buy from another website. For example, I could've bought the IR Receiver from Digikey instead of Parallax, and it would've been cheaper. However, it would not have been the exact same part I would have to do a lot more designing.

TAB Robot IR size	Devantech Size
2.5 cm x 1.5 cm	4.5 cm x 1.5 cm

Figure 11
Layout Size
Comparison

Other than the difference in the price comparison, another important comparison is how much room each approach takes up on the printed circuit board. I had bought a small little robot[15] that uses infrared and I compared the amount of room the infrared took on there to the amount of room the ultrasonic took up on the Devantech. Shown in figure 11 is the measurement comparison. The length difference is 55% or almost half.

6.D DS1267 Calibration

The DS1267 is a variable resistor that you can control by writing a number to it from the stamp. I breadboarded the chip so I could test out the resistance values I will get. To the stamp, the DS1267 uses a three-wire interface. The figure 12 is the subroutine I used to communicate to the DS1267. This subroutine is based upon code from the article *Need analog output from the STAMP? Dial it in with a digital potentiometer.* The 1267 has two variable resistor I can control, I will only use the resistor between L0 and W0.

(Figure 12. The DS1267 subroutine)

```
''''''''''''''''''''''
' Test Loop
''''''''''''''''''''''
for DSpot0 = 0 to 255 step 16
 gosub outPot
 lcdout 0, 128, [dec3 DSpot0]
 pause 8000
next
stop


' ====================DS1267 SUBROUTINE=======================
' This code shifts data out to the DS1267. Since it uses the Shiftout
' instruction, which does not alter the variable being shifted, we don't
' have to make a copy of the pot data.DSpot\16 means 16 bits shifted out.
outPot:
 high reset1267
 pulsout clock1267, 1          ' Pulse for stack-select bit
 Shiftout Data1267, clock1267, msbfirst, [DSpots\16]
 low reset1267
return
```

To use this subroutine, the only thing you have to do is assign a number to Dspot0 and call the subroutine. To generate the graph in figure 13, I created a test loop that writes values from 0 to 255 and displays them on the LCD display. The maximum resistance you can get is 10,000 ohms, this is expected because I bought the 10,000 version of the DS1267. The minimum value if 0 is written is 542 ohms.

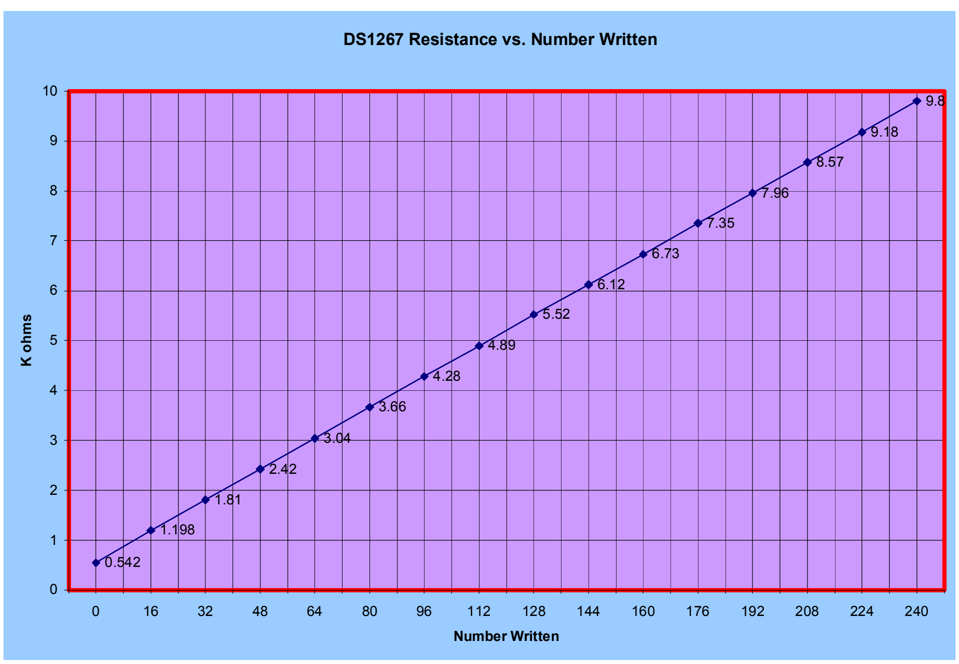


Figure 13

6.E Calibrating the NE555

For this graph, which is shown in figure 16, I bread boarded the NE555 to use the resistor from the 1267. The NE555 is a timer that oscillates the frequency. The frequency that it oscillates at is reliant on 2 resistors and a capacitor.

$$\text{Frequency} \approx \frac{1.44}{(R_A + 2R_B)\,C}$$

$R_A = 470$, C= .002uF, R_B= the resistor from the DS1267.

For a frequency of 40KHz, I need a resistor from the DS1267 of approximately 8765. Looking at figure 7, I approximate that 8765 requires a number of 210. Figure 14 shows the test loop that I used to generate the graph in figure 16. This shows that the frequency varies from 46KHz (value=180) to 40.5 KHz (value=210).

Figure 15 shows the oscilloscope trace for the test loop with the 6 second pause commented out. This shows all the frequencies blurred together. The x axis is 25 microseconds per major division. The longest oscillation is going 5 divisions.

