Interactive Sensor Package Unit

Veselin Dimitrov
Amanda Irish
Chad Jones
Sean Kirk
Joseph Thomas

See next page for additional authors

Follow this and additional works at: http://opus.ipfw.edu/etcs_seniorproj_engineering

Opus Citation

This Senior Design Project is brought to you for free and open access by the School of Engineering, Technology and Computer Science Design Projects at Opus: Research & Creativity at IPFW. It has been accepted for inclusion in Engineering Senior Design Projects by an authorized administrator of Opus: Research & Creativity at IPFW. For more information, please contact admin@lib.ipfw.edu.
Indiana University-Purdue University Fort Wayne
Department of Engineering

ENGR 410 – ENGR 411
Capstone Senior Design Project
Report #2

Project Title: Interactive Sensor Package Unit

Team Members: Mr. Veselin Dimitrov
Ms. Amanda Irish
Mr. Chad Jones
Mr. Sean Kirk
Mr. Joseph Thomas
Mr. David Welter

Faculty Advisors: Dr. Yanfei Liu
Dr. Jiaxin Zhao

Date: April 24, 2007
CONTENTS

ACKNOWLEDGEMENTS V
ABSTRACT VI
SECTION 1: SELECTED DESIGN 1
Section 1.1: Mechanical Design 1
Section 1.1.1: Exploded View 2
Section 1.1.2: Shell Design 4
Section 1.1.3: Motor 7
Section 1.1.4: Encasing for Electrical Components 11
Section 1.1.5: Stress Analysis 14
Section 1.1.6: Mobility Analysis 19
Section 1.2: Electrical Hardware Design 23
Section 1.2.1: LED Circuits 23
Section 1.2.2: Motor Control Circuit 25
Section 1.2.3: Accelerometer Circuits 27
Section 1.2.4: Microphone Circuit 28
Section 1.2.5: Digital IR Remote Control Circuits 29
Section 1.2.6: Speaker circuit 32
Section 1.2.7: Microcontroller 33
Section 1.2.8: PIC16F737 Pin Purposes 34
Section 1.3: Software Design 34
Section 1.3.1: Description of Selected Coding Language, Compiler, and In-Circuit Debugger (ICD) 35
Section 1.3.2: Software Tasks List 37
Section 1.3.3: High-Level Algorithm Flowchart 38
Section 1.3.4: Lower-Level Algorithms Flowcharts 38

SECTION 2: BUILDING PROCESSES AND MODIFICATIONS 45
Section 2.1: Building Process 45
Section 2.2: Mechanical Modifications 46
Section 2.2.1: Replacement of the 0.125 inch Dowel Pins & Clips 46
Section 2.2.2: Revised Motor Couplers 47
Section 2.2.3: Replaced Solid Access Panels with Clear Access Panels 47
Section 2.2.4: Installation of O-ring Seal on Bolts 48
Section 2.2.5: Counter bore Diameter & Depth for Threaded Inserts 49
Section 2.3: Electrical Hardware Modifications 53
Section 2.3.1: Addition of Reed Switch 53
Section 2.3.2: Accelerometer Gain and Input Impedance Increased 54
Section 2.3.3: Microphone Gain Increased 55
Section 2.3.4: Remote Control Receiver Redesign 55
Section 2.3.5: Capacitive divider changed to Resistive divider 57
Section 2.3.6: Addition of filtering capacitors 58
Section 2.3.7: Microcontroller filter capacitors and MCLR resistor 59
Section 2.3.8: Addition of 560pF capacitor to H-bridge 59
Section 2.3.9: Change of 741 op-amp to a LM324 quad op-amp 60
Section 2.3.10: Remote control crystal oscillator replacement 61
Section 2.3.11: Jump Wires 61
Section 2.4: Software Modifications 62
Section 2.4.1: Coding Language 62
Section 2.4.2: Software Tasks List and Algorithms Methods 62
Section 2.5: PCB Design 63
Section 2.5.1: Layout and size optimization 63
Section 2.5.2. Trace wiring 64
Section 2.5.3. Footprint issues 64

SECTION 3: TESTING AND RESULTS, MEASURED PARAMETERS 65
Section 3.1: Mechanical 65
Section 3.1.1: Performance Parameters 65
Section 3.1.2: Testing Methods 65
Section 3.1.3: Durability Parameters 65
Section 3.2: Testing Procedures and Results 66
Section 3.2.1: Mobility Test 66
Section 3.2.2: Pin Tear Out 66
Section 3.2.3: Water Resistance Test 70
Section 3.2.4: PCB Strain Testing 71
Section 3.2.5 Accelerometers 72
Section 3.2.6 Microphone 73
Section 3.2.7 IR Remote – Transmitter and Receiver 73
Section 3.2.8 Motor PWM 77
Section 3.2.9: Testing of A/D Conversions 79
# Table of Figures and Tables

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>Exploded view of individual components for the unit.</td>
<td>2</td>
</tr>
<tr>
<td>Figure 2</td>
<td>Exploded View of Internal Components</td>
<td>3</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Detailed End Cap Drawing</td>
<td>4</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Illustration of End Cap</td>
<td>5</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Detailed Print of Shell</td>
<td>6</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Shell Assembly</td>
<td>7</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Motor MS-25030-370</td>
<td>7</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Detailed Drawing of Motor</td>
<td>8</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Detailed Drawing of Clamp</td>
<td>8</td>
</tr>
<tr>
<td>Figure 10</td>
<td>3 Dimensional Drawing of Clamp</td>
<td>9</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Detailed Drawing of Motor Mount</td>
<td>9</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Illustration of Motor Mount</td>
<td>10</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Detailed Drawing of Coupler</td>
<td>10</td>
</tr>
<tr>
<td>Figure 14</td>
<td>3 Dimensional Drawing of Coupler</td>
<td>11</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Detailed Drawing of Mounting Tray</td>
<td>11</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Illustration of Mounting Tray</td>
<td>12</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Detailed Drawing of Battery Tray</td>
<td>12</td>
</tr>
<tr>
<td>Figure 18</td>
<td>3 Dimensional Drawing of Battery Tray</td>
<td>13</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Detailed Drawing of Linkage</td>
<td>13</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Illustration of Linkage</td>
<td>14</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Cross Section of Shaft</td>
<td>15</td>
</tr>
<tr>
<td>Figure 22</td>
<td>S-N Diagram to determine lifetime of rotating shaft.</td>
<td>16</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Illustration of Shear Areas</td>
<td>17</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Torque and Force of Friction on the shell</td>
<td>20</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Free Body Diagram Used in Back Torque Analysis</td>
<td>21</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Circuit Diagram for LEDs</td>
<td>24</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Motor Control Circuit</td>
<td>25</td>
</tr>
<tr>
<td>Figure 28</td>
<td>Circuit Diagram For Accelerometers</td>
<td>28</td>
</tr>
<tr>
<td>Figure 29</td>
<td>Circuit Diagram for Microphone</td>
<td>29</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Circuit Diagram for Digital IR Remote Control Handheld Unit</td>
<td>30</td>
</tr>
<tr>
<td>Figure 31</td>
<td>Circuit Diagram for Digital IR Remote Control Receiver Unit</td>
<td>31</td>
</tr>
<tr>
<td>Figure 32</td>
<td>Circuit Diagram for Speaker</td>
<td>32</td>
</tr>
<tr>
<td>Figure 33</td>
<td>PIC16F737 Microcontroller pin diagram</td>
<td>33</td>
</tr>
<tr>
<td>Figure 34</td>
<td>Use of the asm (...) directive in mikroC</td>
<td>36</td>
</tr>
<tr>
<td>Figure 35</td>
<td>Configuration Bits in MPLAB</td>
<td>37</td>
</tr>
<tr>
<td>Figure 36</td>
<td>Program Target Device Icon</td>
<td>37</td>
</tr>
<tr>
<td>Figure 37</td>
<td>Highest-level algorithm flowchart</td>
<td>38</td>
</tr>
<tr>
<td>Figure 38</td>
<td>Program Beginning</td>
<td>39</td>
</tr>
<tr>
<td>Figure 39</td>
<td>Drive Forwards</td>
<td>40</td>
</tr>
<tr>
<td>Figure 40</td>
<td>Driving Timing</td>
<td>41</td>
</tr>
<tr>
<td>Figure 41</td>
<td>Stop After Forwards</td>
<td>42</td>
</tr>
<tr>
<td>Figure 42</td>
<td>Stop Timing</td>
<td>43</td>
</tr>
<tr>
<td>Figure 43</td>
<td>Drive Backwards</td>
<td>44</td>
</tr>
<tr>
<td>Figure 44</td>
<td>Stop After Backwards</td>
<td>45</td>
</tr>
<tr>
<td>Figure 45</td>
<td>Exploded view of assembly</td>
<td>46</td>
</tr>
<tr>
<td>Figure 46</td>
<td>Dowel Pin &amp; Clip With Replacement 6-32 SHCS &amp; Nyloc Nut</td>
<td>47</td>
</tr>
<tr>
<td>Figure 47</td>
<td>Initial &amp; Revised Access Panels</td>
<td>48</td>
</tr>
<tr>
<td>Figure 48</td>
<td>O-rings Installed on 8-32 SHCS</td>
<td>49</td>
</tr>
<tr>
<td>Figure 49</td>
<td>Revised Print for ISPU Shell</td>
<td>50</td>
</tr>
<tr>
<td>Figure 50</td>
<td>Revised Access Panel Print</td>
<td>51</td>
</tr>
<tr>
<td>Figure 51</td>
<td>Revised Motor Coupler Print</td>
<td>52</td>
</tr>
<tr>
<td>Figure 52</td>
<td>Circuit diagram of reed switch (S3) added to input of voltage regulator</td>
<td>53</td>
</tr>
<tr>
<td>Figure 53</td>
<td>Installation of Reed Switch on the bottom of the battery tray</td>
<td>53</td>
</tr>
<tr>
<td>Figure 54</td>
<td>Redesigned accelerometer circuit</td>
<td>55</td>
</tr>
<tr>
<td>Figure 55</td>
<td>Load Line</td>
<td>57</td>
</tr>
<tr>
<td>Figure 56</td>
<td>Remote Control circuits Pre-Amp</td>
<td>57</td>
</tr>
<tr>
<td>Figure 57</td>
<td>Change from Capacitive to Resistive divider circuit</td>
<td>58</td>
</tr>
<tr>
<td>Figure 58</td>
<td>Filter capacitors and MCLR resistor</td>
<td>59</td>
</tr>
<tr>
<td>Figure 59</td>
<td>Comparison of 741 Op-amp and LM324 quad op-amp sizes</td>
<td>60</td>
</tr>
<tr>
<td>Figure 60</td>
<td>Remote control circuit</td>
<td>61</td>
</tr>
<tr>
<td>Figure 61</td>
<td>Tinius Olsen machine used for the tear out testing</td>
<td>67</td>
</tr>
</tbody>
</table>

---

iii
Figure 62: Test apparatus for tear out testing
Figure 63: Assembled apparatus for tear out testing  68
Figure 64: Test setup of tear out test  68
Figure 65: Link after failure  69
Figure 66: Link after tear out  69
Figure 67: Water resistance testing  70
Figure 68: Test setup for PCB torsion testing  71
Figure 69: Red arrow shows axis of acceleration for testing. Yellow lines show accelerometer placement  72
Figure 70: Op Amp Output of Accelerometers Feeding the Microcontroller  72
Figure 71: Output of microphone (yellow trace) and op amp circuit (blue trace)  73
Figure 72: Output of the IR transmitter (yellow) and receiver transistor collector (blue), 8 kHz signal  74
Figure 73: IR voltage seen before (yellow) and after (blue) the operational amplifier, 8 kHz signal  75
Figure 74: IR remote transmitter (yellow) and op amp output (blue), 8 kHz signal at ~1 ft.  76
Figure 75: IR remote transmitter (yellow) and op amp output (blue), 8 kHz signal at 4 feet.  77
Figure 76: IR remote transmitter (yellow) and op amp output (blue), 16 kHz signal at 4 feet.  77
Figure 77: Motor PWM signal with 10 kΩ resistor used as the load  78
Figure 78: Motor PWM signal with motors as the load. The blue line represents the voltage across the motors; the yellow line is the output of the microcontroller.  79
Figure 79: Robust battery pack suggested for  82

Table 1: Nomenclature Used in Mechanical Design and Analysis ............................... 1
Table 2: Pin Purposes................................. 34
Table 3: Times and distances recorded for mobility testing ................................. 66
Acknowledgements

The ISPU Senior Design team wishes to acknowledge the continued support and input of their advisors, Dr. Yanfei Liu and Dr. Jiaxin Zhao. Without their knowledge and guidance, this project would not have proceeded in the straightforward way it has. Their insights have pushed us to improve our reasoning skills, and back up each assertion with sound engineering reasoning.

Additionally, we would like to acknowledge Dr. Hal Broberg for his knowledge of the PIC family of microcontrollers, and Peter Goodman for his assistance in electrical component selection and PCB layout. Their insights were vital to our success.

We would also like to thank our sponsor, Practical Robotic Innovations, Inc. (PRI), for their financial and technological assistance. In particular, Scott Maddox, PRI President, was personally responsible for bringing the ISPU project to the attention of IPFW administration, and has remained an active participant throughout the design phase of the project. His technical and business experience has been invaluable, and this project could not have happened without his support.

Special thanks to Jason Davis and John Mitchell who both helped with the manufacturing of the unit’s shell as well as correcting various components and machining necessary tooling for our testing. Without the attention required in manufacturing the shell throughout its 30 hour build, the shell may have not turned out as good as it did.

Additional thanks goes to Kronmiller Machine and Tool. They were able to supply and machine the clear access panels, the docking station for the unit, the o-rings for the 8-32 SHCS on the access panels, and the linkages that connects the mounting tray to the battery tray.
Abstract

Members of the family come in all shapes and sizes. But what they all have in common is the need for interaction. Unfortunately, no product exists which demonstrates the ability to actively engage an individual family member. It is the goal of the Interactive Sensor Package Unit (ISPU) project team to create a product platform for devices that will engage these family members in play while they are left at home unsupervised. To achieve this, the ISPU receives inputs from a microphone and accelerometers, and then adjusts sound, light, and motion output based on algorithms programmed into a microcontroller.
Section 1: Selected Design

Section 1.1: Mechanical Design

Variable Classification

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Correction Factor</td>
</tr>
<tr>
<td>CLOAD</td>
<td>Load Constant</td>
</tr>
<tr>
<td>CRELIABILITY</td>
<td>Reliability Constant</td>
</tr>
<tr>
<td>CSIZE</td>
<td>Size Constant</td>
</tr>
<tr>
<td>CSURFACE</td>
<td>Surface Constant</td>
</tr>
<tr>
<td>CTEMP</td>
<td>Temperature Constant</td>
</tr>
<tr>
<td>dt</td>
<td>Change In Time</td>
</tr>
<tr>
<td>dv</td>
<td>Change In Velocity</td>
</tr>
<tr>
<td>dx</td>
<td>Change In Displacement</td>
</tr>
<tr>
<td>F</td>
<td>Force</td>
</tr>
<tr>
<td>FDP</td>
<td>Force On Dowel Pins</td>
</tr>
<tr>
<td>FIMP</td>
<td>Impulse Force</td>
</tr>
<tr>
<td>g</td>
<td>Gravity</td>
</tr>
<tr>
<td>G</td>
<td>Modulus Of Rigidity</td>
</tr>
<tr>
<td>h</td>
<td>Height</td>
</tr>
<tr>
<td>I</td>
<td>Moment Of Inertia</td>
</tr>
<tr>
<td>J</td>
<td>Polar Moment Of Inertia</td>
</tr>
<tr>
<td>Je</td>
<td>Effective Polar Moment Of Inertia</td>
</tr>
<tr>
<td>EK</td>
<td>Kinetic Energy</td>
</tr>
<tr>
<td>L</td>
<td>Length</td>
</tr>
<tr>
<td>M</td>
<td>Equivalent Moment</td>
</tr>
<tr>
<td>m</td>
<td>Mass</td>
</tr>
<tr>
<td>P</td>
<td>Equivalent Concentrated Force</td>
</tr>
<tr>
<td>EP</td>
<td>Potential Energy</td>
</tr>
<tr>
<td>r</td>
<td>Radius</td>
</tr>
<tr>
<td>Se</td>
<td>Effective Alternating Stress</td>
</tr>
<tr>
<td>Se'</td>
<td>Modified Effective Alternating Stress</td>
</tr>
<tr>
<td>SM</td>
<td>Mean Stress In Fatigue Cycle</td>
</tr>
<tr>
<td>τ</td>
<td>Shear Stress</td>
</tr>
<tr>
<td>t</td>
<td>Thickness</td>
</tr>
<tr>
<td>T</td>
<td>Torque</td>
</tr>
<tr>
<td>tf</td>
<td>Final Time</td>
</tr>
<tr>
<td>ti</td>
<td>Initial Time</td>
</tr>
<tr>
<td>Tm</td>
<td>Torque Of Motor</td>
</tr>
<tr>
<td>v</td>
<td>Velocity</td>
</tr>
<tr>
<td>vf</td>
<td>Final Velocity</td>
</tr>
<tr>
<td>vi</td>
<td>Initial Velocity</td>
</tr>
<tr>
<td>w</td>
<td>Width</td>
</tr>
<tr>
<td>w'</td>
<td>Width Of Edge Of Dowel Hole To Edge Of Linkage</td>
</tr>
<tr>
<td>xf</td>
<td>Final Displacement</td>
</tr>
<tr>
<td>θ</td>
<td>Angle Of Twist</td>
</tr>
<tr>
<td>σ1</td>
<td>Principle Stress 1</td>
</tr>
<tr>
<td>σ2</td>
<td>Principle Stress 2</td>
</tr>
<tr>
<td>σa</td>
<td>Alternating Stress</td>
</tr>
<tr>
<td>σu</td>
<td>Ultimate Stress</td>
</tr>
<tr>
<td>σt</td>
<td>Axial Stress</td>
</tr>
<tr>
<td>σys</td>
<td>Yield Stress</td>
</tr>
</tbody>
</table>

Table 1: Nomenclature Used in Mechanical Design and Analysis
Section 1.1.1: Exploded View

Figure 1: Exploded view of individual components for the unit.
Figure 2: Exploded View of Internal Components
Section 1.1.2: Shell Design

Section 1.1.2.1: End Caps
The ISPU Prototype utilizes end caps that were designed such that they are held in place by eight 8-32 socket head cap screws, as seen in Figure 3. This is due to the need of the team to have accessibility to the internal electronics. The finished model will utilize rivets so that the consumer cannot access the internal electronics. There are also four 0.123” diameter holes located at the center of the access panel for the 0.125” steel dowel pins to be pressed into for coupling of the internal components.

Figure 3: Detailed End Cap Drawing
Section 1.1.2.2: Mid Shell
The mid shell is the main housing of the ISPU and protection of the electronics. It also serves as the medium as the transference of rotational torque generated by the motors that result in the unit’s mobility. The shell has eight holes to accommodate eight steel press fit 8-32 threaded inserts. These threaded inserts allow for the access panels to be removed multiple times without wearing out the shell itself. There is also an o-ring groove to hinder ambient moisture/water from reaching the units electronics.