$1/25\mu s = 40$ KHz

The fastest is going 4.3 divisions or 21.5 µs (46.5 KHz).

```
''''''''''''''''''''''
' Test Loop
''''''''''''''''''''''
start:
for DSpot0 = 180 to 210
 gosub outPot
 lcdout 0, 128, [dec3
DSpot0]
 ' pause 6000
next
goto start
stop
```

Figure 14
Test Loop For Measuring the Frequencies

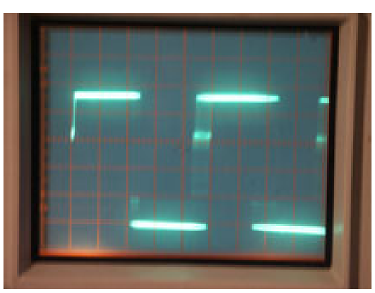
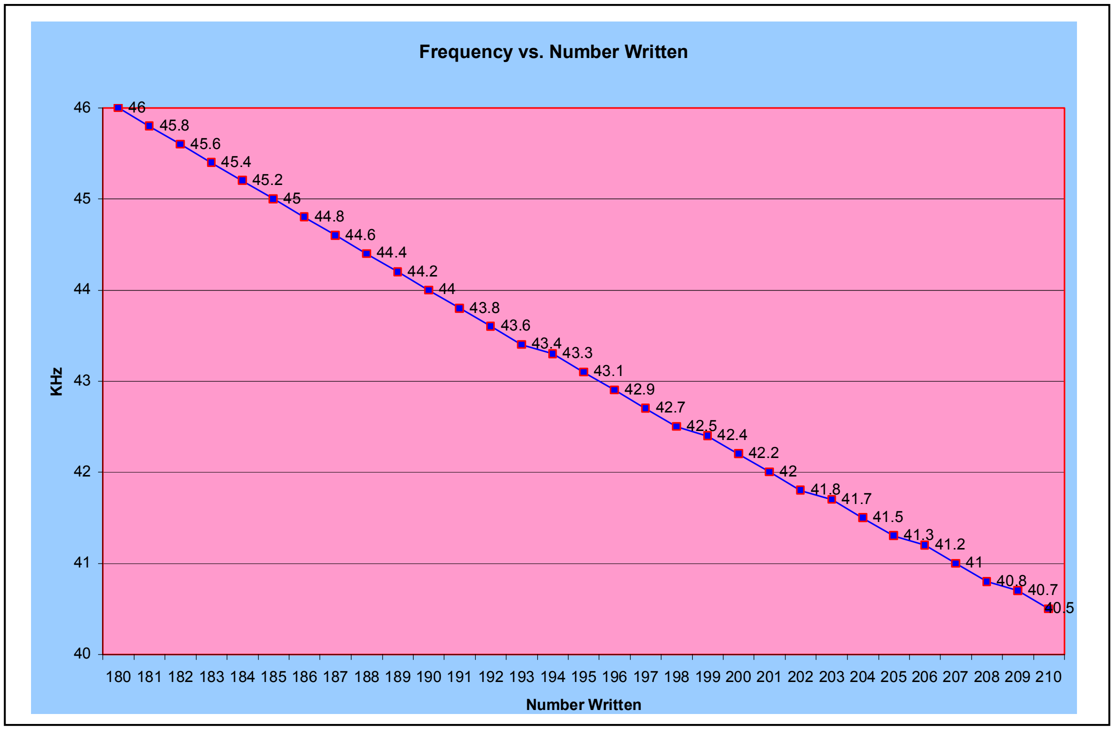


Figure 15 Oscilloscope trace showing frequency change from 40KHz to 46.5 KHz
(X axis is 5us per major division)

Figure 16. Frequency vs. Number Written



6.F Finding the Center Frequency

I had assumed that the receiver worked best at 40 KHz. Actually, I don't care about the actual frequency, just the number I write into the 1267. I had to find a real center number for which the infrared receiver detects the longest distances. The actual center can be off from 40 KHz for many reasons. For example once reason is that maybe the filter inside the receiver was designed for something other than 40 KHz. The capacitor and the resistor on the 555 is only 10% accurate so the frequencies I'm transmitting can be inaccurate. Anyway, I can write a program to find a center under actual test conditions.

In figure 17 is the test program I wrote to find the center frequency. The figure next to it, figure18, are the results for the program. In the results, 1 means that the receiver (pin 13) did not find a reflection. 0 means that the receiver found a reflection. I expected the results to look like a series of ones followed by a series of zeros and then another series of ones. However, as you can see, the actual results were not simple transitions. If the receiver is having trouble seeing a reflection, it can sometimes read zero or one at the transition.

For my purposes, this data does provide me with a center point. The center-point is 227.

Figure 17. Center frequency test program

```
lastin = 1

debug "innit", cr

''''''''''''''''''''''
' Test Loop
''''''''''''''''''''''

for DSpot0 = 220 to 240
 gosub outPot
 pause 2000

 debug DEC DSpot0, "  ", BIN in13, cr

 if lastin = 1 then at1

 at0:
 if in13 = 1 then tran01
 goto done

 at1:
 if in13 = 0 then tran10
 goto done

 tran01:
 DSpot1 = DSpot0 - 1
 lcdout 0, 192, [dec3 DSpot1]
 lastin = 1
 goto done

 tran10:
 lcdout 0, 128, [dec3 DSpot0]
 lastin = 0

 done:
next

alldone:
```

Figure 18 Program output

```
innit
220 1
221 1
222 0
223 1
224 0
225 0
226 0
227 0
228 0
229 0
230 0
231 1
232 0
233 1
234 1
235 1
236 1
237 1
238 1
239 1
240 1
```

6.G Infrared Distance Calibration

In order to make an accurate distance measurement using the infrared, I shall find a relationship between the DS1267 resistor numbers and the distance being measured. In order to do this I wrote a program to do this calibration. The program shown in figure 19 scans the DS1267 values between 185 to 227. 227 were the center frequency I found earlier that should represent the largest distance possible for me with this flaky apparatus. 185 correspond to the closest distance I can measure.

As I start to scan at 185, I expect the IR detector to read 1 as that means it isn't detecting a reflection. When I find a 0, that number relates to the distance to the object. When I first started out I was getting really inconsistent readings so I made the program require that it detects a reflection consistently 3 times for the same reading, even though I found the measurements to be pretty flaky. I found that I had to hold the target in a straight line in front of the infrared diodes, which were slightly angled upwards. For this experiment, I used my mothers bread box, which is shiny white and should do an excellent job of reflecting the light beam.

Shown in figure 20 is the data table for my measurements and shown in figure 21 is the graph of the data. As you can see, the data is not a straight line. This would give me problems in translating the measured data to measured inches. The maximum distance I could measure was 48 inches. I did my measurements in inches because I didn't have a rular in centimeters that was long enough.

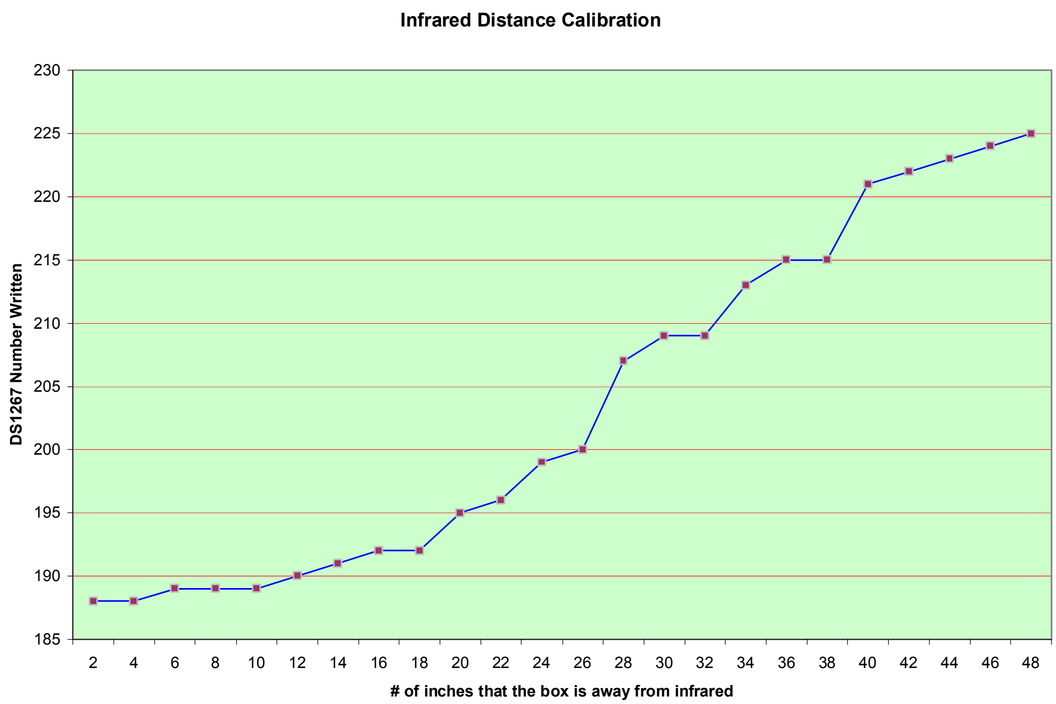
Figure 19. Program for Calibrating the Infrared Distances

```
''''''''''''''''''''''
' Test Loop
''''''''''''''''''''''

for DSpot0 = 185 to 227
 gosub outPot
 pause 500
 thisin = in13
 debug DEC DSpot0, "  ", BIN thisin, cr

 if thisin = 0 then tran10
 goto done

 tran10:
 if in13 = 1 then done
  pause 100
 if in13 = 1 then done
 lcdout 0, 128, [dec3 DSpot0]
 goto alldone

 done:
next

debug DEC DSpot0, "  Too far, move in!",
cr
lcdout 0, 128, ["Too Far"]

alldone:
```

Figure 20. Infrared distance calibration program results

Inches away	DS1267 Value	Inches away	DS1267 Value
2	188	26	200
4	188	28	207
6	189	30	209
8	189	32	209
10	189	34	213
12	190	36	215
14	191	38	215
16	192	40	221
18	192	42	222
20	195	44	223
22	196	46	224
24	199	48	225

Figure 21. Infrared Distance Calibration Graph



6.H Infrared Measurement Translation

In my final program, I need to translate the DS1267 value into the actual distance. I can do this using the previous calibration data. Shown in figure 22 is the translation table to change the DS1267 number to the distance in inches.

DS1267 Value	Inches	cm
186	2	5
187	3	8
188	3	9
189	8	20
190	12	30
191	14	36
192	17	43
193	18	46
194	19	48
195	20	51
196	22	56
197	23	58
198	23	59
199	24	61
200	26	66
201	27	69
202	27	69
203	27	70
204	28	71
205	28	71
206	28	72
207	28	72
208	29	74
209	31	79
210	31	80
211	32	81
212	33	84
213	34	86
214	36	91
215	37	93
216	37	94
217	38	96
218	38	97
219	39	99
220	39	100
221	40	101
222	42	107
223	44	112
224	46	117
225	48	122
226	50	127
227	52	132

Figure 22
Infrared Measurement Translation Table

I made this table from the calibration table by approximating the values that didn't have a measurement. For the centimeter column, I took the inch measurements and multiplied that number by 2.54. Inside the program, I have to translate each DS1267 value into inches and centimeters. There is a basic stamp command called lookup that does this translation in 1 step. The following is a excert from the stamp program that shows how the lookup command is used to find the centimeter value.

```
'the following table takes IRindex and converts
'it to cm as IRcm
 lookup IRindex,
[5,8,9,20,30,36,43,46,48,51,56,58,59,61,66,69,6
9,70,71,71,72,72,74,79,80,81,84,86,91,93,94,96,
97,99,100,101,107,112,117,122,127,132], IRcm
```

The first translation in the lookup table is the index = 0. Therefore, I had to subtract 186 from the DSPot0 to equal IRindex.

6.I Using the LCD

Figure 23. LCD Display Locations

128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207

The LCD display I'm using has 16 columns and 2 rows . This table in figure 23 shows each number used to position the output on the LCD display.

```
lcdout 0, 134, [dec2 IRin, "in"]
```

This example starts the display at location 134 which is in the center of the top line. By using "dec2" it means to write a decimal number with 2 places[16]. This 2 digit number is IRin which is at the locations 134 and 135. Then "in" is written at the next two locations. The output for this example is shown in figure 24.