Figure 4: Illustration of End Cap
**Section 1.1.3: Motor**
The DC motor chosen was for the ISPU were Mabuchi RF-370 that had a gear ratio of 30:1 as shown in Figure 7. From a dynamic analysis (see section 4.3.4.5) it was determined that a torque of 3.75 in-lb per motor would be needed to provided satisfactory acceleration performance and a speed of 150 RPM’s. The Mabuchi motor supplies 9 in-lb of no-load torque and has a no load speed of 300 RPM’s. From the team’s analysis, these motors will be capable of the performance required by the design specifications. Figure 8 shows detailed physical information of the RF-370 motor, such as housing diameter & length, bolt mounting and shaft diameter and length.
Section 1.1.3.1: Clamp

It was decided that since the motor (and its shafts) support the entire mass of the internal components, that the two 4-40 threaded holes (on the front of the motor) would not provide enough strength alone. Therefore, a rear clamp was designed (Figure 9) that could be attached to the rear of the motor and bolted to the motor mount, that would support the rear of the motor and reduce the chance for any unwanted deflection. In modeling the motors and other components in the CAD system, it became clear that there would not be enough space between the motors for the PCB and other electronics. When designing the rear clamp (see Figure 10), the top was made flat with two 4-40 threaded holes, providing a place for the PCB & electronics or, a way of connecting the motors to increase the overall strength of the mounting tray.
Section 1.1.3.2: Mount

The motors are connected rigidly to the mounting tray by the motor mount shown in. The mount is designed to be used in conjunction with the rear clamp and is then connected to the mounting tray with eight 6-32 socket head cap screws. The motor mount is made from 1020 HRS sheet metal that is 0.060” thick as detailed in Figure 11.
Section 1.1.3.3: Coupler
The coupler (shown in Figure 13) is what connects the motors torque to the ISPU’s shell to provide translation. The coupler is connected to the motors shaft by a 6-32 socket head set screw that connects the coupler rigidly to the shafts motor. Power is then transferred to the shell by four 0.125” steel dowel pins that are pressed into the access panels and slip through the couplers four 0.127” holes. These holes have tolerances (see Figure 13) such that a minimal gap will exist between the coupler and dowel pins when assembled.
Section 1.1.4: Encasing for Electrical Components

Section 1.1.4.1: Mounting Tray
The mounting tray is the back bone to the ISPU’s internal structure. It is made from 1020 HRS sheet metal and is 0.060” thick and is formed as shown in Figure 15. It has holes for mounting the motor assemblies and for the connecting links that connect the battery tray hanging below. It has been designed such that the PCB board can be mounted to the underside for protection from the heat generated by the motors.
Section 1.1.4.2: Battery Tray
The battery tray is constructed utilizing the same material as the mounting tray and is formed in
the same manner as shown in Figure 17. It provides two purposes; the first is that it provides a
place to store the battery for the ISPU. The second is that it also provides a counter weight that
resists the motors torque that tries to revolve the internal assembly about the motors shafts,
resulting in no translational motion of the unit.

Figure 16: Illustration of Mounting Tray

Figure 17: Detailed Drawing of Battery Tray
Section 1.1.4.2.1: Linkages
The connecting link is a piece of steel that is used to suspend the IPSU’s battery pack below the mounting tray. It is made from 1020 HRS sheet metal that is 0.060” thick as shown in Figure 19. The dimensions shown were determined with packaging constraints in mind as well as a stress analysis that is included in section 4.3.5.3.

NOTES:
1. MATERIAL 1020 HRS 0.06 THICK
2. TWO PLACE DECIMAL 0.01
3. THREE PLACE DECIMAL 0.005
Section 1.1.5: Stress Analysis

Section 1.1.5.1: Motor Shaft

Due to extended use of the motor, fatigue analysis was performed to ensure the motor shafts would not fail due to enhanced revolutions of the shaft. Finding the modified effective alternating stress, $S_e'$, and then comparing it to the alternating stress is what the fatigue analysis consisted of. If this value is below the effective alternating stress the theoretical lifetime of the shaft is infinite. This comparison is visualized in Figure 22. The red line is an indication of where a rotating shaft would fail when subjected to various alternating stresses for this shaft’s particular material which was assumed to be 4140 annealed steel as the material was not given on the data sheet and the distributor did not have that information.

The modified endurance limit was found by multiplying the ultimate stress for the material by the coefficient for torsion resulting in following equation.

$$S_e' = 0.5 \cdot \sigma = 47.49 \text{ ksi}$$

The effective alternating stress can then be found by multiplying the modified endurance limit by the correction factors for load, size, surface, temperature, and reliability. The loading, size, and temperature correction factors are 1 as the load is due to bending, the size being less than 0.3 inches, and the temperature being less than 450º F, respectively. The remaining correction factors are found from the equations following.

$$S_e = C_{LOAD} \cdot C_{SIZE} \cdot C_{SURFACE} \cdot C_{TEMP} \cdot C_{RELIABILITY} \cdot S_e'$$
\[ Se = 1.1 \cdot 0.5058 \cdot 1.897 \cdot 47490 = 21,546 \text{ psi} \]

\[ C_{\text{SURFACE}} = A \cdot \sigma_u^B = 1.34 \cdot 0.94975^{-0.085} = 0.5058 \]

(\text{ground finish therefore } A = 1.34 , B = 0.085)

\[ C_{\text{RELIABILITY}} = 0.897 \ (\text{RELIABILITY} = 90\%) \]

The mean stress in the fatigue cycle, \( S_m \), is found by taking 90\% of the ultimate strength.

\[ S_m = 0.9 \cdot 0.94975 = 85,478 \text{ psi} \]

The only thing left to find to analyze the fatigue analysis from a S-N diagram is to find the alternating stress. It is first necessary to find the area moment of inertia and the polar moment of inertia for the motor shaft, the equation for that is shown below along with equation used to find the angle of the missing segment, Figure 21 shows the related geometry for the shaft. The principle stress is the square root of the axial stress squared added to the shear stress squared. The equations for the stress in x direction, principle stresses, and alternating stresses are found below.

\[ \theta = 2 \cdot \pi - \sin^{-1} \left( \frac{b/2}{r} \right) = 2 \cdot \pi - \sin^{-1} \left( \frac{125/2}{0.0785} \right) = 4.44 \text{ rad} \]

\[ I = \frac{r^4}{8} \left( \theta - \sin(\theta) + 2 \cdot \sin(\theta) \cdot \sin^2 \left( \frac{\theta}{2} \right) \right) \]

\[ = \frac{0.0785^4}{8} \left( 4.44 - \sin(4.44) + 2 \cdot \sin(4.44) \cdot \sin^2 \left( \frac{4.44}{2} \right) \right) = 1.986 \cdot 10^{-5} \text{ in}^4 \]

\[ J = \frac{r^4}{4} \left( \theta - \sin(\theta) + \frac{2}{3} \cdot \sin(\theta) \cdot \sin^2 \left( \frac{\theta}{2} \right) \right) \]

\[ = \frac{0.0785^4}{4} \left( 4.44 - \sin(4.44) + \frac{2}{3} \cdot \sin(4.44) \cdot \sin^2 \left( \frac{4.44}{2} \right) \right) = 5.659 \cdot 10^{-5} \]

\[ \sigma_x = \frac{P \cdot L \cdot r}{I} = \frac{0.945 \cdot 2.930 \cdot 0.0785}{1.986 \cdot 10^{-5}} = 10.95 \text{ ksi} \]

Figure 21: Cross Section of Shaft
\[
\tau = \frac{T \cdot r}{J} = \frac{1.346 \cdot 0.0785}{5.659 \cdot 10^{-5}} = 1.87 \text{ ksi}
\]

\[
\begin{align*}
\sigma_1 &= \frac{\sigma_x}{2} + \sqrt{\left(\frac{\sigma_x}{2}\right)^2 + \left(\sigma\right)^2} = \frac{10.95 \cdot 10^3}{2} + \sqrt{\left(10.95 \cdot 10^3\right)^2 + \left(1.87 \cdot 10^3\right)^2} = 11.26 \text{ ksi} \\
\sigma_2 &= \frac{\sigma_x}{2} - \sqrt{\left(\frac{\sigma_x}{2}\right)^2 + \left(\sigma\right)^2} = \frac{10.95 \cdot 10^3}{2} - \sqrt{\left(10.95 \cdot 10^3\right)^2 + \left(1.87 \cdot 10^3\right)^2} = -0.31 \text{ ksi} \\
\sigma_a &= \sqrt{\left(\frac{\sigma_x}{2}\right)^2 + \left(\sigma\right)^2} = \sqrt{\left(10.95 \cdot 10^3\right)^2 + \left(1.87 \cdot 10^3\right)^2} = 5.78 \text{ ksi}
\end{align*}
\]

Figure 22: S-N Diagram to determine lifetime of rotating shaft.

While these shafts will survive normal operation they would not survive a drop from 5 feet. Due to time and monetary constraints a motor with adequate shafts is unavailable. The impact force of a drop from this distance is upwards of 200 g’s, at this great of a force, custom motors with custom shafts would have to be purchased in order to survive such a drop therefore stress analysis is not included in this report.

**Section 1.1.5.2: Coupler and Dowel Pins**
The shearing force that acted on the coupler dowel pins during a drop from 5 feet was calculated to ensure that the dowel pins would not fail in the instance that the unit was tossed to the dog from a standing position. The first step in this was to determine the impulse force of the ball hitting the ground from that height. To do this the velocity had to be determined from the conservation of momentum equation due to an impact from free fall motion. This resulted in the following equations:

\[ E_{p1} + E_{k1} = E_{p2} + E_{k2} \Rightarrow mgh + 0 = 0 + \frac{1}{2}mv^2 \]

\[ v = \sqrt{2 \cdot g \cdot h} = \sqrt{2 \cdot 386.4 \left( \frac{in}{s^2} \right) \cdot 60 \left( in \right)} = 215.33 \text{ in/s} \]

\[ F_{imp} = m \cdot \frac{dv}{dt} = \frac{5 \text{ (lbs)}}{386.4 \left( \frac{in}{s^2} \right)} \cdot \frac{215.33 \left( \frac{in}{s} \right)}{s} \cdot \frac{1}{.0025 \left( s \right)} = 1114.29 \text{ lbs} \]

This impulse force is modeled as being evenly divided amongst the 8 dowel pins. If this were the case, each dowel pin is subjected to 139.2 lbs of shearing force. The tear out stress due to the impulse force on each dowel pin is found by dividing the impulse force by the contact area of the dowel pin. This is shown in the following equation & the related areas are shown in Figure 23.

Figure 23: Illustration of Shear Areas

Left and Right Tear Out Equations
Tear Out Stress = \frac{F_{\text{IMP}}}{8} \cdot \frac{1}{2 \cdot (A_{\text{Left}} + A_{\text{Right}})} = \frac{1114.29}{8} \cdot \frac{1}{2 \cdot (0.047 + 0.015)} = 1123.276 \text{ psi}

Top and Bottom Tear Out Equations

\text{Tear Out Stress} = \frac{F_{\text{IMP}}}{8} \cdot \frac{1114.29 \text{ lbs}}{2 \cdot A_{\text{Top,Bottom}}} = \frac{1114.29}{8} \cdot \frac{1}{2 \cdot 0.033} = 2110.398 \text{ psi}

This is well below the yield strength of 29.73 ksi of the coupler indicating the coupler will not fail as there is a factor of safety of \(\approx 14\). The dowel pins are rated by the manufacturer to sustain 1600 lbs of shearing force without failing thus also indicating they will not fail either as the factor of safety is \(\approx 11.5\) in the case of shearing force.

**Section 1.1.5.3: Connecting Links**

To find the minimum thickness that could be used for the connecting linkages the tear out analysis was performed once again. The force used was determined from multiplying the mass of the internal components by the velocity when the ball hit the ground and divided by the impact time. From this calculation it was determined that the minimum thickness of the linkages was found to be 0.02294 inches with a factor of safety of 2. The thickness used was 0.060 inches for being consistent with the thickness of the same stock material for cost effectiveness. The equations used in determining these numbers are as follows:

\[ F_{\text{IMP}} = m \cdot \frac{dv}{dt} = \frac{5(\text{lbs})}{386.4 \left(\frac{\text{in}}{\text{s}^2}\right)} \cdot \frac{215.33(\text{in})}{(\text{s})} \cdot \frac{1}{0.0025(\text{s})} = 1114.29 \text{ lbs} \]

\[ t = \frac{F_{\text{imp}}}{8} \cdot \frac{1114.29 \text{ lbs}}{2 \cdot w' \cdot \sigma_{ys}} = \frac{1114.29}{8} \cdot \frac{1}{2 \cdot 0.130 \cdot 29,730} = 0.02294 \text{ in} \]

**Section 1.1.5.4: Mounting Tray and PCB**

It was necessary to determine the stress due to the maximum torque supplied from the motors to the PCB to ensure the electrical components would not be damaged from normal operation of the unit. The analysis chosen to model this situation is from the equating the angles of twist of the mounting tray and the PCB. Since both are fixed to each other, they both must have the same angle of twist due to a certain torque. From this the torque acting on each component could be determined. With the torque acting on the PCB the shear stress on the PCB could be found.
\[
\theta = \frac{T_{PCB} \cdot L_{PCB}}{G_{PCB} \cdot J_{e,PCB}} = \frac{T_{MT} \cdot L_{MT}}{G_{MT} \cdot J_{e,MT}} \Rightarrow T_{PCB} \cdot 3(in) = 1.6 \cdot 10^3 (psi) \cdot 1.953 \cdot 10^{-3} (in^4) = 1.15 \cdot 10^3 (psi) \cdot 5.992 \cdot 10^{-4} (in^4)
\]

\[
\therefore T_{PCB} = 0.04535 \cdot T_{MP}
\]

\[
T_{total} = T_{PCB} + T_{MP} = 3.75 (in \cdot lb) \Rightarrow T_{PCB} = 1.620 (in \cdot lb) \quad \& \quad T_{MP} = 3.588 (in \cdot lb)
\]

\[
\tau = \frac{T_{PCB} \cdot t_{PCB}}{J_{e,PCB}} = \frac{1.620 (in \cdot lb) \cdot .125 (in)}{1.953 \cdot 10^{-3}} = 10.37 (psi)
\]

These results show that the PCB will not fail according to the max shear stress theory as the yield strength of the PCB is 8.5 ksi.

**Section 1.1.6: Mobility Analysis**

To the motors required to meet the mobility requirements, a dynamic analysis was performed. A weight of 15 pounds was used since the ISPU must travel through 3” tall grass of nominal density and travel up a five-degree incline. The unit is expected to weigh much less, but this will allow for some of the things that we cannot account for.

**Section 1.1.6.1: Torque Analysis**

First, the time to travel an arbitrary distance was determined. A distance of 3 feet (36 inches) was used along with the required maximum velocity of 3 mph (52.8 in/s).

Selecting a desired time to reach max speed, 3 seconds, was used in calculation:

\[
t_f = 3 \text{ seconds}
\]

Next, the acceleration of the ISPU was determined:

\[
a = \frac{2 \cdot x_f}{t^2} = \frac{2 \cdot 36}{3^2}
\]

Plugging in the values for the final velocity and time gives:

\[
a = 8 \frac{in}{s^2}
\]

Now the force tangent to the units shell can be determined using a mass of 10lbs. 10 lbs was used to account for the unit having to go up an incline of 5º;
The overall torque is then determined by taking the summation of torques about the motor shafts. Figure 24 shows the free body diagram showing the forces involved in the torque analysis.

\[
\sum F = m \cdot a \\
F_f = m \cdot a \\
F_f = \left( \frac{10}{386.4} \right) \cdot 8 \\
F_f = 0.20704 \text{ lb}
\]

Since there are two motors that will provide units mobility, the torque found can be divided in half, giving the torque required per motor.

\[
\sum T = I \cdot \alpha \\
T - F_f \cdot r = m \cdot r^2 \cdot \left( \frac{a}{r} \right) \\
T = \left[ \left( \frac{2}{386.4} \right) \cdot 5^2 \right] \cdot \left( \frac{8}{5} \right) + 0.2074 \cdot 5 \\
T = 1.24 \text{ in} \cdot \text{lbs}
\]

\[T_{\text{per motor}} = \frac{T}{2}\]
\[T_{\text{per motor}} = 0.6211 \text{ in} \cdot \text{lbs}\]
\[T_{\text{per motor}} = 9.94 \text{ in} \cdot \text{oz}\]

The final velocity of the unit at the indicated time, distance, and calculated acceleration is found to be:
\[ v_f^2 = v_o^2 + 2 \cdot a \cdot (x_f - x_o) \Rightarrow v_f = 1.36 \text{ mph} \]

From our data it was found that the parameters specified will not meet the required velocity of 3 mph. In order to meet this requirement, it was found that with keeping the time of 3 seconds, the displacement would have to be lengthened to 78 inches which would then in turn give an acceleration of 17.33 in/s². The torque analysis was then recalculated giving a torque of 1.3458 in·lb, 16.15 in·oz. This is well under the maximum torque allowed for the motor which is 144 in·oz. A factor of safety is determined from this analysis to be 8.92.

**Section 1.1.6.2: Back Torque Analysis**

The back torque analysis was performed to determine the amount of “counter torque” the hanging mass (i.e. the battery pack & tray it mounts to) exerts when the motors are engaged, opposing the motor torque therefore, keeping the internal components from spinning around inside the unit and resulting in zero translational motion. For the analysis a free body diagram (FBD) was constructed as seen in Figure 25.

![Figure 25: Free Body Diagram Used in Back Torque Analysis](image)

As seen in the FBD, when the motors are engaged, the torque they generate will be inclined to rotate the mass of the internal components in the direction of the motor torque. To ensure that the hanging mass would provide adequate resisting torque to prevent this, an analysis was
performed. Since the mass of the battery pack and tray are fixed, the only variable is the length from the motor shaft center to the mass center of the battery pack and tray.

As the motor torque is increased, the angle, theta, will increase as the battery pack & tray are lifted. The back torque is found by taking the compliment of the battery pack & tray weight that acts perpendicular to the moment arm, in this case, $T_B = w \cdot \cos(\beta) \cdot l$ and $\beta = (90 - \theta)$, as shown in Figure 25.

Since the maximum back torque will occur when theta equals 90 degrees, the target angle for theta was chosen to be 45 – 50 degrees. Using this target angle allows for situations where motor torque may need to be increased because a faster acceleration is needed, translation up a steep incline or, if the unit is rolling through a denser medium such as thick grass.

From the torque analysis performed, the motor torque is known to be 2.69 in-lb and with this, the counter torque can be determined, as shown.

Summing the torques about the motor shafts & taking the clockwise direction to be positive gives:

$$T_{Motors} - T_B = 0$$

Where:

$$T_{Motors} = Motor Torque$$

$$T_B = Back Torque$$

Substituting in:

$$T_{Motors} = 2.69 \text{ in-lb}$$

$$T_B = w \cdot \cos(90 - \theta) \cdot l$$

where:

$$w = 0.89 \text{ lb (weight of battery pack & tray)}$$

$$l = \text{to be calculated}$$

Through an iterative process, the target angle for theta (within the target range) and length were determined to be;

$$\theta = 49.1^\circ$$

$$l = 4.00 \text{ inches}$$

With this length, the maximum torque that occurs when theta equals 90 degrees is:

$$T_{Max \ @ \ \theta = 90^\circ} = 3.56 \text{ in-lb}$$
This maximum torque is well above the motor torque calculated for the unit to reach its maximum velocity and will allow for higher torques if needed.

Section 1.1.6.3: Speed Analysis

It was a requirement that the unit roll at a velocity of 3 mph. To calculate the required RPMs of the motor it is simply a conversion of the required speed to the units of required for the motor selection. This equation is as follows.

\[
\frac{3 \text{ (mi)}}{\text{(hr)}} \cdot \frac{5280 \text{ (ft)}}{\text{(mi)}} \cdot \frac{12 \text{ (in)}}{\text{(ft)}} \cdot \frac{1 \text{ (rev)}}{2 \pi r} = \frac{3 \text{ (mi)}}{\text{(hr)}} \cdot \frac{5280 \text{ (ft)}}{\text{(mi)}} \cdot \frac{12 \text{ (in)}}{\text{(ft)}} \cdot \frac{1 \text{ (rev)}}{2 \pi 5} = 100.84 \text{ RPM}
\]

Section 1.2: Electrical Hardware Design

This section is comprised of all electrical hardware needed to control inputs and outputs of the microcontroller of the Equilibrium Seeking Unit.