Figure 24. An Excerpt Showing an Example of How to use the LCD Display

128	129	130	131	132	133	**134** **#**	135 **#**	136 i	137 n	138	139	140	141	142	143
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207

Figure 25
LCD Display Picture

6.J Ultrasound Distance Measuring Software

Shown in figure 6 is the SRF04 Timing Diagram. You must provide a trigger pulse that is greater than 10 microseconds. The Stamp has a built in command called PULSOUT that outputs a pulse that is used as the trigger pulse. Because each different Stamp chip has different timing units, we cannot simply tell it to output a pulse of 10 microseconds because each Stamp runs at a different speed. The BS2P Stamp that I'm using has 1.18 microseconds per unit for the PULSOUT command. So if I use 9 as the PULSOUT command, it will generate a pulse width of 10.62 microseconds.

```
''''''''''''''''''''''
' Ultrasound
''''''''''''''''''''''

 pulsout D_init, 9    '10.62 usec init pulse
 pulsin  D_echo, 1, wdist  ' read high echo pulse
 USin = wdist / convin  ' convert to inches
 UScm = wdist / convcm  ' convert to cm
 lcdout 0, 198, [dec2 USin, "in"]
 lcdout 0, 203, [dec3 UScm, "cm"]
 lcdout 0, 192, ["UlSnd"]
```

Figure 26. Ultrasound Measurement Code Software

I've already explained how to generate the calibration measurements for translating the measured pulse width into inches and centimeters. In Figure 26 is the Ultrasound Measurement Code Software. It is extremely simple compared to what is required for the infrared measurement.

6.K Complete Distance Measurement Software

Figure 27. Complete Distance Measurement Software.

```
'{$STAMP BS2p}

''''''''''''''''''''''''''''''''''''''''''
' I/O pin definitions and other constants
''''''''''''''''''''''''''''''''''''''''''

IR_det_pin         con      13
clock1267          con      12
reset1267          con      11
Data1267           con      10
D_echo             con       9
D_init             con       8

convin             con     197    'echo in inches
convcm             con      78    'echo in cm

''''''''''''''''''''''''''''
' I/O pin direction settings
''''''''''''''''''''''''''''

input  IR_det_pin  ' recieves the infrared signal
output clock1267   ' 3-wire clock
output reset1267   ' 3-wire inable
output Data1267    ' 3-wire data
input  D_echo      ' echo time pulse
output D_init      ' starts the echo reading
```

```
''''''''''''''''''''''''
' Variable declarations
''''''''''''''''''''''''

counter            var    nib
wdist              var    word
USin               var    byte
UScm               var    byte
DSpots             var    word
DSpot0             var    DSPots.lowbyte
DSpot1             var    DSPots.highbyte
thisin             var    bit
IRindex            var    byte
IRin               var    byte
IRcm               var    byte

''''''''''''''''''''''''
' Initialization
''''''''''''''''''''''''

initialize:
' This is the LCD display initialization
pause 1000
lcdcmd 0, %00110000    ' 1st wake up
pause 10
lcdcmd 0, %00110000    ' 2nd wake up
pause 1
lcdcmd 0, %00110000    ' 3rd wake up
pause 1
lcdcmd 0, %00100000    ' 4 bit data bus
pause 1
lcdcmd 0, %00001100    ' display on, no cursor or blink
pause 1
lcdcmd 0, %00101100    ' 2 display lines, 5x10 font
pause 1
lcdcmd 0, %00000001    ' clear display, move cursor home
DSpot1 = 0
low reset1267
debug "innit", cr

'''''''''''''''''''''''
' Infrared
'''''''''''''''''''''''

for DSpot0 = 185 to 227
 gosub outPot
 pause 500
 thisin = in13
 debug DEC DSpot0, "  ", BIN thisin, cr

 if thisin = 0 then tran10
 goto done

 tran10:
 if in13 = 1 then done
  pause 100
 if in13 = 1 then done
 lcdout 0, 128, [dec3 DSpot0]
 goto alldone

 done:
next

alldone:
 gosub urchin


''''''''''''''''''''''
' Ultrasound
''''''''''''''''''''''

 pulsout D_init, 9    '10.62 usec init pulse
 pulsin  D_echo, 1, wdist  ' read high echo pulse
 USin = wdist / convin  ' convert to inches
```

```
 UScm = wdist / convcm  ' convert to cm
 lcdout 0, 198, [dec2 USin, "in"]
 lcdout 0, 203, [dec3 UScm, "cm"]
 lcdout 0, 192, ["UlSnd"]
stop


' ====================DS1267 SUBROUTINE=======================
' This code shifts data out to the DS1267. Since it uses the Shiftout
' instruction, which does not alter the variable being shifted, we don't
' have to make a copy of the pot data.DSpot\16 means 16 bits shifted out.
outPot:
 high reset1267
 pulsout clock1267, 1          ' Pulse for stack-select bit
 Shiftout Data1267, clock1267, msbfirst, [DSpots\16]
 low reset1267
return



' ==========================URCHIN=============================
' this translation transfers a value into in. and cm
' after translating it also prints it to the LCD display
urchin:
 if DSPot0 < 187 then toosml
 if DSPot0 > 226 then toobg
 IRindex = DSPot0 - 186
 'the following table takes IRindex and converts it to inches as IRin
 lookup IRindex,
[2,3,3,8,12,14,17,18,19,20,22,23,23,24,26,27,27,27,28,28,28,28,29,31,31,32,33,34,36,37,37
,38,38,39,39,40,42,44,46,48,50,52],IRin
 'the following table takes IRindex and converts it to cm as IRcm
 lookup IRindex,
[5,8,9,20,30,36,43,46,48,51,56,58,59,61,66,69,69,70,71,71,72,72,74,79,80,81,84,86,91,93,9
4,96,97,99,100,101,107,112,117,122,127,132], IRcm
 lcdout 0, 134, [dec2 IRin, "in"]
 lcdout 0, 139, [dec3 IRcm, "cm"]
 goto IRp

toosml:
 lcdout 0, 134, ["Too close"]
 goto IRp

toobg:
 lcdout 0, 134, ["Too far"]

IRp:
 lcdout 0, 128, ["IR"]
 lcdout 0, 130, [dec3 DSPot0]
return
```

In Figure 27 is the final software that I used for my Science Fair measurements. The I/O Pin Definitions and other Constants section is to allow you to give a name to each of the I/O pins. These constants will always have the value I gave it. These are not instructions that the Stamp executes. Constants allow me to use the name throughout the rest of the program. If I wanted to change the number for D_Echo, I could just change the value at the top of the page.

The next section defines which I/O pins are inputs and outputs from the Stamp. For example the trigger pin is an output from the Stamp to the Devantech while the Echo pin is an input to the Stamp.