Section 1.2.1: LED Circuits

As discussed in the design description, LEDs will be used for light stimulation. The LEDs will operate in an On/Off mode. In order to turn on and off the LEDs, transistor switches will be used. The 2N2222A transistor is chosen for the switching element. This transistor has a very widespread use and therefore low cost. The 2N2222A transistor can provide up to 800mA of current which is sufficient to drive the LEDs, as they require only 20mA of current each. Having such a great difference between the max current through the switch and the required current to turn on an LED means that if need be a single transistor switch can switch multiple LEDs without fear of exceeding the maximum switch current and therefore damaging the transistor. The calculations for the LED switches are shown below. The LEDs we have selected are high output LEDs and have a max current of 20mA at 2V. Factoring small element tolerances and a margin for safety, the max current through an LED should be no more than 18mA.

\[
V_{bat} = 9.6V
\]
\[
V_{cc} = \frac{V_{bat}}{2}
\]
\[
V_{cc} = 4.8V
\]
\[
V_{CEsat} = 0.3V
\]
\[
I_{f_{LED, max}} = 20 \cdot 10^{-3} \text{ A}
\]
\[
I_{f_{LED}} = 18 \cdot 10^{-3} \text{ A}
\]
\[
Hfe_{typ} = 100
\]
\[
Hfe_{min} = 75
\]
\[
R_c = \frac{V_{cc} - V_{f_{LED}} - V_{CEsat}}{5 \cdot I_{f_{LED}}} \Omega
\]
\[
R_c = 27.778 \Omega
\]

- The value for Rc is not a standard resistor value, so a standard value of 27Ω will be used.
In order to operate the transistor as a switch, the condition $I_b \cdot H_{fe} > I_c$ must hold. As discussed in electronics classes, transistor $H_{fe}$ can vary greatly between transistors. According to the 2N2222A datasheet the minimum $H_{fe}$ is 75, but that only holds for certain conditions such as the power supply voltage, collector current, and ambient temperature. For a worst case scenario calculation $H_{fe}$ is estimated to be 50.

The collector current is 5 times the current needed for a single LED.

Since this switch will be controlled directly by the microcontroller the input voltage is going to be 4.8V. According to the datasheet, the forward voltage $V_{BE_{sat}}$ is between 0.6 and 1.2V. I estimate it to be 0.7V.
$V_{BE_{sat}} = 0.7V$

The base resistor will have to limit the current to no less than 1.8mA as calculated above.

$VRb = V_{cc} - V_{BE_{sat}}$
- The voltage across the base resistor is 5V minus the base emitter saturation voltage.

$VRb = 4.1V$

The resistor values determined above will be the same for all LED transistor switches. All resistors have standard +/-5% tolerance. The designed circuit is shown above.

$$Rb\_LED = \frac{VRb}{Ib_{min}}$$

$$Rb\_LED = 2.278 \cdot 10^3 \Omega$$

- 2.2kΩ is a standard resistor value and is therefore used

**Section 1.2.2: Motor Control Circuit**

![Figure 27: Motor Control Circuit](image)

Four more transistor switches are necessary for controlling the motors. A standard H-bridge circuit topology will be used for motor control. The H-bridge will allow for direction reversal. The motor power will be controlled by a PWM (Pulse Width Modulation) signal from the microcontroller. A PWM signal is a standard way of controlling the speed of a DC motor. Bi-Polar junction transistors will be used for the switching elements. Fly-back diodes will have to be
placed in parallel to the CE junction of the transistors to provide path for the discharge of the energy stored in the motor coils, when direction reversal is called for. The selected transistor for a switching element is the BJT, NPN 2SD2150 and the PNP 2SB1592. Both can handle continuous current of up to 3A, have 0.3V collector-emitter saturation voltage and have $H_{fe\min}=120$. The selected fly-back diodes are the T6A05L. This diode can handle up to 6A continuous current and up to 300A of pulse current. To turn on and off the power transistors four additional transistor switches will be used. These additional transistor switches will be based on the 2N2222 transistor and will be directly controlled by the microcontroller. Two of the switches operate in a common emitter and two operate in a common collector schemes. This is done to ensure the same signal from the micro turns on or off two power transistors placed in one of the diagonals of the H-bridge. Calculations are shown below:

\[ V_{CE\text{sat}} = 0.3V \]
\[ V_m = V_{bat} - 2 \cdot V_{CE\text{sat}} \]
\[ V_m = 9V \]

For "lower" transistor common collector switch:

\[ I_{csat}_{T3} = 3A \]
\[ I_{bsat}_{T3} = \frac{3}{20} A \]
\[ I_{bsat}_{T3} = 0.15A \]
\[ I_{csat}_{T6} = 0.06A \]
\[ R_{e_{T6}} = \frac{V_{bat} - V_{CE\text{sat}}}{I_{csat}_{T6} - I_{bsat}_{T3}} \]
\[ R_{e_{T6}} = 930\Omega \]

- A standard value of 910Ω is used.

\[ R_{e_{T6}} = 910\Omega \]
\[ R_{b_{T6}} = \frac{V_{cc} - V_{BE\text{sat}}}{I_{Csat}_{T6}} \]
\[ R_{b_{T6}} = 1.281 \cdot 10^3 \Omega \]

- A standard value of 1.2kΩ is used.

\[ R_{b_{T6}} = 1200\Omega \]
\[ R_{b_{T3}} = \frac{V_{bat} - V_{CE\text{sat}} - V_{BE\text{sat}}}{I_{bsat}_{T3}} \]
\[ R_{b_{T3}} = 57.33\Omega \]

- A standard value of 56Ω is used.

\[ R_{b_{T3}} = 56\Omega \]
The above calculated values fully describe the transistor switch needed to turn on and off. An identical switch is used for T4.

\[ I_{csat_{T1}} = 3A \]

\[ I_{bsat_{T1}} = \frac{I_{csat_{T1}}}{75} \]

\[ I_{bsat_{T1}} = 0.04A \]

- 50mA is assumed.

\[ I_{bsat_{T1}} = 0.05A \]

\[ I_{csat_{T5}} = 0.055A \]

\[ I_{bsat_{T5}} = \frac{I_{csat_{T5}}}{50} \]

\[ I_{bsat_{T5}} = 1.1 \cdot 10^{-3}A \]

\[ R_{cT5} = \frac{V_{bat} - V_{CEsat}}{I_{csat_{T5}} - I_{bsat_{T1}}} \]

\[ R_{cT5} = 1.86 \cdot 10^3 \Omega \]

- A standard value of 1.8kΩ is used.

\[ R_{cT5} = 1800\Omega \]

\[ R_{bT5} = \frac{V_{cc} - V_{BEsat}}{I_{bsat_{T5}}} \]

\[ R_{bT5} = 3.727 \cdot 10^3 \Omega \]

- A standard value of 3.6kΩ is used.

\[ R_{bT5} = 3600\Omega \]

\[ R_{bT1} = \frac{V_{bat} - V_{CEsat} - V_{BEsat}}{I_{bsat_{T1}}} \]

\[ R_{bT1} = 172\Omega \]

- A standard value of 160Ω is used.

\[ R_{bT1} = 160\Omega \]

The above described calculations completely define the resistor values needed for the transistor switches that will turn on and off the "upper" power transistor.

In order to provide current protection a 0.01Ω resistor is used. At 3A of current the voltage across the resistor is going to be only 3mV. That voltage given the 10bit A/D converters in the microcontroller should be sufficient to trip a safety subroutine which will disengage the motor power.

**Section 1.2.3: Accelerometer Circuits**
The selected accelerometers are the **LDT0-028K** from MSI for their low cost and 50mV/g sensitivity. This particular device can sense acceleration on one axis. The accelerometer signal will be used only to sense sharp spikes in acceleration, so precision is not the most important factor in designing the circuit that will amplify the accelerometer signal. A 741 op-amp is used to buffer and amplify the signal from the device, which we expect to be no more than 5V since the accelerometer has sensitivity of 50mV/g and rated 1.4V/g at 180Hz resonance. With these characteristics in mind the signal will only be buffered in order to provide sufficient current output. Acceleration needs to be sensed on two axes and therefore two identical amplifying/buffering circuits will be required. Using the inverting channel of a 741 op-amp, the resistor calculations are shown below:

\[ R_{accel_i} = 10000 \Omega \]
- Value chosen for the input resistor
\[ Aa = 1 \]
- Coefficient of amplification
\[ R_{accel_f} = R_{accel_i} \cdot Aa \]
- Feedback resistor
\[ R_{accel_f} = 1 \cdot 10^7 \Omega \]

The non-inverting input of the op-amp will be grounded. In that way the input resistance to the amplifier is simply \( R_{accel_i} \).

An op-amp requires the use of a power supply of +/- X volts with respect to ground. We use a simple two capacitor voltage divider and connect the ground in between the two capacitors. In doing so, voltage of **4.8V** is achieved, which will also be used to power the microcontroller. The chosen capacitors \( C_2 \) and \( C_3 \) are **10uF**, which should be sufficient since no large currents are expected to be drawn by the op-amps or the microcontroller.

**Section 1.2.4: Microphone Circuit**
The microphone circuit is very much the same topology as the accelerometer circuit. Again precision is not going to be of great significance since only large spikes in the sound levels will be monitored. A loud bark will result in an input to the microcontroller that will indicate a certain action needs to be taken. The selected microphone is the Panasonic WK-62PC, for its low cost and resistance to vibration. The voltage output of the electret microphone is expected to be maximum of 100mV, so an amplification coefficient of roughly 50 is needed. This microphone has impedance of $2.2\,\Omega$, so a matched input resistor of $2.2\,\Omega$ is used for maximum efficiency. A capacitor in series of $0.5\,\mu F$ will be used to separate the DC voltage from the amplifier circuit. The only unknown in the amplifier circuit is the feedback resistor. The calculation for finding the feedback resistor is shown below:

\[
Am = 50 \\
R_{i_{mic}} = 220\,\Omega \\
R_{mic_f} = Am \cdot R_{i_{mic}} \\
R_{mic_f} = 1.1 \cdot 10^5\,\Omega
\]

Section 1.2.5: Digital IR Remote Control Circuits
The digital IR remote control circuits will provide signals with three different frequencies. These frequencies will be interpreted by the microcontroller as power on/off, activity level 1 and activity level 2. The remote control utilizes a 32.768kHz crystal oscillator. 2N2222 transistor switch is used to boost the current to sufficient the sufficient level of 500mA. Hfe is assumed to the same as in the previous 2N2222 switches. This signal is fed into a JK flip-flop connected as a T flip flop. The output of that JK flip flop is fed to another T flip-flop. This circuit configuration provides three different frequencies at 32.768kHz, 16.384kHz and 8.192kHz. Those signals are fed to a transistor switch, which drives an IR LED. The IR LED requires max current of 500mA of pulsing current. The IR diode I have chosen is the Avago Technologies HSDL-4420. This diode operates with at 875nm wavelength. The photo transistor is the QSE113 from Fairchild Semiconductor. The remote control will be powered by two lithium cell batteries at 3V each, for a total DC voltage of 6V.

\[ V_b = 6V \]
\[ I_{f_{IR\_LED}} = 0.500\, A \]
\[ R_{ir} = \frac{V_b - V_f_{LED} - V_{CESAT}}{I_{f_{IR\_LED}}} \]
\[ R_{ir} = 7.4\, \Omega \]
- A standard value of 7.5Ω is used.
\[ I_{C_{IR\_on}} = I_{f_{IR\_LED}} \]
- The collector current is the same as the current needed for the IR LED.
\[ I_{C_{IR\_on}} = 0.5\, A \]
\[ I_{b_{IR\_min}} = \frac{I_{C_{IR\_on}}}{H_{fe_{est}}} \]
\[ I_{b_{IR\_min}} = 0.01\, A \]

Since this switch will be controlled directly by the flip-flops or the crystal oscillator, the input voltage is going to be 6V. The forward voltage VBEsat according to the datasheet is between .6 and 1.2V. I estimate it to be 0.7V.
$V_{BE\text{sat}} = 0.7V$

The base resistor will have to limit the current to no less than 360 microAmperes as calculated above.

$V_{Rb_{\text{IR}}} = V_b - V_{BE\text{sat}}$
- The voltage across the base resistor is 5V minus the base emitter saturation voltage.

$V_{Rb_{\text{IR}}} = 5.3V$

$R_{b_{\text{IR,LED}}} = \frac{V_{Rb_{\text{IR}}}}{I_{b_{\text{IR, min}}}}$

$R_{b_{\text{IR,LED}}} = 530\Omega$
- A standard value of 510$\Omega$ is used.

$R_{b_{\text{IR,LED}}} = 510\Omega$

![Figure 31: Circuit Diagram for Digital IR Remote Control Receiver Unit](image)

The IR transmitter power is about 1W out of which 900mW is power used to create light and 100mW is dissipated as heat. This light is dispersed in a 24 degree angle. We want to be able to operate the remote control from max distance of 10ft. At that distance the area that the LED light will cover will be.
\[ A = \pi \cdot \left[ \tan \left( \frac{14 \cdot \pi}{180} \right) \cdot (10 \cdot 30.48) \right]^2 \]

\[ A = 1.814 \cdot 10^4 \text{cm}^2 \]

\[ P = 0.9W \]

- IR LED power

\[ \text{Pr} = \frac{P}{A} \]

\[ \text{Pr} = 4.96 \cdot 10^{-5} \frac{W}{\text{cm}^2} \]

- Normalized power at receiver in W/cm²

At this power level we expect light current of no more than 1μA. To operate this transistor as a switch the collector is connected to power through a 10MΩ, R38 resistor. That resistor value ensures that the photo transistor operates as a switch. A 741 op-amp is used as a voltage follower to buffer the signal with an amplification coefficient of 1. 1MΩ resistors, R36 and R37, are used for input resistors and for feedback resistors for the inverting channel of the op-amp.

**Section 1.2.6: Speaker circuit**

![Circuit Diagram for Speaker](image)

The speaker will be driven directly by a square wave from the microcontroller. In order to filter out the high frequency components of the square wave a simple RC low pass filter is designed. The cutoff frequency is set to 4kHz. The maximum current the microcontroller can supply is around 20mA, so the DC current is limited to 20mA by the resistor \( R_{filter} \).

\[ R_{filter} + 8\Omega = \frac{V_{cc}}{20 \cdot 10^{-3}} \]

\[ R_{filter} = 232\Omega \]
4000Hz = \frac{1}{2 \cdot \pi \cdot R_{\text{filter}} \cdot C_f}

C_f = 1.715 \cdot 10^{-7} \, F

- A standard value of 180nF is used.

\[ C_f = 180nF \]

**Section 1.2.7: Microcontroller**

**PDIP, SOIC, SSOP (28-pin)**

Operation of the ISPU will be controlled entirely by a microcontroller. The microcontroller will sense sound, via a microphone, and the units’ motion, via two accelerometers mounted in a way that allows to sense acceleration in two perpendicular directions. Based on those inputs and pre-programmed algorithms the units’ motors will be engaged, lights will be turned on and off and sound will be generated. To do all of the above tasks the Microcontroller we have selected is the PIC16F737. It is a **28 pin, 8-bit CMOS Flash Microcontroller**. The microcontroller can operate at clock frequencies of up to **8MHz**, using its own **internal RC oscillator**, and frequencies of up to **20MHz**, using an **external crystal oscillator**. The 28-pin package we have selected has **4KB of program memory** and **368 bytes of Data SRAM**. This package has **three 8-bit ports**. Each pin on all three ports can be individually configured as a digital input or output. Eleven of the I/O pins can be configured as **10-bit A/D converter inputs** and another three can be configured as **PWM outputs**. Pins AN0 and AN1 will the analog inputs used for the two accelerometers. Pin AN2 will be used as the analog input form the microphone. Pin RA4 will be the digital input from the remote control light sensing circuit. Pins RA5, RA6, and RA7 will be the processor outputs going to the RED, GREEN, and YELLOW LED circuits. Pins CCP1 and CCP3 will be the PWM output controlling the motor drive circuit. Pin RB1 will be the output to the speaker circuit.

A very convenient feature for development of new hardware and software, that this microcontroller offers, is in-circuit debugging. What this means is that using a tool called an **In-**
Circuit Debugger one can program the microcontroller via only two pins without having to remove the chip from the circuit board. Using extra two pins an In-Circuit Debugger can even power our circuits, so testing can be done without draining the battery pack. The PIC assembly language, that can be used to program this microcontroller, contains only 35 instructions in its instruction set. In our opinion programming entirely in assembly language may prove to be too great of a challenge for us, so we looked at external C compilers. Through extensive research we have found compilers that allow for programming in languages such as C, C++ and combinations of C/C++ and assembly, for even greater flexibility. Currently we plan on using a compiler that allows for programming in both assembly and C.

Section 1.2.8: PIC16F737 Pin Purposes
The following is a chart of the functions intended for each pin on the PIC16F737:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function Utilized</th>
<th>Intention</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MCLR'</td>
<td>MCLR'</td>
</tr>
<tr>
<td>2</td>
<td>AN0</td>
<td>Accelerometer 1 Input</td>
</tr>
<tr>
<td>3</td>
<td>AN1</td>
<td>Accelerometer 2 Input</td>
</tr>
<tr>
<td>4</td>
<td>AN2</td>
<td>Current Overload Check 1</td>
</tr>
<tr>
<td>5</td>
<td>AN3</td>
<td>Current Overload Check 2</td>
</tr>
<tr>
<td>6</td>
<td>RA4</td>
<td>IR Input</td>
</tr>
<tr>
<td>7</td>
<td>AN4</td>
<td>Microphone Input</td>
</tr>
<tr>
<td>8</td>
<td>VSS</td>
<td>GNDref=leave alone</td>
</tr>
<tr>
<td>9</td>
<td>Unused</td>
<td>x</td>
</tr>
<tr>
<td>10</td>
<td>Unused</td>
<td>x</td>
</tr>
<tr>
<td>11</td>
<td>Unused</td>
<td>x</td>
</tr>
<tr>
<td>12</td>
<td>CCP2</td>
<td>PWM at Motor 1</td>
</tr>
<tr>
<td>13</td>
<td>CCP1</td>
<td>PWM at Motor 2</td>
</tr>
<tr>
<td>14</td>
<td>Unused</td>
<td>x</td>
</tr>
<tr>
<td>15</td>
<td>Unused</td>
<td>x</td>
</tr>
<tr>
<td>16</td>
<td>Unused</td>
<td>x</td>
</tr>
<tr>
<td>17</td>
<td>Unused</td>
<td>x</td>
</tr>
<tr>
<td>18</td>
<td>Unused</td>
<td>x</td>
</tr>
<tr>
<td>19</td>
<td>VSS</td>
<td>GNDref=leave alone</td>
</tr>
<tr>
<td>20</td>
<td>SDD</td>
<td>POWER</td>
</tr>
<tr>
<td>21</td>
<td>RB0</td>
<td>Yellow LEDs</td>
</tr>
<tr>
<td>22</td>
<td>AN10</td>
<td>Microphone Input (jumped to pin #7)</td>
</tr>
<tr>
<td>23</td>
<td>AN8</td>
<td>IR Input (jumped to pin #6)</td>
</tr>
<tr>
<td>24</td>
<td>RB3</td>
<td>Green LEDs</td>
</tr>
<tr>
<td>25</td>
<td>RB4</td>
<td>Red LEDs</td>
</tr>
<tr>
<td>26</td>
<td>RB5</td>
<td>Speaker Output</td>
</tr>
<tr>
<td>27</td>
<td>Unused</td>
<td>x</td>
</tr>
<tr>
<td>28</td>
<td>Unused</td>
<td>x</td>
</tr>
</tbody>
</table>

Table 2: Pin Purposes

The chart is modified from its version in the first semester by the addition of jump wires between pins 22 and 7, and pins 23 and 6. That intention and addition of jump wires is explained thoroughly in the Building Processes and Modifications section of this report.

Section 1.3: Software Design
Section 1.3.1: Description of Selected Coding Language, Compiler, and In-Circuit Debugger (ICD)

Before invoking on the Detailed Design, we had narrowed options for coding languages to C, Assembly Language, and a combination of the two. A compiler was selected based on concern for the parameters in rank order:

- Accessibility
- Ease of use
- Options for expansion (an ability to permit simultaneous and friendly coding in Assembly in case that level of control became necessary)
- Cost

The team was fortunate to find a compiler that met the best possibilities for each parameter. Upon research, compilers were discovered that were easy to order/required going through several representatives, provided useful interfaces/leaves coding to the expertise of the engineer, allowed options for simultaneous coding in Assembly/did not mention the possibility, and cost anywhere from $250 to $1050 (excluding limited freeware). The mikroElektronika C Compiler was available by web download, provided an explained interface for coding in Assembly along with C, and cost USD250. The company’s website, http://www.dontronics-shop.com/product.php?productid=16197, was clear in explaining mikroC’s other benefits by feature name:

- Code Explorer, an interface involving windows with drag options for task simplification
- Quick Help, for making use of built-in routines
- Keyboard Shortcuts
- Code Assistant, to recount and call up previously used algorithms/routines
- Auto Correct
- Code Templates
- Syntax Highlighting
- Undo/Redo

When it came to coding in the second semester, mikroC’s interface for coding in Assembly alongside C proved to be its most useful feature. The final program was written in a near 50/50 combination of C and Assembly Language. Problems were encountered in troubleshooting that could only be overcome by controlling the PIC16F737 bit-by-bit. The following is a mikroC screenshot, showing implementation of the `asm{ ... }` directive to insert blocks of Assembly Language within C (Assembly Language in red, C in white/blue):
Microchip MPLAB was chosen for system ICD, based on team members’ familiarity with MPLAB from prior work. The following are those steps necessary to interface mikroElektronika’s mikroC Compiler to Microchip’s MPLAB ICD to download code to the PIC16F737:

1. Open the mikroC program
2. From the Project menu, select “Add New Project”
   - Name the new project according to its purpose
   - Select and highlight a desired project directory
   - Hit “Ok”
3. Write C source code and Assembly Language in the black window as shown above
4. From the Project menu, select “Save Project”
5. From the Project menu, select “Build.” This compiles the project. Wait until a pop-up occurs that reads “Build Succeeded,” then hit “Ok”
6. Exit mikroC or minimize its window to open MPLAB
7. In MPLAB, from the File menu, select “Import”
   - Find and select the mikroC *.hex file that compilation created. Hit “Ok”
8. Make sure that the PIC16F737 is connected to the ICD and powered for downloading
9. From the MPLAB Programmer menu, under Select Program Device, select “2 MPLAB ICD 2.” Wait while MPLAB verifies that the PIC16F737 is connected and ready

Figure 34: Use of the \texttt{asm} (...) directive in mikroC
10. From the Configure menu, select “Configuration Bits” and use the drop-down menus under “Setting” to change them so that they match this:

![Figure 35: Configuration Bits in MPLAB](image)

11. Exit the Configuration Bits screen by clicking on the X in the upper-right corner of the window.

12. Click on the Program Target Device icon in the upper toolbar.

![Figure 36: Program Target Device Icon](image)

13. The program will be ready to run, and it can be run with the PIC16F737 connected to the ICD still or not.

   - To run the program with the PIC16F737 still connected to the ICD, from the Programmer menu, select “Release from Reset”
   - To run the program with the PIC16F737 disconnected from the ICD, simply disconnect the RJ12 connector.

Section 1.3.2: Software Tasks List

The following is a list of tasks the ISPU software is responsible for.

1. IR Interaction
   a. Recognizing input signals
      i. Distinguishing 16kHz as a signal to turn the unit ON/OFF
      ii. Distinguishing 8kHz as a signal to toggle the unit’s activity level (move fast or slow)
   b. Triggering appropriate response
      i. Turning the ISPU ON/OFF
      ii. Toggling activity level and altering motors’ PWM accordingly

2. Motion Control
   a. Implementing movement algorithm in loop
   b. Forwards for ~10sec, stop for ~5sec, backwards for ~10sec, stop for ~5sec
   c. Account for activity level in PWM duty cycles

3. Lighting
a. Implementing lighting algorithm in loop
b. Having LEDs flash with each start/stop of the motors
c. Having LEDs signal received inputs (IR frequencies of 16 or 8kHz, pet interactions in the form of accelerometer input or sound sensed at the microphone) as suitable for project testing

4. Sound
a. Implementing sound algorithm in loop
b. Having the unit beep with each start/stop of the motors

5. Pet Monitoring
a. Determining whether input is sensed from the pet on either of two accelerometers
b. Determining whether input is sensed from the pet on microphone
c. Checking accelerometers and microphone continually while motors are stopped (stopped motors do not interfere with pet signals)
d. Shutting down unit if pet has not engaged for 10 motion cycles

6. Current Protection
a. Shutting down the unit if current overload is detected

**Section 1.3.3: High-Level Algorithm Flowchart**
The following is a most simple overview of the final ISPU code:

**Figure 37: Highest-level algorithm flowchart**

**Section 1.3.4: Lower-Level Algorithms Flowcharts**
The following are flowcharts explaining the more complicated and integrated functions of the main program, followed by explanations:

Figure 38: Program Beginning
Upon reset, the program begins by clearing its memory registers and initializing/configuring variables and ports to accept input from the IR remote. IR input is sensed at PORTA, pin 4 (RA4) of the PIC16F737. According to the PIC16F737 datasheet, RA4 (when configured for input) is held at default high voltage (+5.0V) until it receives a low voltage (<2.0V). Therefore, the software senses IR input by determining its first received low voltage. From that moment, the program records 80 bits. The number of ones and zeros in the bit stream is compared against < or >15 ones criteria to determine if 10001000 (for turning the unit ON/OFF) or 10000000 (to toggle activity level) is received. From the program beginning, if an ON/OFF signal is detected, the program will jump to its play algorithm (“Drive Forwards”). If the signal is detected for toggling activity level, the program will jump back to its beginning to wait for the right ON/OFF signal to be received.

Figure 39: Drive Forwards
Driving forwards begins by initializing registers for PWM. This is performed in the final code in Assembly Language and is necessary before each time the motors are run because at other points in the program, the registers are used for other purposes. The user-defined Activity Level is used to determine whether the unit should roll fast or slow, then the LEDs and buzzer are turned on for 20ms to capture the pet’s attention and signal the beginning of the driving sequence. The Pet Monitoring Variable, initialized to 10 at the program beginning, is decremented in case the pet is not active. In the case that the Pet Monitoring Variable is let decrease to zero, the unit shuts itself off to save power. “Driving Timing” is the loop used to control how long the ISPU stays in motion. The Motion Variable being set to Forwards allows “Driving Timing” to know where in the program to jump back to after timing is complete.

![Diagram of Driving Timing](image)

**Figure 40: Driving Timing**

“Driving Timing” controls how long the ISPU stays in motion for by a decrementing-variable loop. The Looping Variable is initialized to 100. The number 100 was determined by testing to be approximately the number of times the process has to through its loop to create a delay of 5
seconds (a running time practical for testing in the lab). The loop consists of a check to make sure that circuit-allowable currents are not exceeded (the motors are not overheating) and a check to see if the Looping Variable equals zero and where then the program should jump back to (the point after moving forwards or moving backwards).

The “Stop After Forwards” process first turns off the motors. The LEDs and buzzer are turned on for 10ms to draw the pet’s attention and signify the beginning of the unit’s rest period. The Stop Variable is set to Forwards so that the program knows where to jump back to after “Stop Timing.”

Figure 41: Stop After Forwards

The “Stop After Forwards” process first turns off the motors. The LEDs and buzzer are turned on for 10ms to draw the pet’s attention and signify the beginning of the unit’s rest period. The Stop Variable is set to Forwards so that the program knows where to jump back to after “Stop Timing.”
Stop Timing

Set Looping Variable (LV) to 50

Decrement LV

Is LV < 0 and Stop Variable = Forwards?
Yes
Go to Drive Backwards
No
Is LV < 0 and Stop Variable = Backwards?
Yes
Go to Drive Forwards
No

Configure analog ports for receiving data from all analog interfaces

Does Accelerometer 1 sense pet?
Yes
Reset Pet Monitoring Variable to 10 and flash red LEDs
No
Does Accelerometer 2 sense pet?
Yes
Reset Pet Monitoring Variable to 10 and flash green LEDs
No
Is current limit exceeded?
Yes
Clear configurations to not interfere with other algorithms
No

Does microphone sense pet?
Yes
Reset Pet Monitoring Variable to 10 and flash yellow LEDs
No
IR signal detected?
Yes
Go to Program Beginning
No

Collect bits

Are 80 bits collected?
Yes
< 15 ones means the button to toggle Activity
< > 15 ones in signal?
Yes
Set Activity Level = 2
No
Set Activity Level = 1

Signal 16kHz received with red LEDs
> 15 ones means OFF

Signal 8kHz received with green LEDs

Figure 42: Stop Timing
The “Stop Timing” process operates like the “Driving Timing” process in that it operates on a decrementing-variable loop. 50 was the value of Looping Variable determined by testing to be sufficient in creating a 2.5 second delay (a pause time practical for testing in the lab). The loop consists of checking both accelerometers for input (input would mean the pet is touching the ball, so if input is sensed the Pet Monitoring Variable is reset to 10), checking for current overload (motor overheating, in which case the toy would shut off), checking the microphone for input (input would mean pet sounds are sensed, so if input is sensed the Pet Monitoring Variable to 10), and checking for IR input (to turn the unit off or toggle its Activity Level). According to these processes, “Stop Timing” may or may not exit its loop before the Looping Variable equals zero. The program returns to driving forwards, driving backwards, or the program beginning, whichever is appropriate.

![Diagram of Drive Backwards process](Figure 43: Drive Backwards)
“Drive Backwards” is similar to “Drive Forwards” in logic except that the motors are set to drive in the opposite direction and the Motion Variable is set to Backwards instead of Forwards.

```
Stop After Backwards

Turn off PWM

Turn on LEDs and buzzer for 10ms

Set Stop Variable = Backwards because we're stopping after driving backwards

Go to Stop Timing
```

Figure 44: Stop After Backwards

“Stop After Backwards” is similar to “Stop After Forwards” in logic except that the Stop Variable is set to Backwards (instead of Forwards) so that the program knows where to appropriately jump back to after “Stop Timing.”

The comprehensive, [tentatively] final code for the IPFW ISPU prototype is contained as Appendix to this report.

Section 2: Building Processes and Modifications

Section 2.1: Building Process

The initial step of the building phase was the acquisition of the components necessary. The mounting and battery tray, motor mounts as well as the motor clamps were laser cut and formed by Grainbelt Supply Company per our supplied prints. The connecting links were fabricated by Kronmiller Machine & Tool via milling per our supplied prints. Miscellaneous hardware such as nuts, bolts and dowel pins were ordered from various industrial supply companies. The primary component of the unit, the shell, was created out of Duraform (nylon) via the Engineering Department’s SLS machine.

Upon the attainment of all the components the assembly was as follows:

1. Battery pack & tray were assembled utilizing self adhesive Velcro strips.
2. Circuit board was screwed to the mounting tray utilizing 6-32 x 1 inch pan head standard screws and nyloc nuts. Gromets were utilized to prevent the screws from touching the hot side of the PCB.
3. The motors were attached to the motor clamps and motor mounts using 4-40 socket head cap screws (SHCS) and the coupler was attached to the motor shaft.
4. The motor mounts were attached to the mounting tray with 6-32 SHCS and nyloc nuts.
5. The mounting tray and battery tray were connected using the connecting links, retaining clips and dowel pins.
6. Installed O-rings and threaded inserts in each side of the primary shell component.
7. Installed sub-assembly into the shell and attached access panels with 8-32 SHCS.

The assembly process is shown in Figure 45.

![Figure 45: Exploded view of assembly](image)

**Section 2.2: Mechanical Modifications**

**Section 2.2.1: Replacement of the 0.125 inch Dowel Pins & Clips**

During testing of the ISPU, it was found that when the unit collided with walls or the motion was very sporadic, the clips used to secure the 0.125 inch dowel pins would work themselves loose and the components would fall apart. The solution was to utilize 6-32 SHCS with a self-locking nyloc nut in place of the dowel pins and clips.
Figure 46: Dowel Pin & Clip With Replacement 6-32 SHCS & Nyloc Nut

Section 2.2.2: Revised Motor Couplers

To keep budget costs low and help with manufacturing deadlines, the designed motor couplers were replaced with an existing pair available from a student’s erector set. The print was revised to reflect the changes needed to incorporate the new style of motor couplers.

Section 2.2.3: Replaced Solid Access Panels with Clear Access Panels

Due to budget limitations, the entire shell of the ISPU was manufactured from a nylon material that was solid in color. This later became an issue since the remote control of the unit communicated with the on board computer via infra red (IR) technology. The solution was to have the access panels machined from blocks of clear acrylic plastic. This allowed the remote to
communicate with the units on board computer. Also, since the bolt circle diameter was modified to fit the new style motor couplers, the corresponding bolt circle for the dowel pins had to be modified as well.

![Figure 47: Initial & Revised Access Panels](image)

**Section 2.2.4: Installation of O-ring Seal on Bolts**

During the water resistance testing it was found that water was penetrating internal of the shell. The water was determined to be coming in around the bolts. The correct solution would be to move the mounting bolts to the outside of the o-ring. However, since this was not possible due to lack of room, small o-rings were installed on each 8-32 SHCS. This provided a seal between the bolt and counter bore of the access panel and eliminated water and moisture from reaching the interior of the shell.
Section 2.2.5: Counter bore Diameter & Depth for Threaded Inserts

During the assembly process it was found that the threaded inserts used for securing the access panels to the shell were not recessed enough to allow proper sealing of the o-ring with the access panels. The problem was determined to be incorrect dimensions of the threaded insert that were posted on the company’s website. The head of the insert was a little larger than indicated and rounded, making it taller than specified on the website. The solution was to increase the diameter and depth of the counter bores in the ISPU’s shell.
Figure 49: Revised Print for ISPU Shell
Figure 50: Revised Access Panel Print
Figure 51: Revised Motor Coupler Print
Section 2.3: Electrical Hardware Modifications

Section 2.3.1: Addition of Reed Switch
In order to have the ability to charge the unit, a reed switch has been added in series between the 9.6 Volt output of the battery pack and input to the voltage regulator.

![Circuit diagram of reed switch (S3) added to input of voltage regulator](image)

Figure 52: Circuit diagram of reed switch (S3) added to input of voltage regulator

The reed switch is a double pole single throw switch that has normally open and normally closed switches in the unit. For this project we utilize only the normally closed portion of the switch. The switch is physically located in the ISPU on the side of the battery tray closest to the shell.

![Installation of Reed Switch on the bottom of the battery tray](image)

Figure 53: Installation of Reed Switch on the bottom of the battery tray
When the ISPU is placed in the charging station, the reed switch enters the field of a strong magnet, opening the reed switch. This disconnects power to the microprocessor, preventing surges to the microprocessor and allowing the batteries to be attached to the charger.

The reed switch has the added benefit of providing a way to cut power to the microprocessor if the unit acts in an erratic way. In the event that the ISPU is unable to respond to inputs from the IR remote to turn off, the magnet can be swiped over the reed switch and the microprocessor will be reset.

**Section 2.3.2: Accelerometer Gain and Input Impedance Increased**

Initial testing of the accelerometer circuits demonstrated that the input voltage to the op amp circuits, fed by the accelerometers, was too low. Through further testing we determined that two factors were influencing the voltage output of the accelerometers and input to the op-amp. The first factor was the low input impedance of the op-amp input resistor. The relatively low input impedance of 10 kΩ caused excessive current draw from the acceleration sensor and therefore was effectively shunting the signal to ground instead of simply feeding it to the op-amp input. The input resistor was changed from 10 kΩ to 220 kΩ. The value of this new input resistor of 220kΩ was determined through experimentation, because sufficient data was not provided in the sensor datasheet.

After the input resistor was changed we needed to also change the feedback resistor to avoid creating an attenuator for the accelerometer signal. Initially the feedback resistor was exchanged for another 220kΩ resistor to achieve unity gain as the original design called for. Testing of the circuits with this resistor configuration was successful in that the circuit buffered the sensor voltage and produced a clean output signal.

Further investigation of the circuit under more “real life conditions” showed that the acceleration the sensor was seeing was less than anticipated and the signal being output to the microcontroller was too small. Scope measurements with an AC coupled input showed voltage amplitude of approximately 0.2V. The scope had to be AC coupled because of the 2.6V DC offset of the signal. The 0.2VAC voltage level is certainly detectable by the microcontroller, but it leaves the circuit more susceptible to noise. To make the circuit more resistant to noise and to avoid future challenges with DC offset, the signal was boosted to about 2VAC. To achieve this gain it was necessary to increase the feedback resistor to 2.2MΩ. Testing this final resistor combination confirmed the signal gain was sufficient for recognition by the microcontroller, even if the DC offset was to change. A picture of the redesigned circuit is shown below in Figure 54.
Section 2.3.3: Microphone Gain Increased
Testing of microphone circuit on a breadboard showed a signal output of the microphone lower than expected. Voltages seen from a strong input to the microphone (a loud noise) resulted in peak voltages of around .25V at the output of the op-amp. Changing the feedback resistor from 110 kΩ to 1.0 MΩ corrected the problem by increasing the gain from 50 to 450. The voltage levels seen at the output of the op-amp were in the 2VAC range and were comparable in amplitude to the accelerometer circuits output signals.

Section 2.3.4: Remote Control Receiver Redesign
The initial design of the remote control proved to work as desired, but it could only operate at a distance less than 1ft. In researching the problem we discovered the root cause to be the required transmission bandwidth. Through the above mentioned research we determined that a standard of 2 kbps or less is used in industry when a length of the transmission is greater than few inches. The signals that the remote control was designed to produce are square wave signals at 32kbps, 16kbps, and 8 kbps. During testing we determined that indeed the lower frequency of the received signal allowed for better reception of the signal at an increased distance. Even at the 8kbps transfer rate only about 1ft of distance was achieved with the original design.

In order to increase the distance a circuit modification was required. The selected LED limited us to speeds no less than 8kbps at the designed power level. Due to time constraints we could not change the infra red (IR) transmitter, so the change had to be on the receiver side. In order to boost the sensitivity of the receiver we increased the gain of the op-amp that was originally supposed to operate simply as a buffer. By changing the input resistor to 10kΩ and the feedback resistor to 1MΩ we were able to achieve gain of 100 or 20dB. This gain was able to increase the operation distance by another foot, so it was not sufficient. We experimented with a gain of a 1000 or 30dB, but this made the amplifier circuit highly unstable, so we went back to a gain of 20dB.