The next section is the variable declarations. These reserve memory for the variables I used inside the program. A variable can be a bit, a nibble (four bits), a byte (eight bits), or a word (16 bits).The value that I shift out to the DS1267 is 16 bits, or a word. You can set one of the two potentiometers through Dspot0.

The next section, Initialization, initializes the LCD Display. This is the exact code that I used in last year's Science Project. This is the first section of which the Stamp actually executes instructions.

The next section, Infrared, is the code to do the Infrared measurements. I pretty much have described how this works in the previous sections. I started off writing a Dspot0 value that would have the most detuned frequency. From my calibration measurements, this would be a number that corresponds to the closest distance that I could measure with the Infrared. In the for-loop, I keep increasing the value until I see a reflection from the IR Detection pin. You have to read a reflection three times in a row for it to be valid. That hopefully gets rid of any noisy readings right at the transition. I translated the measurement to inches and centimeters using the subroutine urchin.

The next section is the ultrasonic measurement. This is very simple, for all you have to do is output the trigger pulse and then read the echo pulse width. The echo pulse width is converted to inches and centimeters using the conversion constants that were calculated earlier.

The next section is the DS1267 Subroutine, which simply writes out the Dspot0 number to the DS1267 chip. It uses a three-wire interface similar to the thermometers from my last year's science fair project.

The last section is the subroutine that converts the Infrared measurement into both inches and centimeters. I used the lookup command, which I described previously.

I used the PC to type in this program. When I'm ready, I check for syntax errors and click on run which transfers the program from the PC to the Stamp. I only have to do that once. To rerun the program for every measurement, I press the reset button on the Stamp. The LCD will display both the Ultrasonic and the infrared measurements which I can record for each measurement.

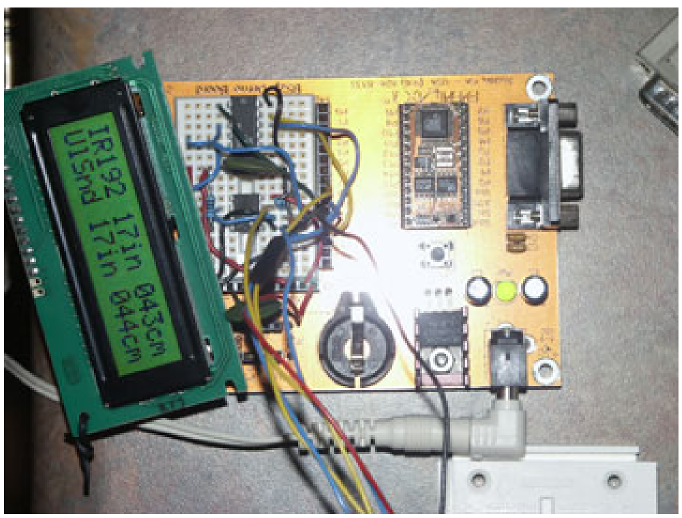
6.L Accuracy Versus Resolution

6.L is Accuracy versus Resolution. When I wrote the program, the smallest output I increment are 1 inch and 1 centimeter. That is called the resolution of my measurements. However, the accuracy of it is much worse than that. From looking at the graph, I can determine that the accuracy is about three inches. Most people would assume that the accuracy is the same as the resolution, but that is not true here.

6.M The Completed Hardware

Figure 29. A picture of the Stamp board where the LCD display, the processor, the serial port for connecting to the computer, and the power source are. Also on the breadboard section, I wired in the DS1267 and the NE555 oscillator.

Figure 29. The Stamp Board



In figure 30 is a picture of the sensor board, where the sensors are kept. The wires between the stamp board and the sensor board transport the measurements that were just taken. There are control signals, power and ground, and data signal wires for the measurements.

Figure 30. The Sensor Board

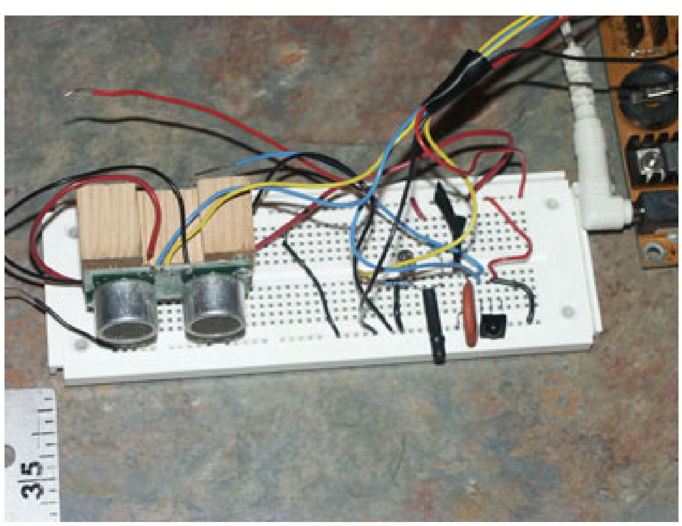
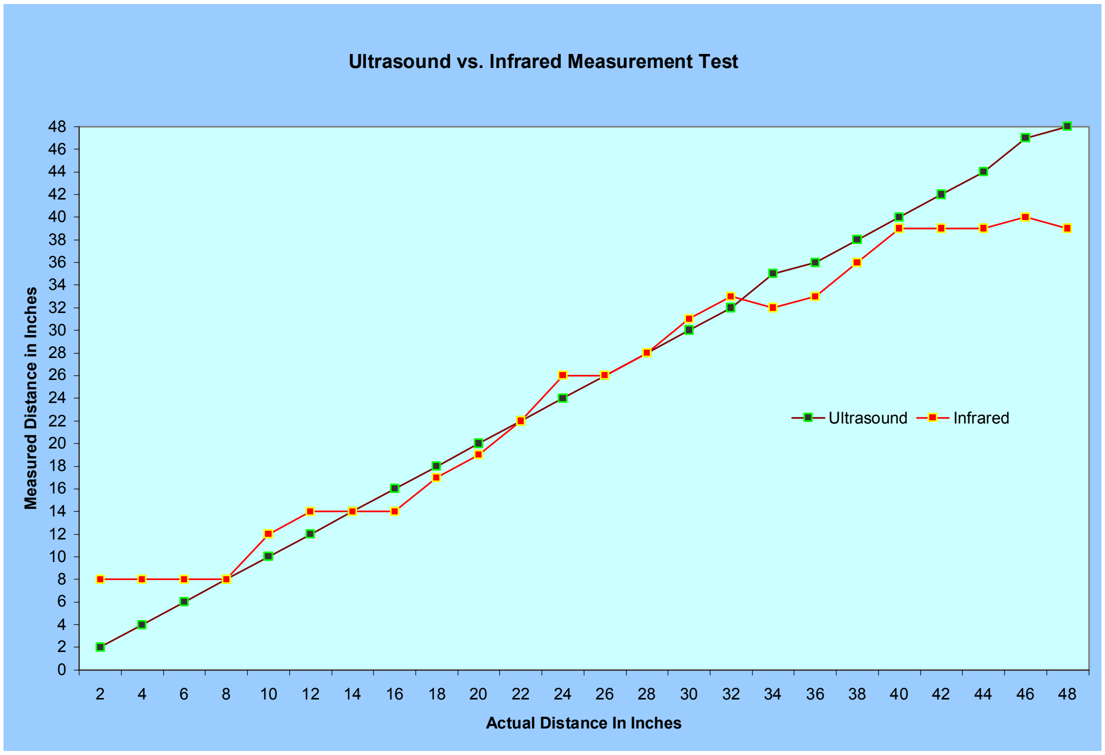