A decision was made to design a pre-amplifier that would boost the overall gain without making the receiver circuit unstable. This amplifier uses a 2N2222 BJT transistor and a “four resistor”
biasing scheme. This transistor was selected because we had it readily available and the amplification parameters seemed to be close to the desired for the application. The input and output for the pre-amplifier are AC coupled through 1µF capacitors. Those have sufficiently small impedance at the used frequencies. The highest impedance would be for the 8kHz signal and it is less than 20Ω.

\[ X_c = \frac{1}{2\pi fC} = \frac{1}{2\pi \cdot 8 \cdot 10^3 \cdot 1 \cdot 10^{-6}} = 19.89\Omega \]

Compared to the input impedance of the pre-amp and the input impedance of the next stage this value is considerably smaller and is therefore considered to be 0 in the AC circuit analysis. The pre-amp was designed to operate at the mid point of the output load line shown in Figure 55 or Vceq = 2.5V. The desired input impedance was decided to be 50kΩ, so it would not shunt the signal from the phototransistor, but amplify it. The circuit was initially designed to provide a gain of 50 by selecting the proper ratio of Rc and Re, but that did not provide sufficient gain. In order to obtain the maximum gain and be able to create a stable DC biased circuit a scheme with a capacitor bypassed emitter resistor was designed. In that way the gain of the circuit is highly dependent on the transistor \( \beta \), but it was the highest possible gain, so we decided to use this circuit topology. The relevant calculations are shown below.

The design parameters are as follows:
- \( R_{in} = 200k\Omega \)
- \( \beta = 100 \)
- \( V_{ceq} = 2.5V \)
- \( V_{bq} = 1.25V \)
- \( R_e = 1k\Omega \)

\[ \begin{align*}
I_{bq} &= \frac{V_{bq} - V_{be(on)}}{R_{in} + (1 + \beta)R_e} = \frac{1.25 - 0.7}{50 + 101 \cdot 1} = 0.00364mA \\
I_{cq} &= \beta \cdot I_{bq} = 0.364mA \\
V_{Rc} &= 5 - 2.5 - \beta \cdot I_{bq} R_e = 2.136V \\
R_c &= \frac{V_{Rc}}{I_{cq}} = 5.8k\Omega
\end{align*} \]

We select \( R_1 = 200k\Omega \) and calculate the resulting value for \( R_2 \).

\[ 50 = \frac{200 \cdot R_2}{200 + R_2} \quad \Rightarrow \quad R_2 = 66.6k\Omega \]

Standard resistor values of 220kΩ for \( R_1 \) and 68kΩ for \( R_2 \) are used in implementing the circuit.

The measured gain of this circuit is approximately 100 or 20dB. We couldn’t get a more precise value for the gain, because during testing it was very hard to keep the output from saturating. The scope graph is shown in the measurement section. In that way the overall gain of the IR receiver circuit is approximately 10,000 or 40dB. This high gain is necessary in order to saturate the output of the op-amp even with a very small input signal generated by the IR phototransistor.
Section 2.3.5: Capacitive divider changed to Resistive divider
The initial design called for capacitive voltage divider on the 5V power line. This divider was supposed to give us a 2.5V reference without consuming any current. During testing it was uncovered that the voltage divider was not operating as desired. Tolerances on the capacitors were sufficient to offset the voltage from 2.5V to 4V. In order to remedy this we decided to switch to a resistive divider instead. We used 2.2kΩ resistors for this purpose. The resistor values were selected for minimizing of the quiescent current draw. Currently the quiescent current in the voltage divider is:

\[
I_{\text{resistive-divider}} = \frac{\frac{V_{cc}}{2 \cdot R}}{4.4k\Omega} = 1.14mA
\]

Resistor tolerances are usually well within 5% tolerance and consequently the actual reference value we measured after this change was 2.6V, which is well within the tolerable range for proper operation.

![Figure 57: Change from Capacitive to Resistive divider circuit](image)

Section 2.3.6: Addition of filtering capacitors
The above mentioned change of removing the capacitive divider effectively removed any filtering we had on the 5V power line. The result was large voltage swings on the 5V power line due to the large interference generated by the DC motors. We believe these large voltage swings were the cause for the code execution problems we run into. The problem consisted in the microcontroller resetting itself shortly after engaging the motors. Under normal operating conditions the microcontroller could reset itself in two ways – by sensing a drop in the power level or by receiving a low signal on the MCLR (Master Clear) pin. In order to remove those voltage spikes we placed a 470µF capacitor in the circuit from the 5V line to ground. After this change the maximum voltage ripple that was measured with a scope on the 5V power line was on the order of 20mV. This ripple in our opinion is too small to interfere with the operation of the microcontroller. The code execution problems persisted, but we felt that keeping this
The 9.6V power also had large voltage spikes due to the direct connection of the motors through the H-bridge. We experimented with placing a 470µF capacitor across the 9.6V power as well. This change removed the voltage spikes to a large extent, but did not seem to improve the circuit operation, so we did not feel there was enough justification to include this capacitor in the final design.

Section 2.3.7: Microcontroller filter capacitors and MCLR resistor
Due to the already mentioned code execution problem we decided to add a resistor and a capacitor to the MCLR (Master Clear) line on the microcontroller. Since we knew there are some small voltage ripples on the microcontroller power line we thought the voltage spikes that appear on the MCLR pin may be large enough to reset the processor through a low voltage on the MCLR pin since it is an active low pin. We added a 2.2kΩ resistor to power and a 10µF electrolytic capacitor to ground. The microcontroller datasheet recommends only 1µF, but we felt that the larger value would be better considering the high noise motors we are causing. The addition of these elements again was not the solution to the problem, but we believe the addition is justified as it is recommended by the manufacturer to include those elements. In Addition to the MCLR filter capacitor and resistor a 100nF capacitor was added to the microcontroller 5V power pin. The function of this capacitor is to additionally filter any voltage spikes on microcontroller power pin. For this capacitor to function properly it is critical for the capacitor to be in close proximity to the power pin. We placed the capacitor directly between the power and ground pins (pin20 and pin19) on the microcontroller. Adding this capacitor improved the circuit operation dramatically. Glitches still happen to reset the microcontroller on rare occasions. Figure 58 shows the added resistor and capacitors.

![Figure 58: Filter capacitors and MCLR resistor](image)

Section 2.3.8: Addition of 560pF capacitor to H-bridge
The above mentioned microcontroller reset issue prompted a very detailed examination of the possible electro magnetic interferences (EMI) and voltage spikes throughout the circuit. An
attempt was made to eliminate EMI by shielding the motor. First we created 7 turn inductors that 
we wound around the motors to absorb any EMI and provide a low impedance path to ground. 
The inductor was grounded through a large copper braid to provide the required minimum 
resistance. The same copper braid was used to ground the motor mounting brackets and in that 
way we also grounded all metal part on the inner mechanics. We thought that in doing so we 
would eliminate practically all EMI and the induced voltage spikes, but that was not the case as it 
did not fix the random microcontroller reset issue.

During motor testing and trouble shooting of the random reset issue it was discovered that when 
an oscilloscope was connected to the connection between the motor leads and the cables coming 
out of the PCB the reset problem would disappear. The reset problem also disappeared if one 
was to hold the bare wire connection on the motors and the PCB cables. The common thing 
between those two instances is the addition of a small capacitance to the middle of the H-bridge. 
So then we placed a small 560pF capacitor from the midpoint of the H-bridge to ground and the 
problem was solved. We do not have a clear understanding of the processes that happen, so that 
the circuit does not reset itself anymore. We believe the capacitors act as a high pass filter and 
therefore ground any high frequency, high amplitude voltage spikes and prevent them from 
reaching the microcontroller. It is also our belief that the placement of the capacitors in close 
proximity to the motors is crucial as the reset problem did not disappear when the scope was 
connected to the midpoint of the H-bridge on the PCB. The added capacitors are labeled C7 and 
C8 and can be seen in the overall circuit diagram.

Section 2.3.9: Change of 741 op-amp to a LM324 quad op-amp
During the process of designing the printed circuit board it was discovered that space was going 
to be a serious issue. In order to save on space and cost we decided to change the 741 op-amps to 
a LM324 quad op-amp. This change allowed us to use one single chip that has all four op-amps 
that the circuit design calls for instead of having 4 separate chips. Effectively we reduce the total 
of 32 pins on the 741 op-amps to 14 pins on the LM324 and reduce the space taken by the 741 
op-amp by more than one half. This change also saves on cost for the final design. A single 741 
op-amp costs 40 cents and a single LM324 chip cost 45 cents. The resulting savings are 95 cent 
of parts cost. Figure 59 shows a size comparison of the 741 op-amp and the LM324 quad op-amp.

![Figure 59: Comparison of 741 Op-amp and LM324 quad op-amp sizes](image-url)
Section 2.3.10: Remote control crystal oscillator replacement

During testing of the remote control we discovered that the crystal oscillator circuit was not working at all. No oscillations were generated even after precisely following the manufacturer recommendations. We decided on a redesign that includes a 555 timer. The 555 timer is a simple 8-pin chip that requires two external resistors and two external capacitors. Depending on the obtained values a square output signal with different frequency and duty cycle is generated. Depending on the ratio of the resistors a different pulse width can also be achieved. The values we calculated and used were a C2=2.2pF ground filter capacitor and a C1=2.2nF capacitor from pins 6 and 7 to ground. Resistor R1=7.5kΩ and R2=1.8kΩ. This configuration gives us near 50% duty cycle at frequency of approximately 32kHz. The operation of the 555 timer is based on the following equations:

\[ T_{lo} = 0.639 \cdot R1 \cdot C = 11\mu s \]
\[ T_{hi} = 0.639 \cdot (R1 + R2) \cdot C = 13\mu s \]

The measured value for \( T_{hi} \) is 15\( \mu \)s. The measured value for \( T_{lo} \) is 14\( \mu \)s. The difference between the theoretical and measured periods is due to the tolerance in the element values. The measured frequency of oscillation for the primary frequency is 34.4kHz. The obtained duty cycle is about 52%.

Having done this redesign the remote control operates sufficiently close to the original design parameters. Figure 60 shows the complete remote control circuit.

Section 2.3.11: Jump Wires

Jump wires were added to the initial design to connect the IR and microphone inputs to PORTA of the PIC16F737 (from PORTB). This was because of time limitations on the project; Assembly language was necessary to invoke enough control on the PIC16F737 for reading inputs, and input reading was already programmed for the accelerometers at PORTA; time was saved by reading at PORTA and the risk of interfering with PORTB outputs was eliminated. A design
modification worth considering for future development of the ISPU is rerouting those wires to PORTA permanently. The jumps made:

1. Pin #22 (RB1, AN10) connected to pin #7 (RA5, AN4, LVDIN, SS’, C2OUT) for microphone input
2. Pin #23 (RB2, AN8) connected to pin #6 (RA4, T0CK1, C1OUT) for IR input

Section 2.4: Software Modifications

Section 2.4.1: Coding Language
Before invoking on the Detailed Design, we had narrowed options for coding languages to C, Assembly, and a combination of the two. After further research and consideration, we determined C as our language of choice. The primary reason behind this was an understanding of the dominance of algorithm waits and loops that would be involved in the final code (recognized through the process of writing pseudocode) and a knowledge of how relatively simple those are to implement in C as opposed to Assembly (through prior work with both languages). When we selected a compiler, we based it on concern for the parameters in rank order:

- Accessibility
- Ease of use
- Options for expansion (an ability to permit simultaneous and friendly coding in Assembly in case that level of control became necessary)
- Cost

When it came to coding in the second semester, the third parameter – options for expansion – ended up being the most fortunate on our list. The final program was written in a near 50/50 combination of C and Assembly Language. The implementation of Assembly Language was vital to our level of control needed over the microprocessor. Problems were encountered in troubleshooting that could only be overcome by controlling the PIC16F737 bit-by-bit.

Section 2.4.2: Software Tasks List and Algorithms Methods
The software tasks list was modified from the Fall semester as follows:

1. IR Interaction
   a. The need for IR to control the unit recharging was eliminated by a reed switch being placed at the MCLR’ (pin #1) of the PIC.

2. Engagement with Power Station
   a. The placement of the reed switch eliminated all responsibility of the unit’s software to control recharging whatsoever. Recharging will be an automatic process once a charging dock is purchased.
   b. The need for the unit to light up when fully recharged was eliminated by the decision to select a charging dock that would perform that indication on its own.

3. Lighting
a. It was decided not to take Activity Level into account when performing the unit’s lighting. The unit would blink for each beginning of start/stop motion.

4. Sound
a. It was decided not to take Activity Level into account when performing the unit’s sound output. The unit would beep for each beginning of start/stop motion.

5. Pet Monitoring
a. The task of pet monitoring was simplified to not involve how the pet is influencing the unit, but rather whether it is influencing the unit.

6. Current Protection
a. It was deemed necessary to install current overload protection for the motors via software. >3A sensed at pins #4 or #5 shuts the unit down.

The algorithms were altered tremendously over the Spring semester as the team came to recognize the capabilities and limitations of its system. The code changed with respect to new objectives (altered as stated in the Software Tasks List) and new development in learning what actually works best. The most major development was the introduction of Assembly Language to the program. The final program was written almost halfway in Assembly Language. Final project flowcharts are provided in the Selected Design section of this report. The final project code is provided as Appendix.

Section 2.5: PCB Design

Section 2.5.1: Layout and size optimization

Initially we wanted to simply use a proto-board to solder all the circuits, but that quickly proved to be very impractical due to the size constraints placed on us by the size of the mounting tray we use. The mounting tray is approximately 3.5 by 6.5 inches and the circuit board could not exceed those limits. In order to fit in those limits it was absolutely imperative that we use a printed circuit board (PCB). A two layer PCB was selected for the design.

The PCB design began by exporting all components from the MultiSim circuit diagrams to Ultiboard. Then one by one components were placed in approximately the positions that they are expected to be left at. It proved to be a good practice to group components that have to stay together.

Once the initial layout was done it was determined that the board size needed to be reduced by about two inches in each direction. In order to achieve that it proved to be easier to restart the design from the very beginning and try to place components closer to each other and sometimes use ingenuity to place components at seemingly more distant locations, but locations that had more empty space. Some of the components had to be placed in strategic locations and that sometimes reduced flexibility of the PCB design. For instance the IR phototransistor had to be placed near the edge of the board facing the clear end caps in order to receive the IR signal.
Due multiple connections on both sides of the chip, the microcontroller was placed in the center of the circuit to provide best access to all pins. The accelerometers that were used had to be placed perpendicular to each other in order to sense acceleration in both X and Y directions.

The final board size is about 3.5” x 5” and was milled out of a two layer pre tinned board for easier soldering.

**Section 2.5.2. Trace wiring**

Once the components were properly laid out the traces had to be routed. The standard trace width used is 32mils except for the 5V power trace, where 50mils is used. Another exception is the motor circuits, which need to conduct a maximum of 3A of current. That required a trace width of 100mils. The calculations were done through The Circuit Calculator, a website suggested by Professor Peter Goodman.

In order to save space and be able to wire the circuit properly both layers of the board were used. On occasions a pin would become cut off from the rest of the circuit and the only way to wire a trace to it was to jump over to the other layer and back to the starting layer. Connecting to the top layer was also necessary for connecting to the surface mount parts we are using. In connecting the circuit the bottom copper layer is used as the ground plane and the copper top is used as the 9.6V power plane.

**Section 2.5.3. Footprint issues**

In order to create a computer model for the printed circuit board every component has a footprint. The footprint of a component describes the pin locations relative to one another. During the export of parts from MultiSim to Ultiboard all parts are exported with a default footprint. Except for the resistors none of the parts had a footprint matching the footprint of the parts we ordered.

The transistors and the microcontroller had standard packages listed on the datasheets, so the footprints could easily be modified from the default to the standard packages that were ordered. For the rest of the components footprints had to be designed. For the accelerometers, for instance, I had to measure the distance between the pins and then leave hole spaced apart by the measured distance and at the correct diameter. The surface-mount resistors and capacitors we ordered did not specify the standard packages, so we had to measure the parts and find a package in Ultiboard that matched the measurements or again design a footprint that would fit the measurements.

Even when every single component, except for through-hole resistors, was touched upon and verified, issues still came up. For instance, all of the transistors including the surface mount ones were mirrored about the center pin. This is a very odd error that we attribute to the software, since it is the software that should have the pin definitions correct for the specified standard packages.
Luckily all parts except for the IR phototransistor could be simply turned 180 degrees. The IR phototransistor could not be turned around as it would face the inside of the ball and not be able to detect the IR signal. In order to still be able to use the part we had, we had to bend the legs so that the transistor would face the direction it was intended to face in – the outside of the ball. The final design is shown in the Appendix.

Section 3: Testing and Results, Measured Parameters

Section 3.1: Mechanical

With a final design selected and details having been fully developed the project moves to its testing phase. This document will outline all electrical and mechanical parameters or quantities that are to be measured or determined during the testing process in order to successfully verify the operation of the prototype.

Section 3.1.1: Performance Parameters

- Mobility – The unit must be capable of movement through grass of nominal density at a minimum speed of 0.5 mph with a maximum speed of ~3 mph.
- Climbing Requirements - The unit must be able to maintain forward movement up a slope with a ~5° incline.

Section 3.1.2: Testing Methods

- Mobility – The unit was tested for single-axis mobility on the following surfaces to verify its performance:
  - Smooth, flat surface: The engineering building hallways was used to verify basic operation of the unit.
  - Grass of nominal density: Mowed, maintained IPFW campus grass in common areas was utilized to test the performance of the unit on grass. 5° incline: Concrete walkways outside of Helmke Library were used to test the unit’s uphill performance.
  - The speed of the unit was calculated by recording the time it took for the unit to travel a known distance and dividing the distance by the time required.

Section 3.1.3: Durability Parameters

- Pin tear out – A Tinius Olsen machine was utilized to perform testing of the connecting links to verify sufficient resistance to pin tear out.
- Water resistance – The shell plastic material of the ISPU was tested to determine if it will withstand water. Additionally, the unit was tested to ensure that the interior components stay dry when unit is subjected to simulated rain fall.
- Force testing – Strain gages was applied to mounting tray and printed circuit board to verify the stress on the circuit board did not exceed the yield stress of the material.
Section 3.2: Testing Procedures and Results

Section 3.2.1: Mobility Test
The unit must be capable of movement through grass of nominal density at a minimum speed of 0.5 [mph] and at a maximum speed of approximately 3 [mph].

The unit must be able to maintain forward movement up a ~5 degree incline.

Procedure:
A program was downloaded into the unit that allowed the unit be ramped into full power over the course of 5 seconds. Power to the batteries was programmed to begin at 12.5% for 2 seconds, increased to 25% for 3 seconds and finally to 100%. The batteries were fully charged and the unit was taken outside to the grassy area between Neff Hall and the Classroom Medical Building. The unit was placed on the paved walkway, turned on and allowed to reach maximum speed before it encountered the grass. Upon examination of the program, it was found that it was not at 100% power.

For the incline test, the inclined walkway that runs between Neff Hall and the Classroom Medical Building was utilized. Again, the unit was allowed to reach maximum speed before it began its travel up the incline. Additionally an eight foot long piece of plywood was raised by 8 and \( \frac{3}{8} \) inches to create a 5º incline.

Results:
For the flat surface the distances and resulting times are shown in Table 3. Once it encountered the grass, the unit maintained forward motion for a short period before stopping. This was repeated since there was concern that the unit may have encountered a crevasse in the ground. After the unit had reached the grassy area, it maintained forward motion for a short period but then, came to rest. Even though several attempts were made, the end results were the same. Unfortunately, the inclined testing resulted in the same outcome. The unit would progress up the incline of 1.2º but begin to slow down, roll off to one side and roll back down the incline. With a modified program the unit was capable of maintaining forward progress up a 5º incline.

<table>
<thead>
<tr>
<th>Trial</th>
<th>Distance</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>19.5</td>
<td>2.6</td>
</tr>
<tr>
<td>2</td>
<td>19.5</td>
<td>2.3</td>
</tr>
<tr>
<td>3</td>
<td>19.5</td>
<td>2.2</td>
</tr>
<tr>
<td>4</td>
<td>19.5</td>
<td>2.8</td>
</tr>
</tbody>
</table>

Table 3: Times and distances recorded for mobility testing

Conclusion:
From these results it can be found that the unit meets the mobility requirements specified in the problem statement.

Section 3.2.2: Pin Tear Out
As mentioned in the durability parameters section, A Tinius Olsen machine was utilized to perform testing of the connecting links to verify sufficient resistance to pin tear out and verify calculations done in the previous semester. The overall setup of the testing experiment is shown in Figure 61.

The way that this machine works, is that you attach your test specimen in the upper and lower threaded holders in the left hand side of the Figure 61. Once the procedures given are carried out, the machine is then given a very small axial load on each end. Then the dial registers clockwise to a certain position depending on the axial load that is applied to the test specimen. Our test specimen is shown in Figure 62 and Figure 63. We took a steel test specimen that is used for tension tests in the Materials Lab, and cut it in half. Then we placed a notch into each side to allow for our linkage to slip inside. Two dowel pin holes were then drilled to allow for our dowel pins to connect the original threaded guide to our linkage arm.
We then taped where the dowel pin holes were located to ensure that the pins wouldn’t slip out during testing. The reason for the use of the dowel pins is to simulate how the linkage is mounted in the unit. From material specs of each piece, it was determined that the linkage arm would fracture before the dowel pins sheared. The specimen was placed into the axial loading area.

The test took approximately 15 minutes. After the specimen failed by fracture, the gage showed us that the ultimate load of the linkage arm was 1200 lbs and the fracture load was determined to be 800 lbs. Displayed below are some photos of the test specimen after fracture.
From our calculations from last semester, we determined an impact force of the shell to be 1114.29 lbs. In order to see if the linkages are going to tear out, we need to compare the force of tear out with the impact force. The original weight we intended the unit to consist of 5 lbs. The impact force based on that weight is as follows:

\[ F_{IMP} = m \cdot \frac{dv}{dt} = \frac{5(lbs)}{386.4 \left(\frac{in}{s^2}\right)} \cdot \frac{215.33(in)}{(s)} \cdot \frac{1}{.0025(s)} = 1114.29 \text{ lbs} \]

After weighting the whole unit, our actual weight increased to 5.3 lbs. The calculations were taken with the new weight and the impact force was found to be:

\[ F_{IMP} = m \cdot \frac{dv}{dt} = \frac{5.3(lbs)}{386.4 \left(\frac{in}{s^2}\right)} \cdot \frac{215.33(in)}{(s)} \cdot \frac{1}{.0025(s)} = 1181.41 \text{ lbs} \]

Since the tear out force was 1200 lbs, we can actually say that our linkages shouldn’t tear out. They will, however, after sometime get weaker and weaker if they are subjected to this kind of drop for multiple times which is not a likely scenario. That would weaken the material and eventually break them.

A suggestion for improvement is to go with a stronger steal, or even to increase the width of the linkages so the tear out area would be greater. This will allow for a greater force to be applied without damaging the linkages.
Section 3.2.3: Water Resistance Test
To ensure that the internal components stayed dry during operation in the rain, a water resistance test was performed in the fluid mechanics lab as shown in Figure 67.

Procedure:
All electronic components were removed and paper towels were placed inside so that any signs of water inside the shell would be indicated by discoloration of the paper towel. The access panels were installed and securely tightened by all eight 8-32 SHCS. Then, water was deliberately run over the sealing area to see if any water penetrated through to the inner surface.

Results:
After running the water over the sealing areas of the shell, the unit was inspected for any moisture inside through the clear access panels. The paper towels indicated that water had penetrated to the interior of the shell. The o-ring seals were inspected and it was determined that the water had not penetrated past the o-rings. Further inspection revealed that the water had penetrated through the 8-32 SHCS holes of the access panels. The test was then performed again with o-rings around the shank of each bolt and no water penetrated the interior of the shell.

Conclusion:
To resolve the issue of water penetrating to the interior of the shell through the 8-32 SHCS bolt holes, we fit small o-rings to around the shank of each bolt. With these o-rings in place, an appropriate seal will be made to eliminate any water penetration to the interior of the shell.

Section 3.2.4: PCB Strain Testing

To verify that the printed circuit board would not crack during operation the strain in the circuit board was tested.

Procedure:
Two strain gages were attached to the printed circuit board perpendicular to each other such that the strain in both directions could be measured. The circuit board and the mounting tray were loaded with torsional loads as if the motors operated at varied RPM’s, thus creating a torque on the tray and PCB. The test setup is shown in Figure 68.

Results:
At a torque of 6.56 in·lbs the resulting shear stress is 34.3 psi. This result is different from the analytical value most likely due to the material properties being used in the analysis. The material properties for the PCB used were not available and values for Polychlorinated Biphenyl, the most common material for PCBs, was used which has a yield stress of 8.5 ksi.

Conclusion:
From this data the PCB should not fail during normal usage.

**Section 3.2.5 Accelerometers**
The voltage output of each of the accelerometers feeds an input resistor which is connected to the inverting input of the operational amplifier. Analog op amp output to the microcontroller was tested using a Tektronix TDS3032 Two-Channel Digital Oscilloscope. In Figure 69 below, we see the output of both accelerometers when a given a strike on the side of the along the axis of the motor shafts. Figure 69 shows the axis of acceleration denoted by a red line and the placement of the accelerometers denoted by yellow circle and lines.

![Image of PCB](image)

**Figure 69:** Red arrow shows axis of acceleration for testing. Yellow lines show accelerometer placement

![Image of oscilloscope](image)

**Figure 70:** Op Amp Output of Accelerometers Feeding the Microcontroller
In Figure 70 above, the blue line represents the accelerometer perpendicular to the axis of acceleration. As expected, the blue line output is slightly higher than the accelerometer that is mounted parallel to the axis of rotation (yellow line). This is expected since the accelerometer mounted perpendicular to the line of strike will see a greater force than the parallel one. The moment of the strike is captured at the left side of the screen by the large swing in the blue line.

Section 3.2.6 Microphone

Testing of the microphone output to the microcontroller was performed in a similar fashion to the accelerometers, except that a sound input to the microphone in the form of human huming replaced a strike to the circuit board. A screen shot of the test results is shown in Figure 71 below.

![Figure 71: Output of microphone (yellow trace) and op amp circuit (blue trace)](image)

The yellow trace in Figure 71 represents the direct output of the microphone, and the blue trace represents the output of the operational amplifier circuit. The measured gain from the op amp circuit is:

\[ V_i = 4.12 \text{ mV} \quad V_o = 2.42 \text{ V} \quad A_i = V_o/V_i = 587, \]

Where \( V_i \) is the input voltage, \( V_o \) is the output voltage, and \( A_i \) is the inverting gain from the op amp. The measure gain of 587 is better than the nominal gain of 450.

Section 3.2.7 IR Remote – Transmitter and Receiver
The Infrared Remote had two sections to test: The output of the IR transmitter and the input as seen by the microcontroller. The IR transmitter was designed to send square wave frequencies of 8-, 16-, and 32-kHz. Initial testing revealed that the IR transmitter was capable of switching at all 3 frequencies, but the transistors of the receiving unit were not able to switch fast enough to interpret the 32 kHz frequency. Consequently, the IR remote was only tested at 8 kHz and 16 kHz. The remote was tested under a variety of conditions. Below is a screen shot of the testing of the remote at 8 kHz, measured at 4 ft.

![Figure 72: Output of the IR transmitter (yellow) and receiver transistor collector (blue), 8 kHz signal](image)

As Figure 72 shows, the output of the IR transmitter (shown in yellow, top trace) is very clean and stable with voltage peaking at 5 V. The frequency is measured at 9174 Hz. In contrast, the voltage from the photodiode as seen at the transistor collector (shown in blue, bottom trace) is quite weak and unstable with a maximum swing of 0.2 mV. The low voltage is one of the reasons that the amplifier circuit was redesigned to increase gain to roughly 10,000.

Following the op amp circuit, the voltage as seen prior to the transistor (yellow, bottom trace in Figure 73) and directly after the transistor (blue, top trace in Figure 73) section of the amplifier circuit, measured at a distance of 6 inches is measured as follows:
It can be seen in Figure 73 that prior to the op amp, there is a 30mV signal (yellow trace), and after the op amp it is nearly 3V (blue trace), representing a gain of 100 from the transistor.

After the signal passes through the op amp, the signal is amplified again, resulting in the signal shown in Figure 74 below:
Figure 74: IR remote transmitter (yellow) and op amp output (blue), 8 kHz signal at ~1 ft.

With the transmitter pointed nearly directly at the receiver, it can be seen that it is possible to achieve a very clean signal. The blue line in Figure 74 represents the signal that will be fed to the microcontroller.

When the transmitter is moved to a distance of 4 ft, the signal spends more time ramping up to the maximum voltage:
Now the IR unit is tested at the higher frequency of 16 kHz at 4 feet:

![Figure 76: IR remote transmitter (yellow) and op amp output (blue), 16 kHz signal at 4 feet.](image)

The time that the receiver spends as a portion of the period to ramp up to full voltage (blue line) is nearly 100%. In other words, the dwell time at full voltage is very small. The microcontroller is still able to recognize the higher frequency at this range, but one must be very accurate at pointing the transmitter directly at the phototransistor.

**Section 3.2.8 Motor PWM**
The Pulse Width Modulated output of the microcontroller to the motor was tested under two conditions to verify its performance. The first measurement below shows the output from the microcontroller when a 10 kΩ resistor is placed across the terminals of the H-bridge motor drive circuit. The measurement is made across the resistor:
It can be observed in Figure 77 that we are outputting a 50% duty cycle with a peak voltage of 10.1 V. This is as anticipated from programming and circuit calculations. Moreover, the frequency is 3650 Hz. The PWM frequency is calculated by taking the measured period (shown in Figure 77 as $\Delta$: 274 $\mu$s) and dividing this value into one:

$$T = 274 \, \mu$s \quad f = 1/ T \quad f = 3650 \, \text{Hz}$$

When the motors are added to the circuit, replacing the 10 k$\Omega$ resistor, the PWM signal is tested again. We observe the following results:
Figure 78: Motor PWM signal with motors as the load. The blue line represents the voltage across the motors; the yellow line is the output of the microcontroller.

It can be seen from the yellow trace that the signal from the microcontroller is 5 V as expected, with a very clean square wave. The voltage across the motors is nearly as clean, and peaks at 10 V as expected, but there is a curious spike during the dwell time. This spike did not result in errant running of the motors during testing, so one explanation could be that it is caused by the motor brushes releasing and reestablishing contact or another cause.

Section 3.2.9: Testing of A/D Conversions
Real-time sensing of our PIC16F737’s A/D conversions was not available to us, so to explain that the PIC16F737’s A/D conversion is successful, I will demonstrate some of the implemented code for reading IR and explain how it works. The process for reading and interpreting IR for turn-on is coded:

```c
void main()
{
    //Initializations/configurations: Begin:
    ADCON0=0x07;  //Configure analog inputs and Vref (digital input RB2
    ADCON1=0x87;  //for IR)
    TRISA = 0xFF; //PORTA is input
```
TRISB = 0x06; //Defines PORTB outputs with 0 and inputs with 1

//IR reading algorithm:
asm {
  BCF STATUS,5
  BCF STATUS,6
  BCF PORTB,5
  BCF PORTB,4 ;Make sure LEDs are off
  BCF PORTB,3
  BCF PORTB,0
}
asm {
  BSF STATUS,5
  BCF STATUS,6
  BCF OSCCON,6 ;Slow clock to check for IR signal
  NOP
  NOP
}

WaitForZero:
  BCF STATUS,5
  BCF STATUS,6
  BTFSC PORTA,4
  goto WaitForZero
  NOP
}

Bits = 1;
Zeros = 1;
Ones = 0;
asm {
  CheckBitAgain:
    NOP
    
    if (Bits == 80) goto InterpretFrequency;
    ...
    goto Playtime;
    ...

TRISA=0xFF; defines PORTA as input. IR input is at PORTA, pin #4. The PIC16F737 datasheet provides that PORTA is held at high voltage level (defined as >2V) until it receives a first low (zero sent by the IR). Under the WaitForZero: label is a BTFSC statement. What this does is skip its following instruction if a zero is sensed at PORTA, pin #4. If the pin is held at default high, program execution will see the goto WaitForZero command and wait until a zero is sensed before it moves on. That, we effectively see upon execution of the program. Once the ISPU is turned off (sent to the Begin: label), nothing happens until we press an IR button. The first zero sensed from IR breaks the program flow from that loop by skipping the goto WaitForZero command and going to NOP. Appropriately, thereafter in execution, we experience the motors turning on the Playtime algorithm.

Section 4: Evaluation and Recommendation

Section 4.1: Mechanical Evaluations and Recommendations
No matter how many things you plan for, nothing ever goes the way that you expect them to. As the group has seen, there have been many accommodations that had to be made in order to fit certain problems. Recommendations to help the outcome of the project are suggested as follows:
1) Clear Shell: The intended material of the outer shell was to be of a plastic of some sort that would be translucent. Since our prototype wasn’t the case, we found that we ran into some problems with turning the shell on and off when we wanted to. The IR remote couldn’t penetrate through the surface so the end caps had to be made specifically to meet the need of controlling the unit by remote.

2) Traction: We ran into some problems with the ball actually going through grass. Some of the problem comes from the lack of power drive from the motors, but from a mechanical standpoint, we could have helped the situation with a traction material on the outside of the shell. We originally thought that it would be a bad idea because it would be easier for the dog to get a hold of and destroy. Since we didn’t have it though, we didn’t get much range of motion in the grass like we had intended.

3) Ventilation: The interior electronics of the unit sometimes got the interior of the ball at a higher temperature than sought. We couldn’t figure out how we were going to let cool air in and hot air out without having opening for the dog to be able to grip their teeth into and possibly destroy the shell and also keep moisture out. With further study, I am sure that some type mechanism could be designed to fix that problem.

4) Down Scale: As a prototype, we found that our shell was somewhat on the large scale and a little heavier than intended. We also wanted to keep the shell large enough to not allow for the largest size dog breed to put it in their mouth. This has something to do with the large scale. But ultimately, the mass production of this product would have to be down sized if it were to be sold to a consumer and allow for some profit for the employer.

5) Different Type of Mobility: It is unclear how our design would compare to some of the other designs we thought of in our conceptual designs. The group made every possible consideration in our knowledge to select the best design. Sometimes the design that seems the best doesn’t always turn out to be that way. We suggest testing another design and seeing if better results are exposed.

Section 4.2: Electrical Hardware Evaluations and Recommendations

Section 4.2.1 Quiescent current optimization for remote control

The present design has a 8mA quiescent current. This current is insignificant for a non-battery operated (plugged into the wall) product, but for a battery operated product is almost unacceptable. The high capacity 3V Lithium batteries we are using have 300mAh capacity. Even if the remote just sits without being used the batteries will drain after:

\[
T = \frac{300mAh \cdot h}{8mA} = 38h
\]

And with use that number would drop even further. Having a disconnect switch is the first thing that comes to mind to solve that problem and is currently implemented. This solution needs to be improved since it is very impractical to have an on/off switch for a remote control. With more time we could design the circuits to consume less current while idling and therefore extend the battery life on the remote. A target for the battery life in our opinion should be approximately one year.
Section 4.2.2 Optimize LED Circuits
The LED configuration could be improved by placing all five of the LEDs of one color in series and connecting the anode of the first diode in the series directly to 9.6 V from the battery source rather than 5 V from the voltage regulator. This arrangement would reduce the current drawn from the voltage regulator during illumination, reducing the risk of overheating the regulator. Additionally, it would eliminate the condition in which the brightness of the LEDs is dictated by the diode with the smallest voltage drop. If one of the diodes has a smaller voltage drop across it, all diodes in parallel light less brightly. Most importantly, changing to a series arrangement would eliminate the 27 $\Omega$ resistor which wastes nearly 0.2 W of power.

Section 4.2.3 Add a Heat Sink to the 7805 Voltage Regulator
During repeated 100% duty cycle testing of the ISPU, the main source of heat on the circuit board was the voltage regulator. Although the unit has current and thermal protection, a heat sink can be easily added to the regulator via the hole in the upper body of the TO220 package. One point to note is that if the optimization of the LED circuit is implemented, the heat sink on the 7805 might be no longer necessary. The LEDs are the main source of current draw from the regulator in the current, so if they are connected directly to the battery pack, the draw on the regulator would be significantly reduced.

Section 4.2.4 Improve the Reliability of the Battery Pack
Towards the end of ISPU testing, it became apparent that the two battery packs purchased from Radio Shack to house our 8 AA batteries were the weak link in our system. On two occasions, the terminals failed to maintain contact with the batteries, causing the batteries to overheat significantly. On one occasion, the plastic housing on both batteries started to melt from the excessive heat. It is also believed that the high heat may have compromised the performance of the batteries, although that could not be quantitatively determined.

A potential solution is to have a battery pack constructed at a local merchant (e.g., Batteries Plus) that would be more robust. The pack would be wrapped in shrunk plastic, similar to the pack shown below.

![Robust battery pack suggested for ISPU](image_url)
The two differences from the pack in Figure 79 would be the use of high capacity (2500 mAh) batteries and a stacked configuration rather than an in-line configuration.

This service was not known until the writing of this report, so it could not be implemented. For the long-term success of this project, the use of a high quality battery pack is strongly recommended.

**Section 4.2.5 IR remote range**
Currently the maximum range of the remote control is around 4 ft. This range ideally should be extended to 10ft. This goal could be easily achieved if we had more time. The major factor in increasing the distance would be decreasing the frequency of oscillations. This can be done by adjusting the resistor values on the 555 timer in the remote control. This lower frequency would require exchanging the IR transmitter for a new one as the one that we currently use can not handle the lower frequencies at the current levels it is being run at. Decreasing the current would decrease the brightness and therefore the distance, which is the opposite of what the desired goal is. If the lower frequency requires less current through the IR transmitter a two LED design could be implemented, where the signal strength would be doubled by using two IR transmitters connected in parallel or driven from the same signal through separate driving gates.

**Section 4.3: Software Evaluations and Recommendations**
The project software evolved tremendously from the pseudocode created at the end of the Fall semester. There was noteworthy advancement and is room for improvement.

**Section 4.3.1: Noteworthy Software Advancement**
The introduction of Assembly Language to the midst of the C program allowed for greatest control over the PIC16F737. Perhaps the most significant contribution of the Assembly Language was its role in port configuration and definition. Because of the number of different, simultaneous processes that were involved in the final program, problems of register definition were encountered. Ports defined can override processes on multifunction pins. Assembly language allows individual bits to be set/cleared before implementing of each kind of process.

**Section 4.3.2: Room for Software Improvement**
The final software displays two characteristics that would be fixed with more time allotted to the Senior Design project: a release power-up and a misread of IR.

The ISPU turns on upon reset even though it is not programmed to. This is problematic because swiping a magnet across our reed switch was designed with the intention of turning the unit off. To fix the problem, the PIC16F737 datasheet would have to be read for characteristics upon reset and the software would have to be altered to set/reset interfering bits.

The logic behind reading IR input is wrongly defined and therefore reacts to whether either button was pressed for a long or short time as opposed to which button was pressed. The standing logic looks for a stream of 10001000 to detect 16kHz and a stream of 10000000 to detect 8kHz. What is actually seen is either 11001100 (for 16kHz) or 11110000 (for 8kHz). The
Standing logic compares quantities of ones and zeros received to determine action and requires almost complete renovation for the future development of the toy.

Section 4.3.3: Software Recommendation
The software programmer’s recommendations for PRI’s future development of the ISPU are:

1. Change the program to not power the toy on at reset. Research the PIC16F737 datasheet to determine register and port values upon reset and remedy those that are cause the motors and lights to come on at that time.
2. Revamp the IR reading logic. The standing logic looks for 10001000/10000000 to distinguish 16kHz and 8kHz as opposed to 11001100/11110000 as it should.
3. Program IR for response to more than two buttons as the IR control is developed by PRI’s professional capabilities.
4. Alter start/stop times per household appropriateness. 10-second drives and 5-second stops were requested for the play algorithm, however prototype testing within a tiled laboratory required those intervals compressed so the ISPU would not crash into walls.
Conclusion

The Interactive Sensor Package Unit was conceived to fill a market void as perceived by the project sponsor, Practical Robotic Innovations, LLC. The concept was to create a product platform that would intelligently interact with its environment through motion, light, and sound.