Figure 28. Ultrasound vs. Infrared Measurement Test Plot



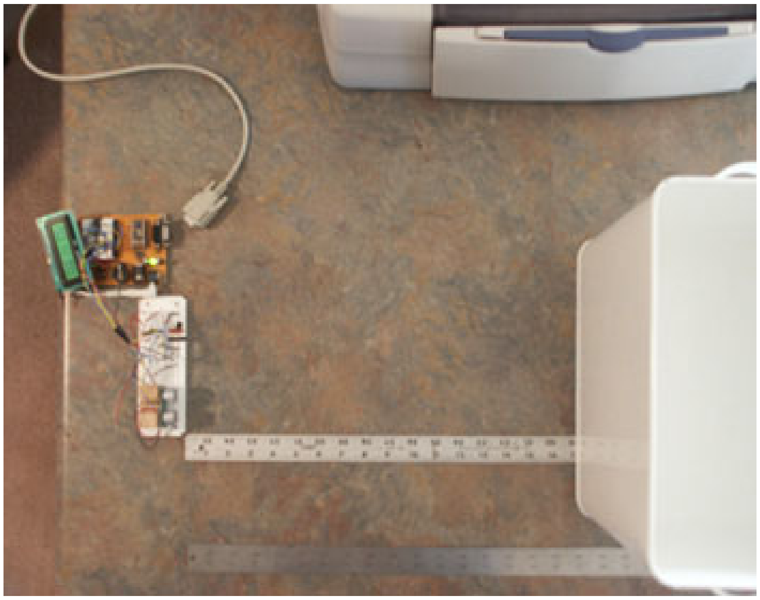


Figure 31. Measuring the distance to the standard material (breadbox).

7. Procedure

For the experiments, I had to determine a standard material and a standard distance.

Standard Material

For the calibration and interference test, I needed a material that would reflect both ultrasound and infrared the best. The standard material should be very solid, white, shiny, flat, and smooth. I chose a white breadbox that has all of these characteristics.

Standard Distance

For the distance measurements of the different materials, I needed to pick a standard distance for the distance measurements of the different materials. From the Ultrasound vs. Infrared Measurement Test Plot, I determined that 28 inches was the part where the IR and US measurements were the same for the longest. Therefore, that’s where the standard distance was.

Procedure For Different Material Tests

1. Place material at the standard distance of 28 inches in front of the sensor apparatus.
2. Press the reset button on the Stamp board.
3. Wait for the measurements to complete.
4. Write down the Infrared and Ultrasonic measurements displayed on the LCD.
5. Repeat step 1-4 for each material.

Procedure for Interference Tests

1. Place standard material (breadbox) at standard distance (28 inches).
2. Start the interference.
3. Press the reset button.
4. Wait for the measurements to complete.
5. Write down the Infrared and ultrasonic measurements displayed on the LCD.
6. Turn off interference.
7. Repeat step 1-6 for each material.

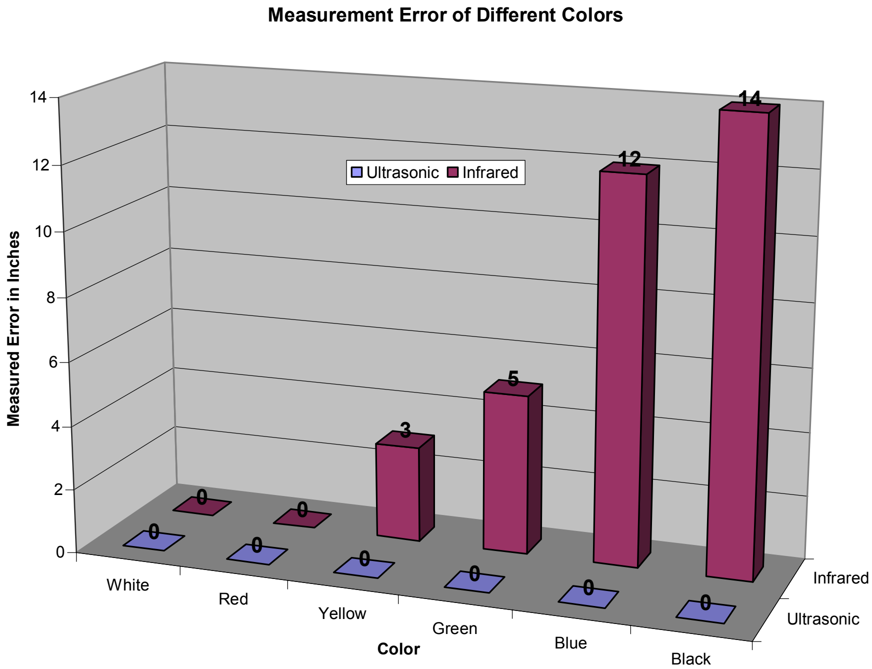
If the Infrared measurement was too far, then I recorded it at an error of 24 inches.