To that end, the designed ISPU receives sound input through a microphone and motion input via two orthogonally mounted accelerometers. The microcontroller receives these inputs, and adjusts sound, light, and motion output based on programmed algorithms.

The Interactive Sensor Package Unit meets the sponsor’s speed requirements for mobility on a smooth flat surface, has sound and light outputs that are within the auditory and visual range of its subject, and through selection of the PIC16F737 microcontroller provides a platform for future product expansion.
References

## Appendix A: Materials List

<table>
<thead>
<tr>
<th>Part</th>
<th>Retailer</th>
<th>Website</th>
<th>Cost</th>
<th>Quantity</th>
<th>Total</th>
<th>Retailer Part Number</th>
<th>Part Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dowel Pins</td>
<td>Grainger</td>
<td><a href="http://www.grainger.com">www.grainger.com</a></td>
<td>$0.09</td>
<td>100</td>
<td>$8.67</td>
<td>2MB33</td>
<td>Dowel Pin, pk 100</td>
</tr>
<tr>
<td>Retaining Rings</td>
<td>Barnhill Bolt Co.</td>
<td><a href="http://www.barnhillbolt.com">www.barnhillbolt.com</a></td>
<td>$0.05</td>
<td>100</td>
<td>$4.79</td>
<td>1285-18-A-1405-12</td>
<td>1/8 Retaining Ring, External PushOn, Steel, 1405-12</td>
</tr>
<tr>
<td>6-32 Screw</td>
<td>Grainger</td>
<td><a href="http://www.grainger.com">www.grainger.com</a></td>
<td>$0.05</td>
<td>100</td>
<td>$4.62</td>
<td>4X892</td>
<td>Socket Cap Screw, 4-40 x 3/8,PK100</td>
</tr>
<tr>
<td>8-32 Screw</td>
<td>Grainger</td>
<td><a href="http://www.grainger.com">www.grainger.com</a></td>
<td>$0.05</td>
<td>100</td>
<td>$5.19</td>
<td>4X612</td>
<td>Socket Cap Screw, 8-32 x 3/8,PK100</td>
</tr>
<tr>
<td>6-32 Nut</td>
<td>Grainger</td>
<td><a href="http://www.grainger.com">www.grainger.com</a></td>
<td>$0.02</td>
<td>100</td>
<td>$1.78</td>
<td>1F3A0</td>
<td>Locknut w/ Nylon Insert, 6.32PK,100</td>
</tr>
<tr>
<td>Threaded Insert</td>
<td>MSC</td>
<td><a href="http://www.msdirect.com">www.msdirect.com</a></td>
<td>$0.23</td>
<td>25</td>
<td>$5.88</td>
<td>8680217</td>
<td>Threaded Inserts For Wood Plastic</td>
</tr>
<tr>
<td>O-Ring</td>
<td>MSC</td>
<td><a href="http://www.msdirect.com">www.msdirect.com</a></td>
<td>$0.78</td>
<td>10</td>
<td>$2.88</td>
<td>75749150</td>
<td>Buna-N AS568A Aerospace Standard ID No. 256</td>
</tr>
<tr>
<td>Battery Holders</td>
<td>Walmart</td>
<td></td>
<td>$0.02</td>
<td>2</td>
<td>$0.40</td>
<td></td>
<td>Plastic 4 pack Battery Holders</td>
</tr>
<tr>
<td>Velcro</td>
<td>Walmart</td>
<td></td>
<td>$0.17</td>
<td>1</td>
<td>$0.37</td>
<td></td>
<td>Self Adhesive Velcro Strips</td>
</tr>
<tr>
<td>Duraf orm Shell</td>
<td>IPFW</td>
<td></td>
<td>$0.18</td>
<td>1</td>
<td>$0.18</td>
<td></td>
<td>Duraf orm Shell via SLS Rapid Prototype Machine</td>
</tr>
<tr>
<td>Motor</td>
<td>BaneBots</td>
<td><a href="http://www.trossenrobotics.com">www.trossenrobotics.com</a></td>
<td>$12.20</td>
<td>2</td>
<td>$24.40</td>
<td>MS-25030-370</td>
<td>30:1 5mm Spur Gearmotor</td>
</tr>
<tr>
<td>Transistors</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.21</td>
<td>10</td>
<td>$2.14</td>
<td>PN2222AGOS-ND</td>
<td>BJT, NPN, 2N222A</td>
</tr>
<tr>
<td>Capacitors</td>
<td>Digi-Key</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>$0.09</td>
<td>2</td>
<td>$0.18</td>
<td>604-1061-ND</td>
<td>CAP_ELECTROLIT, 10uf-POL</td>
</tr>
<tr>
<td>Resistors</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.69</td>
<td>4</td>
<td>$2.76</td>
<td>108126</td>
<td>RESISTOR, 100Ohm, 5%</td>
</tr>
<tr>
<td>Resistors</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.69</td>
<td>4</td>
<td>$2.76</td>
<td>108054</td>
<td>RESISTOR, 2.2kOhm 5%</td>
</tr>
<tr>
<td>Resistors</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>691358</td>
<td>RESISTOR, 10kOhm 5%</td>
</tr>
<tr>
<td>Capacitor</td>
<td>Digi-Key</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>$0.39</td>
<td>10</td>
<td>$3.94</td>
<td>ECI-3V1C564K</td>
<td>CAPACITOR, 560pF</td>
</tr>
<tr>
<td>Resistors</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>691041</td>
<td>RESISTOR, 5kOhm 5%</td>
</tr>
<tr>
<td>LEDs</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.17</td>
<td>10</td>
<td>$1.74</td>
<td>197616</td>
<td>LED_Red</td>
</tr>
<tr>
<td>LEDs</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.17</td>
<td>10</td>
<td>$1.74</td>
<td>156961</td>
<td>LED_Green</td>
</tr>
<tr>
<td>LEDs</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.17</td>
<td>10</td>
<td>$1.74</td>
<td>152792</td>
<td>LED_Yellow</td>
</tr>
<tr>
<td>Resistors</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>690486</td>
<td>RESISTOR, 270Ohm 5%</td>
</tr>
<tr>
<td>Capacitor</td>
<td>Digi-Key</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>$0.21</td>
<td>10</td>
<td>$2.07</td>
<td>2PC1815CT-ND</td>
<td>CAPACITOR, 180pF</td>
</tr>
<tr>
<td>Resistors</td>
<td>Digi-Key</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>$1.65</td>
<td>10</td>
<td>$16.50</td>
<td>RJ3W,0.1GCT-ND</td>
<td>RESISTOR, 0.1Ohm 5%</td>
</tr>
<tr>
<td>Resistors</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>690988</td>
<td>RESISTOR, 1kOhm 5%</td>
</tr>
<tr>
<td>Resistors</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>690996</td>
<td>RESISTOR, 3.6kOhm 5%</td>
</tr>
<tr>
<td>Resistors</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>690929</td>
<td>RESISTOR, 1.8kOhm 5%</td>
</tr>
<tr>
<td>Resistors</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>690871</td>
<td>RESISTOR, 160Ohm 5%</td>
</tr>
<tr>
<td>Resistors</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>690857</td>
<td>RESISTOR, 910Ohm 5%</td>
</tr>
<tr>
<td>LEDs</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.03</td>
<td>100</td>
<td>$3.03</td>
<td>661204</td>
<td>LED_Green</td>
</tr>
<tr>
<td>Transistors</td>
<td>Digi-Key</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>$0.80</td>
<td>2</td>
<td>$1.60</td>
<td>25B15920ACT-ND</td>
<td>BJT PNP, 25B1592</td>
</tr>
<tr>
<td>Transistors</td>
<td>Digi-Key</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>$0.52</td>
<td>2</td>
<td>$1.04</td>
<td>2SD2150T100RCT-ND</td>
<td>BJT_NPN, 2SD2150</td>
</tr>
<tr>
<td>Diode</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.32</td>
<td>10</td>
<td>$3.20</td>
<td>177754</td>
<td>DOIOE, 1N400</td>
</tr>
<tr>
<td>Resistor</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>690793</td>
<td>RESISTOR, 510Ohm 5%</td>
</tr>
<tr>
<td>Resistor</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>690347</td>
<td>RESISTOR, 7.5Ohm 5%</td>
</tr>
<tr>
<td>LED</td>
<td>Digi-Key</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>$0.66</td>
<td>1</td>
<td>$0.66</td>
<td>516-1263-ND</td>
<td>LED_J</td>
</tr>
<tr>
<td>Switch</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.23</td>
<td>3</td>
<td>$0.69</td>
<td>454504</td>
<td>SWITCH, SPST</td>
</tr>
<tr>
<td>JK Flip-Flops</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.83</td>
<td>1</td>
<td>$0.83</td>
<td>48039</td>
<td>74LS, 74L18</td>
</tr>
<tr>
<td>Resistor</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>691817</td>
<td>RESISTOR, 10kOhm 5%</td>
</tr>
<tr>
<td>Resistor</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.01</td>
<td>100</td>
<td>$1.00</td>
<td>691817</td>
<td>RESISTOR, 10kOhm 5%</td>
</tr>
<tr>
<td>Microphone</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.27</td>
<td>1</td>
<td>$0.27</td>
<td>160979</td>
<td>Electret Microphone</td>
</tr>
<tr>
<td>Oscillator</td>
<td>Jameco</td>
<td><a href="http://www.jameco.com">www.jameco.com</a></td>
<td>$0.23</td>
<td>1</td>
<td>$0.23</td>
<td>252543</td>
<td>Crystal Oscillator</td>
</tr>
<tr>
<td>Capacitors</td>
<td>Radio Shack</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>$1.59</td>
<td>3</td>
<td>$4.77</td>
<td></td>
<td>CAPACITOR, 1nF</td>
</tr>
<tr>
<td>Piezo Speaker</td>
<td>Radio Shack</td>
<td><a href="http://www.digikey.com">www.digikey.com</a></td>
<td>$2.59</td>
<td>1</td>
<td>$2.59</td>
<td></td>
<td>Piezo electric speaker</td>
</tr>
<tr>
<td>Op Amp</td>
<td>Mouser</td>
<td><a href="http://www.mouser.com">www.mouser.com</a></td>
<td>$0.35</td>
<td>1</td>
<td>$0.35</td>
<td>51-LM324AN</td>
<td>LM324 Quad Op Amp</td>
</tr>
<tr>
<td>Reed Switch</td>
<td>Hamlin Inc</td>
<td></td>
<td>-$</td>
<td>1</td>
<td>$-</td>
<td>-</td>
<td>Magnetic Reed Switch</td>
</tr>
<tr>
<td>Op Amp</td>
<td>Mouser</td>
<td><a href="http://www.mouser.com">www.mouser.com</a></td>
<td>$0.40</td>
<td>4</td>
<td>$1.60</td>
<td>51-LM7805CV</td>
<td>5V TA Op Amp</td>
</tr>
<tr>
<td>ICD</td>
<td>Mikroelektronika</td>
<td><a href="http://www.mikroe.com">www.mikroe.com</a></td>
<td>$89.99</td>
<td>1</td>
<td>$89.99</td>
<td></td>
<td>MikroElektronika ICD</td>
</tr>
<tr>
<td>ICD</td>
<td>MP Lab</td>
<td><a href="http://www.microchip.com">www.microchip.com</a></td>
<td>$160.00</td>
<td>1</td>
<td>$160.00</td>
<td></td>
<td>PIC ICD</td>
</tr>
<tr>
<td>C Compiler</td>
<td>Mikroelektronika</td>
<td><a href="http://www.mikroe.com">www.mikroe.com</a></td>
<td>$175.00</td>
<td>1</td>
<td>$175.00</td>
<td></td>
<td>MikroElektronika C-Compiler</td>
</tr>
</tbody>
</table>

Total $763.74

Note: The quantities ordered are the minimum order possible.
Appendix B: Programming Code
The following is the final code for the IPFW ISPU prototype (Assembly language in red, C in blue):

```
//Microcontroller: PIC16F737

unsigned int Bits=0;
unsigned int Zeros=0;
unsigned int Ones=0;
unsigned int Activity_var=1;
unsigned int Accel_1;
unsigned int Accel_2;
unsigned int i_Overload_1;
unsigned int i_Overload_2;
unsigned int mik;
unsigned int Pet_Monitor=10;
unsigned int Time1_time;
unsigned int Time2_time;
unsigned int JumpBackVar;
unsigned int Jump_Back_Var;

void main()
{

//Initializations/configurations:
Begin:
    ADCON0=0x07;
    ADCON1=0x87;    //Configure analog inputs and Vref (digital input RB2
    //for IR)
    TRISA = 0xFF;   //PORTA is input
    TRISB = 0x06;   //Defines PORTB outputs with 0 and inputs with 1

//IR reading algorithm:
    asm {
        BCF   STATUS,5
        BCF   STATUS,6
        BCF   PORTB,5
        BCF   PORTB,4    ;Make sure LEDs are off
        BCF   PORTB,3
        BCF   PORTB,0
    }
    asm {
        BSF   STATUS,5
        BCF   STATUS,6
        BCF   OSCCON,6    ;Slow clock to check for IR signal
        NOP
        NOP
    }

WaitForZero:
    BCF   STATUS,5
    BCF   STATUS,6
    BTFSC PORTA,4
    goto  WaitForZero
    NOP
}

Bits = 1;
```
Zeros = 1;
Ones = 0;
asm {
    CheckBitAgain:
        NOP
}
if (Bits == 80) goto InterpretFrequency;
asm {
    BCF      STATUS,5
    BCF      STATUS,6
    BTFSC    PORTA,4
    BTFSS    PORTA,4
    goto    ZerosPlusOne
    goto    OnesPlusOne
    NOP
    NOP
ZerosPlusOne:
    NOP
}
Zeros = Zeros+1;
Bits = Bits+1;
asm {
    NOP
    goto    CheckBitAgain
OnesPlusOne:
    NOP
}
Ones = Ones+1;
Bits = Bits+1;
asm {
    NOP
    goto    CheckBitAgain
    NOP
}
InterpretFrequency:
    if (Ones >15) {                //10001000 = 16kHz = Turn on
        Delay_ms(10);
        asm {
            BCF STATUS,5
            BCF STATUS,6
            BSF PORTB,4 ;Red
        }
         Delay_ms(10);
         asm {
            BCF STATUS,5
            BCF STATUS,6
            BCF PORTB,4
        }
        goto PlayTime;
    }
    if (Ones <15) {                //10000000 = 8kHz = toggle activity level
        asm {
            BCF STATUS,5
            BCF STATUS,6
            BSF PORTB,3 ;Green
        }
    }
Delay_ms(72);    //If user pushes wrong button to turn
    //unit on,
asm {
    BCF STATUS,5
    BCF STATUS,6
    BCF PORTB,3
}
goto Begin;      //wait 2.5 sec then wait for correct
    //button
}

PlayTime:
    do {
        asm {
            ;Forwards for 10 sec
            BSF STATUS,5 ;Reference page 92 of datasheet
            BCF STATUS,6
            MOVLW 0x01
            MOVWF PR2
            BCF STATUS,5
        }
        if (Activity_var==1) asm {
            ;50% duty
            MOVLW 0x01
            MOVWF CCPR1L
            BCF CCP1CON,5
            BSF CCP1CON,6
        }
        if (Activity_var==2) asm {
            ;23% duty
            MOVLW 0x00
            MOVWF CCPR1L
            BSF CCP1CON,5
            BSF CCP1CON,6
        }
        asm {
            BSF STATUS,5 ;CCPR1L=0x02, bits 6,5 of CCP1CON
            ;unimportant yields 100% duty
            BCF STATUS,6
            BCF TRISC,2 ;CCPR1L=0x01, bits 6,5 of CCP1CON=11
            ;yields 80% duty
            BSF TRISC,1
            BCF STATUS,5 ;CCPR1L=0x01, bits 6,5 of CCP1CON=10
            ;yields 50% duty
            BCF T2CON,1 ;CCPR1L=0x01, bits 6,5 of CCP1CON=01
            ;yields 77% duty
            BCF T2CON,0 ;CCPR1L=0x01, bits 6,5 of CCP1CON=00
            ;yields 50% duty
            NOP ;CCPR1L=0x00, bits 6,5 of CCP1CON=11
            ;yields 23% duty
            NOP ;CCPR1L=0x00, bits 6,5 of CCP1CON=10
            ;yields 0% duty
            MOVLW 0x04 ;CCPR1L=0x00, bits 6,5 of CCP1CON=01
            ;yields 29% duty
            MOVWF T2CON ;CCPR1L=0x00, bits 6,5 of CCP1CON=00
            ;yields 0% duty
            BSF CCP1CON,3
            BSF CCP1CON,2
            BSF STATUS,5
            BCF STATUS,6

90
BCF PORTB,4
BCF PORTB,3
BCF PORTB,0

BCF STATUS,5
BCF STATUS,6
BSF PORTB,5
BSF PORTB,4
BSF PORTB,3
BSF PORTB,0

Delay_ms(20);
asm {
    BCF STATUS,5
    BCF STATUS,6
    BCF PORTB,5
    BCF PORTB,4
    BCF PORTB,3
    BCF PORTB,0
}

Pet_Monitor = Pet_Monitor-1;
if (Pet_Monitor==0) goto Begin;
Jump_Back_Var = 1;
goto TimeGo;

End_of_TimeGo:
asm {
    BCF STATUS,5
    BCF STATUS,6
    CLRF CCPR1L
    BCF CCP1CON,5
    BCF CCP1CON,6
    BSF PORTB,5
    BCF PORTB,4
    BCF PORTB,3
    BCF PORTB,0
}

Delay_ms(10);
asm {
    BCF STATUS,5
    BCF STATUS,6
    BCF PORTB,5
}

JumpBackVar = 2;
goto TimeStop;

End_of_TimeStop:
asm {
    BSF STATUS,5
    BCF STATUS,6
    MOVLW 0x01
    MOVWF PR2
    BCF STATUS,5
}

if (Activity_var==1) asm {
    MOVLW 0x01
    MOVWF CCPR2L
}
if (Activity_var==2) asm {
    ;23% duty
    MOVLW 0x00
    MOVWF CCPR2L
    BSF CCP2CON,5
    BSF CCP2CON,6
}
asm {
    BSF STATUS,5 ;CCPR2L=0x00, bits 6,5 of CCP1CON
    ;unimportant yields 100% duty
    BCF STATUS,6
    BCF TRISC,1 ;CCPR2L=0x01, bits 6,5 of CCP2CON=11
    ;yields 80% duty
    BSF TRISC,2
    BCF STATUS,5 ;CCPR2L=0x01, bits 6,5 of CCP2CON=10
    ;yields 50% duty
    BCF T2CON,1 ;CCPR2L=0x01, bits 6,5 of CCP2CON=01
    ;yields 77% duty
    BCF T2CON,0 ;CCPR2L=0x01, bits 6,5 of CCP2CON=00
    ;yields 50% duty
    NOP ;CCPR2L=0x00, bits 6,5 of CCP2CON=11
    ;yields 23% duty
    NOP ;CCPR2L=0x00, bits 6,5 of CCP2CON=10
    ;yields 0% duty
    MOVLW 0x04 ;CCPR2L=0x00, bits 6,5 of CCP2CON=01
    ;yields 29% duty
    MOVWF T2CON ;CCPR2L=0x00, bits 6,5 of CCP2CON=00
    ;yields 0% duty
    BSF CCP2CON,3
    BSF CCP2CON,2
    BSF STATUS,5
    BCF STATUS,6
    BCF PORTB,4
    BCF PORTB,3
    BCF PORTB,0
    BCF STATUS,5
    BCF STATUS,6
    BSF PORTB,5
    BSF PORTB,4
    BSF PORTB,3
    BSF PORTB,0
}
Delay_ms(20);
asm {
    BCF STATUS,5
    BCF STATUS,6
    BCF PORTB,5
    BCF PORTB,4
    BCF PORTB,3
    BCF PORTB,0
}
Pet_Monitor = Pet_Monitor-1;
if (Pet_Monitor==0) goto Begin;
Jump_Back_Var = 3;
goto TimeGo;

End_of_TimeGo3:
    asm {
        BCF STATUS,5          ;Stop
        BCF STATUS,6
        CLRF CCPR2L
        BCF CCP2CON,5
        BCF CCP2CON,6
        BCF PORTB,5
        BCF PORTB,4
        BCF PORTB,3
        BCF PORTB,0
    }
    Delay_ms(10);
    asm {
        BCF STATUS,5
        BCF STATUS,6
        BCF PORTB,5
    }
    JumpBackVar = 4;
goto TimeStop;

End_of_TimeStop4:
    Delay_ms(10);
}
while (1);

TimeGo:
    Timel_time=100;
Loop1:
    Timel_time = Timel_time-1;
    if (Timel_time == 0 && Jump_Back_Var ==1) goto End_of_TimeGo1;
    if (Timel_time == 0 && Jump_Back_Var ==3) goto End_of_TimeGo3;
    ADCON1=0x80;  //Configure analog inputs and Vref. Analog input setup
                  //step #1
    ADCON2=0x08;

    asm {  
        BCF TRISB,4
        BCF TRISB,3
        BCF TRISB,0
        BSF PORTB,4
        BSF PORTB,3
        BSF PORTB,0
    }
    ADCON0=0x11;
    ADCON0=0x15;
    i_Overload_1 = Adc_Read(2);
    if (i_Overload_1>6) {
        i_Overload_1 = Adc_Read(2);
        if (i_Overload_1>6) {
            i_Overload_1 = Adc_Read(2);
            if (i_Overload_1>6) {

93
asm {
    CLRF CCPR1L
    BCF CCP1CON,5
    BCF CCP1CON,6
    CLRF CCPR2L
    BCF CCP2CON,5
    BCF CCP2CON,6
}
goto Begin;
}

ADCON0=0x19;
ADCON0=0x1D;
i_Overload_2 = Adc_Read(3);
if (i_Overload_2>6) {
    i_Overload_2 = Adc_Read(3);
    if (i_Overload_2>6) {
        i_Overload_2 = Adc_Read(3);
        if (i_Overload_2>6) {
            asm {
                CLRF CCPR1L
                BCF CCP1CON,5
                BCF CCP1CON,6
                CLRF CCPR2L
                BCF CCP2CON,5
                BCF CCP2CON,6
            }
goto Begin;
        }
    }
}

ADCON0=0x07;
ADCON1=0x87;    //Configure analog inputs andVref (digital input RB2 for
//IR)
goto Loop1;

TimeStop:
    Time2_time=50;
    asm {
        BSF STATUS,5
        BCF STATUS,6
        BCF OSCCON,6 ;Slow clock to check for IR signal
        NOP
        NOP
    }

WaitForZero2:
    NOP
    }
    Time2_time = Time2_time-1;
    if (Time2_time == 0 && JumpBackVar == 2) goto End_of_TimeStop2;
    if (Time2_time == 0 && JumpBackVar == 4) goto End_of_TimeStop4;

ADCON1=0x80;  //Configure analog inputs and Vref. Analog input setup
             //step #1
ADCON2=0x08;
ADCON0=0x01; //Get results of AD conversions, 10-bit each
ADCON0=0x05; //Turns on A/D, then converts value
Accel_1 = Adc_Read(0);
if (Accel_1>570 || Accel_1<470) {
    asm {
        BSF PORTB,4
    } //Red
    asm {
        BCF PORTB,4
    }

    Pet_Monitor=10;
    asm {
        BSF PORTB,3
    } //Green
    asm {
        BCF PORTB,3
    }

    asm {
        CLRF CCPR1L
        BCF CCP1CON,5
        BCF CCP1CON,6
        CLRF CCPR2L
        BCF CCP2CON,5
        BCF CCP2CON,6
    }
    asm {
        CLRF CCPR1L
        BCF CCP1CON,5
        BCF CCP1CON,6
        CLRF CCPR2L
        BCF CCP2CON,5
        BCF CCP2CON,6
    }

    asm {
        BSF PORTB,0
    } asm {
        BCF PORTB,0
    }

    //Instead of being centered on 512,
    //...these values are centered on 520
    //...because of circuit imperfection

    //Configure analog inputs and Vref (digital input RB2
    //for IR)
    asm {
        BCF STATUS,5
        BCF STATUS,6
        BTFSC PORTA,4
        goto WaitForZero2
        NOP
    }

    asm {
        CheckBitAgain2:
    }
asm {
  BCF STATUS,5
  BCF STATUS,6
  BTFSC PORTA,4
  BTFSS PORTA,4
  goto ZerosPlusOne2
  goto OnesPlusOne2
  NOP
  NOP
ZerosPlusOne2:
  NOP
  }
asm {
  NOP
  goto CheckBitAgain2
OnesPlusOne2:
  NOP
  }
asm {
  NOP
  goto CheckBitAgain2
  NOP
  }
  //10001000 = 16kHz = Turn off
  asm {
    BSF STATUS,5
    BCF STATUS,6
    BCF TRISB,4
    BCF STATUS,5
    BCF STATUS,6
    BSF PORTB,4 ;Red
    NOP
  }
asm {
    BSF STATUS,5
    BCF STATUS,6
    BSF OSCCON,6 ; Put clock back
                   ; at 1MHz for play
    NOP
    NOP
}

Pet_Monitor=10;
else
    ADCON0=0x09;
    ADCON0=0x0D;
    Accel_2 = Adc_Read(1);
    if (Accel_2>570 || Accel_2<470) {
    }
else
    ADCON0=0x11;
    ADCON0=0x15;
    i_Overload_1 = Adc_Read(2);
    if (i_Overload_1>6) {
        i_Overload_1 = Adc_Read(2);
        if (i_Overload_1>6) {
            i_Overload_1 = Adc_Read(2);
            if (i_Overload_1>6) {
                goto Begin;
            }
        }
    }
    ADCON0=0x19;
    ADCON0=0x1D;
    i_Overload_2 = Adc_Read(3);
    if (i_Overload_2>6) {
        i_Overload_2 = Adc_Read(3);
        if (i_Overload_2>6) {
            i_Overload_2 = Adc_Read(3);
            if (i_Overload_2>6) {
                goto Begin;
            }
        }
    }
    ADCON0=0x21;
    ADCON0=0x25;
    mik = Adc_Read(4);
    if (mik>570 || mik<470) {
        Pet_Monitor=10;
        // Yellow
    }
else
    ADCON0=0x07;
    ADCON1=0x87;
    Bits = 1;
    Zeros = 1;
Ones = 0;
if (Bits == 80) goto InterpretFrequency2;
    Zeros = Zeros+1;
    Bits = Bits+1;
Ones = Ones+1;
    Bits = Bits+1;
InterpretFrequency2:
    if (Ones > 15) {
    _ms(10);
    Delay
    Delay_ms(100);
    goto Begin;
    }
    if (Ones < 15) {
    Delay_ms(10);
    Delay_ms(10);
    if (Activity_var==1) {
    Activity_var=2;
    End_of_TimeStop2;
    if (JumpBackVar == 4) goto
    End_of_TimeStop4;
    }
    if (Activity_var==2) {
    if (JumpBackVar == 2) goto
    Activity_var=1;
    asm {
    BSF   STATUS,5
    BCF   STATUS,6
    BSF   OSCCON,6 ;Put clock back
    ;at 1MHz for play
    NOP
    NOP
    }
    if (JumpBackVar == 2) goto
    End_of_TimeStop2;
    if (JumpBackVar == 4) goto
    End_of_TimeStop4;
    }
    }
Appendix C: Summary of primary electronic components’ specifications

PIC16F7X7
28/40/44-Pin, 8-Bit CMOS Flash Microcontrollers with 10-Bit A/D and nanoWatt Technology

Low-Power Features:
- Power Managed modes:
  - Primary Run (XT, RC oscillator, 75 µA, 1 MHz, 2V)
  - RC_RUN (7 µA, 31.25 kHz, 2V)
  - SEC_RUN (9 µA, 32 kHz, 2V)
  - Sleep (0.1 µA, 2V)
- Timer1 Oscillator (1.8 µA, 32 kHz, 2V)
- Watchdog Timer (0.7 µA, 2V)
- Two-Speed Oscillator Start-up

Oscillators:
- Three Crystal modes:
  - LP, XT, HS (up to 20 MHz)
- Two External RC modes
- One External Clock mode:
  - ECIO (up to 20 MHz)
- Internal Oscillator Block:
  - 8 user-selectable frequencies (31 kHz, 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, 8 MHz)

Analog Features:
- 10-bit, up to 14-channel Analog-to-Digital Converter:
  - Programmable Acquisition Time
  - Conversion available during Sleep mode
- Dual Analog Comparators
- Programmable Low Current Brown-out Reset (BOR) Circuitry and Programmable Low-Voltage Detect (LVD)

Peripheral Features:
- High Sink/Source Current: 25 mA
- Two 8-bit Timers with Prescaler
- Timer1/RTC module:
  - 16-bit timer/counter with prescaler
  - Can be incremented during Sleep via external 32 kHz watch crystal
- Master Synchronous Serial Port (MSSP) with 3-wire SPI™ and I²C™ (Master and Slave) modes
- Addressable Universal Synchronous Asynchronous Receiver Transmitter (AUSART)
- Three Capture, Compare, PWM modules:
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10 bits
- Parallel Slave Port (PSP) – 40/44-pin devices only

Special Microcontroller Features:
- Fail-Safe Clock Monitor for protecting critical applications against crystal failure
- Two-Speed Start-up mode for immediate code execution
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Programmable Code Protection
- Processor Read Access to Program Memory
- Power Saving Sleep mode
- In-Circuit Serial Programming™ (ICSP™) via two pins
- MPLAB® In-Circuit Debug (ICD) via two pins
- MCLR pin function replaceable with input only pin

<table>
<thead>
<tr>
<th>Device</th>
<th>Program Memory (# Single-Word Instructions)</th>
<th>Data SRAM (Bytes)</th>
<th>I/O</th>
<th>10-bit A/D (ch)</th>
<th>CCP (PWM)</th>
<th>MSSP</th>
<th>AUSART</th>
<th>Timers 8/16-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F737</td>
<td>4096</td>
<td>368</td>
<td>25</td>
<td>18</td>
<td>11</td>
<td>2</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>PIC16F747</td>
<td>4096</td>
<td>368</td>
<td>36</td>
<td>17</td>
<td>14</td>
<td>2</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>PIC16F767</td>
<td>8192</td>
<td>368</td>
<td>25</td>
<td>16</td>
<td>11</td>
<td>2</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>PIC16F777</td>
<td>8192</td>
<td>368</td>
<td>36</td>
<td>17</td>
<td>14</td>
<td>2</td>
<td>3</td>
<td>Yes</td>
</tr>
</tbody>
</table>
LOW POWER QUAD OPERATIONAL AMPLIFIERS

- WIDE GAIN BANDWIDTH : 1.3MHz
- LARGE VOLTAGE GAIN : 100dB
- VERY LOW SUPPLY CURRENT/AMPLI : 375μA
- LOW INPUT BIAS CURRENT : 20nA
- LOW INPUT OFFSET VOLTAGE : 3mV max.
- LOW INPUT OFFSET CURRENT : 2nA
- WIDE POWER SUPPLY RANGE:
  SINGLE SUPPLY : +3V TO +30V
  DUAL SUPPLIES : ±1.5V TO ±15V
- INPUT COMMON-MODE VOLTAGE RANGE
  INCLUDES GROUND
- ESD INTERNAL PROTECTION : 2kV

DESCRIPTION

These circuits consist of four independent, high gain, internally frequency compensated operational amplifiers. They operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

All the pins are protected against electrostatic discharges up to 2000V (as a consequence, the input voltages must not exceed the magnitude of $V_{CC}^+$ or $V_{CC}^-$.)

PIN CONNECTIONS (top view)

ORDER CODE

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Temperature Range</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM124A</td>
<td>-55°C, +125°C</td>
<td>N D P</td>
</tr>
<tr>
<td>LM224A</td>
<td>-40°C, +105°C</td>
<td>N D P</td>
</tr>
<tr>
<td>LM324A</td>
<td>0°C, +70°C</td>
<td>N D P</td>
</tr>
</tbody>
</table>

Example : LM224AN

N = DIP = Dual in Line Package
D = SO14 = Small Outline Package (SO) - also available in Tape & Reel (TR)
P = TSSOP14 = Thin Shrink Small Outline Package (TSSOP) - only available in Tape & Reel (TR)
L7800 SERIES

POSITIVE VOLTAGE REGULATORS

- OUTPUT CURRENT TO 1.5A
- OUTPUT VOLTAGES OF 5, 5.2, 6, 8, 8.5, 9, 10, 12, 15, 18, 24V
- THERMAL OVERLOAD PROTECTION
- SHORT CIRCUIT PROTECTION
- OUTPUT TRANSITION SOA PROTECTION

DESCRIPTION
The L7800 series of three-terminal positive regulators is available in TO-220, TO-220FP, TO-220FM, TO-3 and D²PAK packages and several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltage and currents.

Figure 1: Schematic Diagram

November 2004
L7800 SERIES

Table 1: Absolute Maximum Ratings

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V_i$</td>
<td>DC Input Voltage</td>
<td>for $V_o = 5$ to $18V$</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for $V_o = 20$, $24V$</td>
<td>40</td>
</tr>
<tr>
<td>$I_o$</td>
<td>Output Current</td>
<td>Internally Limited</td>
<td></td>
</tr>
<tr>
<td>$P_{tot}$</td>
<td>Power Dissipation</td>
<td>Internally Limited</td>
<td></td>
</tr>
<tr>
<td>$T_{slg}$</td>
<td>Storage Temperature Range</td>
<td>-65 to 150 °C</td>
<td></td>
</tr>
<tr>
<td>$T_{op}$</td>
<td>Operating Junction Temperature Range</td>
<td>for L7800</td>
<td>-55 to 150 °C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for L7800C</td>
<td>0 to 150 °C</td>
</tr>
</tbody>
</table>

Absolute Maximum Ratings are those values beyond which damage to the device may occur. Functional operation under these condition is not implied.

Table 2: Thermal Data

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>D²PAK</th>
<th>TO-220</th>
<th>TO-220FP</th>
<th>TO-220FM</th>
<th>TO-3</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{th-case}$</td>
<td>Thermal Resistance Junction-case Max</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>°C/W</td>
</tr>
<tr>
<td>$R_{th-amb}$</td>
<td>Thermal Resistance Junction-ambient Max</td>
<td>62.5</td>
<td>50</td>
<td>60</td>
<td>60</td>
<td>35</td>
<td>°C/W</td>
</tr>
</tbody>
</table>

Figure 2: Schematic Diagram
Figure 3: Connection Diagram (top view)

Table 3: Order Codes

<table>
<thead>
<tr>
<th>TYPE</th>
<th>TO-220 (A Type)</th>
<th>TO-220 (C Type)</th>
<th>TO-220 (E Type)</th>
<th>D^2PAK (A Type) (*)</th>
<th>D^2PAK (C Type) (T &amp; R)</th>
<th>TO-220FP</th>
<th>TO-220FM</th>
<th>TO-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>L7805</td>
<td>L7805C</td>
<td>L7805CV</td>
<td>L7805CV1</td>
<td>L7805CD2T</td>
<td>L7805C-D2TR</td>
<td>L7805CP</td>
<td>L7805CF</td>
<td>L7805CT</td>
</tr>
<tr>
<td>L7806C</td>
<td>L7806CV</td>
<td>L7806C-V</td>
<td>L7806CD2T</td>
<td>L7806CP</td>
<td>L7806CF</td>
<td>L7806CT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L7808</td>
<td>L7808C</td>
<td>L7808CV</td>
<td>L7808C-V</td>
<td>L7808CD2T</td>
<td>L7808CP</td>
<td>L7808CF</td>
<td>L7808CT</td>
<td></td>
</tr>
<tr>
<td>L7809C</td>
<td>L7809CV</td>
<td>L7809C-V</td>
<td>L7809CD2T</td>
<td>L7809CP</td>
<td>L7809CF</td>
<td>L7809CT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L7810C</td>
<td>L7810CV</td>
<td>L7810CD2T</td>
<td>L7810CP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>L7812T</td>
</tr>
<tr>
<td>L7812</td>
<td>L7812CV</td>
<td>L7812C-V</td>
<td>L7812CD2T</td>
<td>L7812CP</td>
<td>L7812CF</td>
<td>L7812CT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L7815</td>
<td>L7815C</td>
<td>L7815CV</td>
<td>L7815CD2T</td>
<td>L7815CP</td>
<td>L7815CF</td>
<td>L7815CT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L7818</td>
<td>L7818C</td>
<td>L7818CV</td>
<td>L7818CD2T</td>
<td>L7818CP</td>
<td>L7818CF</td>
<td>L7818CT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L7820C</td>
<td>L7820CV</td>
<td>L7820CD2T</td>
<td>L7820CP</td>
<td>L7820CF</td>
<td>L7820CT</td>
<td>L7820CT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L7824</td>
<td>L7824CV</td>
<td>L7824CD2T</td>
<td>L7824CP</td>
<td>L7824CF</td>
<td>L7824CT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(*) Available in Tape & Reel with the suffix "-TR".
**LM555 Timer**

**General Description**
The LM555 is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For astable operation as an oscillator, the free running frequency and duty cycle are accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output circuit can source or sink up to 200 mA or drive TTL circuits.

**Features**
- Direct replacement for NE555
- Timing from microseconds through hours
- Operates in both astable and monostable modes
- Adjustable duty cycle
- Output can source or sink 200 mA
- Output and supply TTL compatible
- Temperature stability better than 0.005% per °C
- Normally on and normally off output
- Available in 8-pin MSOP package

**Applications**
- Precision timing
- Pulse generation
- Sequential timing
- Time delay generation
- Pulse width modulation
- Pulse position modulation
- Linear ramp generator

**Schematic Diagram**

![Schematic Diagram of LM555 Timer](image)
Connection Diagram

Dual-In-Line, Small Outline and Molded Mini Small Outline Packages

GND 1
TRIGGER 2
OUTPUT 3
RESET 4

+Vcc 8
DISCHARGE 7
THRESHOLD 6
CONTROL VOLTAGE 5

Top View

Ordering Information

<table>
<thead>
<tr>
<th>Package</th>
<th>Part Number</th>
<th>Package Marking</th>
<th>Media Transport</th>
<th>NSC Drawing</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-Pin SOIC</td>
<td>LM555CM</td>
<td>LM555CM</td>
<td>Rails</td>
<td>M08A</td>
</tr>
<tr>
<td></td>
<td>LM555CMX</td>
<td>LM555CM</td>
<td>2.5k Units Tape and Reel</td>
<td></td>
</tr>
<tr>
<td>8-Pin MSOP</td>
<td>LM555CMMM</td>
<td>Z55</td>
<td>1k Units Tape and Reel</td>
<td>MUA08A</td>
</tr>
<tr>
<td></td>
<td>LM555CMMMX</td>
<td>Z55</td>
<td>3.5k Units Tape and Reel</td>
<td></td>
</tr>
<tr>
<td>8-Pin MDIP</td>
<td>LM555CN</td>
<td>LM555CN</td>
<td>Rails</td>
<td>N08E</td>
</tr>
</tbody>
</table>