8. Results

Figure 32. The final results in inches showing each material and the measurements

Wall Material	**Actual Distance Inches**	**IR Measurement Inches**	**IR Error Inches**	**Ultrasonic Measurement Inches**	**Ultrasonic Error Inches**
White cardboard	28	28	0	28	0
Black cardboard	28	42	14	28	0
Yellow cardboard	28	31	3	28	0
Red cardboard	28	28	0	28	0
Blue cardboard	28	40	12	28	0
Green cardboard	28	33	5	28	0
White Wood	28	27	1	28	0
Off-White Plasterboard	28	32	4	28	0
Black conductive foam	28	Too Far	24	50	22
Light Maple wood	28	29	1	28	0
Green-painted wood	28	34	6	28	0
Dark oak	28	28	0	28	0
Brick	28	42	14	28	0
Styrofoam	28	42	14	28	0
Tan Rug	28	31	3	29	1
See-through Plastic	28	40	12	29	1
Green-kitchen sponge	28	42	14	41	13
Window Screen	28	40	12	28	0
Clear Glass	28	Too Far	24	28	0
Tin Cylinder White Waste Can	28	44	16	29	1
White Volleyball	28	39	11	29	1
Brown Basketball	28	42	14	28	0
Black Cotton T-shirt	28	Too Far	24	30	2
$20 bill	28	46	18	30	2

Figures 33 and 34. These are graphs showing measurement errors for different colors and different materials



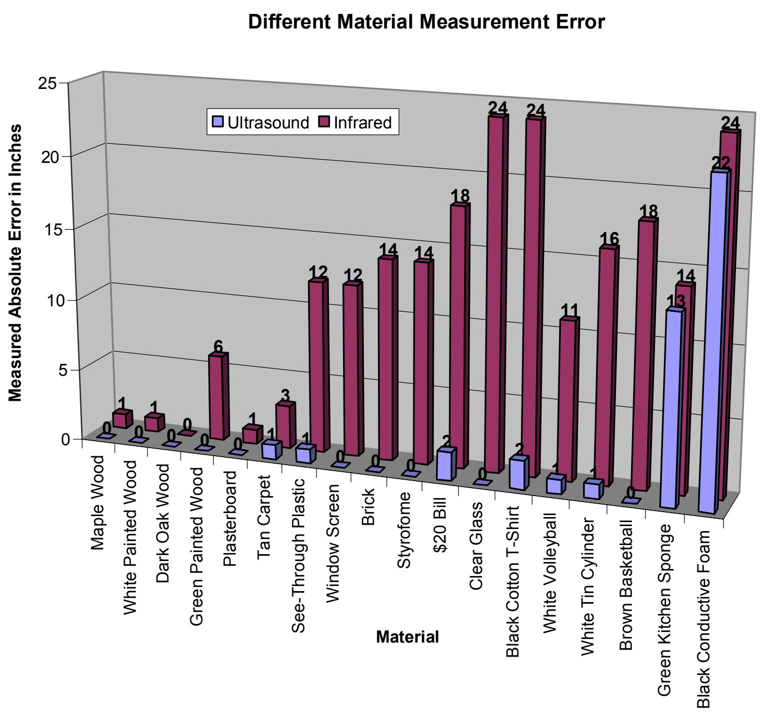


Figure 35. The distance measurement results in inches to a white bread box at 28 inches under different types of interference.

Interference	**Actual Distance Inches**	**IR Measurement Inches**	**IR Error Inches**	**Ultrasonic Measurement Inches**	**Ultrasonic Error Inches**
Halogen Light	28	36	8	28	0
Fluorescent Light	28	28	0	28	0
Flashlight	28	Too Far	24	28	0
Dog Whistle	28	28	0	28	0
Rock Music	28	28	0	28	0

9. Data Interpretation

Different colors
Ultrasound had no problem measuring the distance with the different color materials, yet the infrared did. The color of the material is what color is being reflected. Infrared had no trouble measuring the colors red and white (white means that the material is reflecting all the colors). Black doesn't reflect any colors. Therefore, infrared couldn't measure the distance to the black cardboard at all. The other colors, yellow, green, and blue, will reflect infrared, but at a lower level. The closer the color is to red, the better it will reflect it. However, the infrared is really only accurate with red and white.

Different Wall Materials
The first wall material is the wood, which includes Maple, White Painted, Dark Oak, and Green Painted. The wood is a solid material and that is what is needed for the ultrasonic. They all had small error except the green painted wood, which did worse on the IR's part. This may be because of the fact that green has no red in it at all. I was surprised that the Dark Oak worked so well. This may be because, although it's dark, it may still reflect red. The others worked well and this may be because their color is light.

Next we have plasterboard. This did very well, which may be because it is light colored. Sometimes carpet may be used as a baseboard, so we measured it and I'm surprised that the carpet did as good as it did with the Ultrasonic. It messed up with the Infrared because of the rough texture. Then there are brick walls. The brick, although it is red, has a large error under infrared. This may be because of the rough texture.

See-through Materials

Under the description of see-through materials fits See-Through Plastic, Window Screen, and Clear Glass. These got high Infrared error because the light passes completely through the glass and there is only a little light reflecting through the plastic and the screen. The Ultrasound only had a little bit of error on these because of the fact that the materials are actually there so, sound reflects off of them.

Rounded Objects

The rounded objects include a white volleyball, a white tin cylinder, and a brown basketball. Throughout all of these the Ultrasonic did well. In infrared, the volleyball got an error of 11 because of its round, yet smooth, surface. The cylinder did worse on the infrared because it has some indentations on it. The basketball did worse than the other objects in this section because there are bumps all over the spherical ball.

Sound-Absorbing Materials

The two objects in this section are a green kitchen sponge and black conductive foam. The sponge got an error of 13 in ultrasound, and 14 in infrared. These may be because of the fact that this object has a rough surface and absorbs sound. The black conductive foam got a 22 in ultrasound and a 24 in infrared. These may be because it absorbs sound and had a rough, black surface.

Misc.

Under this section is Styrofoam, a Black Cotton T-shirt, and a $20 Bill. The Styrofoam was expected to absorb sound and it didn't, and it did badly in infrared probably because of its rough texture. The T-shirt had an error of 2 in ultrasound probably because it's fabric. And for infrared the T-shirt got an error of 24 probably because of the fact that it is black. The $20 bill got an ultrasonic error of 2, because it had to stand on its own so it formed an angular pose, and an infrared error of 18 because it was green and was being reflected in different directions.

Interferences

The infrared was sensitive to some of the light sources I used as interferences. I was surprised when I shined the flashlight directly in the receiver that it didn't work at all. I was also surprised that the ultrasound worked fine with the dog whistle and the rock music (the CD I used is: The Lonely Position Of Neutral" by Trust Company). The Devantech must be designed in such a way that its not sensitive to other sound interference.

10. Conclusion

The infrared sensor did not detect the material as well as the ultrasound did in any of the measurements. My hypothesis is correct because I thought that the infrared would have trouble reflecting off of black and other surfaces. The infrared had trouble with all of the colors except white and red, rough surfaces, see-through materials, and rounded surfaces. Ultrasound only had trouble with sound absorbent materials such as the sponge and the conductive foam. Overall, Ultrasound is much more reliable than infrared for measuring distances.

As for the application to society, I am surprised that many robots use infrared instead of ultrasound for a wall sensor. You can adjust the infrared to sense white walls to work correctly, but if the robot goes into another room that is painted green, the robot will have some trouble. Also, infrared will have trouble detecting a black shoe. As long as the material is solid, ultrasound appears to have little if any error. I would think that the reason that people buy infrared more than ultrasound is because it takes up less space and costs less then the ultrasonic sensor. I do not agree with this. I think the choice should be determined by the quality and not the cost.

11. Possible Sources Of Error

I noticed after I took my measurements that the infrared was not always calibrated. That might be due to the change of background sunlight coming into the room. The day I took the calibration, it was a cloudy and rainy day and there was very little sunlight coming into the room. From the interference experiment, I know that the calibration can be affected by other sources of light.

The infrared seemed to be affected by the angle of the material it was sensing. On my board, both the IR Receiver and Transmitter are pointed upwards a little bit. Sometimes I had angled the test material so that it would reflect back to the IR receiver.

When I was testing the different colors, I used book covers so that they could stand up on their own. I hope the reflections were caused by the different colors, not the different textures.

12. Other Possible Experiments

The infrared only had a certain range where it would work. So it couldn't be too close or too far. The short distances less then eight inches could be a problem. Another possible experiment would be to switch in different values of resistors from the Stamp so that I can measure shorter distances with better accuracy. If the resistor is bigger, then the LED will not be as bright, and you should be able to measure closer distances.

The other thing I wasn't happy with was the centimeter reading for the ultrasonic. The conversion factor had a fraction of .5 which I ignored because the Stamp can only divide by an integer. I should look into someway of dividing using fractions with the stamp or use the lookup command.

My breadboards are very fragile and some of the wires kept coming out and it wouldn't work correctly. If I had made a printed circuit board I could've moved it around and tested other different objects in other situations.

13. What I have Learned

Ultrasonic is very much more dependable than Infrared. I have learned more about sound and light waves and how they work. I have learned that light waves are sensitive to the color reflected by the materials. I have gotten more experience in programming. I have learned how to build an apparatus using a microprocessor to control and interpret measurements. I learned the difference between accuracy and resolution in building my measurement apparatus. Sometimes people decide to use the cheapest solution instead of the more dependable solution.

14. Literature Cited

[1] J.W.S. Rayleigh, *The Theory Of Sound*, New York; Dover Publications Inc. 1945

[2] ."Parallax," "Basic Stamp Application Notes 12: Sonar Range Finding",
November 11, 2002

[3] "The theory of Light" February 8, 2003, http://homepages.ihug.com.au/~flavios/light&optic.htm

[4] "The Genius Of Maxwell" February 8, 2003
http://physics.hallym.ac.kr/reference/physicist/maxwell/maxnew-2.html

[5] James Clerk Maxwell, *A Treatise On Electricity and Magnetism*. New York; Dover Publications Inc. 1954

[6] Peter H. Lindsay and Donald A. Norman, *Human Information Processing,* New York; Academic Press Inc. 1972

[7] Hugh D. Young, *Fundamentals of Optics and Modern Physics*. New York ; McGraw-Hill Book Company 1968

[8] "Electromagnetic Spectrum" January 27, 2003
http://www.lbl.gov/MicroWorlds/ALSTool/EMSpec/EMSpec2.html

[9] "Parallax," "Stamp Weekend Application Kit, Infrared Emitting Diode and 40KHz Infrared Detector", November 11, 2002,
http://www.parallaxinc.com/downloads/documentation/BasicStamp1Appnotesv1.9.PDF

[10] "Parallax" "Parallax homepage" February 8, 2003, http://www.parallaxinc.com

[11] "Parallax," "Devantech SRF04 Ultrasonic Rangefinder", September 2002,
http://www.parallaxinc.com/downloads/documentation/Ranger/devantech%20with%20BASIC%20stamp.PDF

[12] Texas Instruments, *NE555, SA555, SE555 Precision Timers*, Dallas Texas, Texas Instruments Inc. 2002

[13] "Parallax," "Basic Stamp Application Notes 11: Infrared Communication",
November 11, 2002,
http://www.parallaxinc.com/downloads/documentation/BasicStamp1Appnotesv1.9.PDF

[14] Scott Edwards, *The Nuts & Volts of Basic Stamps Volume 1, Need Analog Output From The Stamp? Dial It In With a digital Potentiometer*, Rocklin California, Parallax Inc. 1999

[15] Myke Predko & Ben Wirz *TAB Electronics Build Your Own Robot Kit.* New York, McGraw-Hill 2002

[16] "Parallax," "Basic Stamp Programming Manual version 2.01", November 11, 2002
http://www.parallaxinc.com/downloads/documentation/Basic%20manual%20v2.0.PDF