CONTENTS

I. Abstract

II. Introduction
   A. Background
   B. Competing Products

III. Design Options
   A. Design I: Equipment and Analysis
   B. Design II: Equipment and Analysis
   C. Design III: Equipment and Analysis

IV. Optimal Design
   A. Overview
   B. Method
   C. Views
      i. Top View (Internal)
      ii. Side View (Internal)
      iii. User Interface Panel (External)
      iv. Back View (External)
      v. Front View (External)
      vi. Bottom View (External)
      vii. Overall Flow Chart
   D. Equipment and Analysis
      i. Microprocessor
      ii. Driver
      iii. Linear Actuator
      iv. Syringe Calculations
      v. LabVIEW™ Simulation

V. Realistic Constraints and Safety Issues

VI. Technical Specifications

VII. Timeline

VIII. Total Cost

IX. Conclusions

X. References

XI. Acknowledgements
I. ABSTRACT

In recent years, many independent adults have indicated a need for a method of accurately controlling the doses of their intravenous medications. Millions of people rely on self-medicating techniques that require the use of syringes. Those suffering from type II diabetes, and therefore are insulin dependent, or stroke patients, who use precautionary heparin injections, are some of the patients who most commonly use syringes. Self-dosing can be problematic for many people, especially those who are elderly, visually impaired, or hearing impaired, or suffer from arthritis, Parkinson’s disease, partial paralysis, or the loss of motor skills due to a stroke, heart attack, or other physical ailments. Clearly, there is a need for a reliable, easy-to-use, and inexpensive product to accurately fill syringes with insulin or heparin in a timely manner. Products currently on the market require patients to mechanically fill their syringes, using their fine motor skills to control the syringe mechanism.

Products currently on the market require patients to mechanically fill their syringes, using their fine motor skills to control the syringe mechanism. This method increases the risk of errant dosing by relying on the patient’s physical ability to perform the dosing correctly. The current products, therefore, do not accommodate patients that lack the fine motor skills necessary for correct use. The product described here will provide a digital self-dosing device that will accommodate many of the physical limitations mentioned above, while remaining affordable and competitive in today’s market.

II. INTRODUCTION

This new product has several important basic requirements. It needs to be accessible for people who are hearing impaired, vision impaired, and who lack certain motor functions. Because most of the products already on the market accommodate those who are hearing impaired and vision impaired, the feature that will set this product apart from the others is that it will not require fine motor skills for operation. It must, of course, accurately dose the user’s medication to the nearest 0.01cc. The accessible syringe dosing device must also, as its title implies, be compact, easy-to-use, and cost effective. In order to design such a device, the team of project engineers will employ their knowledge in the areas of control, digital electronics, computer programming, instrumentation, and other essential branches of traditional engineering in addition to the application of the principles of biomedical engineering.

The principal clientele for this project as established by the Rehabilitation Engineering Research Center on Accessible Medical Instrumentation are Lloyd, Sophia, Arnold, Dave, and Wanda, and her father Bob. These imaginary clients are designed to emulate the characteristics and conditions that a typical client of our product would exhibit. Each client is different, possessing unique needs that add to the general product stipulation. The following is a summary of the traits that we feel are most important when considering our design.

Lloyd suffers from type II diabetes and therefore is interested in our product as a convenient method for dosing the syringes used for her insulin injections. She already
has poor eyesight and is nearing 80 years old. In the coming years, her eyesight, hearing, and motor skills may further diminish. In terms of our design, this stipulates that our product should accommodate those who are elderly and who have poor vision. The product must be easy enough to use while not relying on sight to determine the dosing.

Sophia, also around 80 years of age, is a victim of one large stroke and several smaller ones. Because of the risk of recurrence, she now uses heparin injections and therefore requires an accessible dosing mechanism. She has limited use of her right arm and therefore may have a hard time operating a device that requires either fine motor coordination or the use of both hands. The product should therefore accommodate these limitations by limiting, or perhaps eliminating, the need for fine motor functions.

Arnold is also a victim of diabetes and therefore requires insulin injections. He is about 50 years of age, and his fine motor functions are handicapped by the tremors that result from Parkinson’s disease. This once again advocates the need for a dosing device that does not require fine motor skills to operate.

Dave also suffers from type II diabetes and has limited leg mobility. The device should therefore be easily portable to add to its convenience and thus minimize necessary movement.

Wanda also suffers from diabetes. She is only ten years old and despite being deaf is being encouraged to start self-administering her insulin injections. Her father, Bob, is her sole caretaker and is responsible for teaching her how to administer these injections; however, he is limited by blindness. The device should therefore once again accommodate patients with loss of sight as well as those will loss of hearing.

Based on the needs of our clients, our product must accommodate sight and hearing impairments, must be functional even for those with limited motor functions and mobility, and must be easy to use by people of all ages.

The following is a compilation of several of the principal competitors currently on the market. For our design it is necessary to note the good points of such products as well as the shortcomings upon which we can improve. The retail price of each is also an important figure, as it dictates the budget within which we are to produce a competitive product. Also, any additional information about these products that we feel necessary when considering our own design is included below.

Count-A-Dose by Medicol: This product priced at $59.95 is not only the most popular one on the market, but is also the one that best satisfies our requirements. It is lightweight, portable, and easy to use. It accommodates sight and hearing impaired patients by using a clicking sound that can be felt and counts the dosing units. It even has a two-insulin bottle holder at its base for easy mixing. The product is visually appealing and includes a tape cassette with instructions. The product, however, can be improved upon. It only adds to the syringe in 1-unit increments and therefore can be a hassle while dosing. Furthermore, it still relies on a patient’s ability to use fine motor functions in operating the device. It also does not aim to minimize human error in dosing amounts. There is no definite indicator of the dosing on the device; rather it relies on the client’s ability to count the clicks. This may be problematic with patients with bad memories or attention disorders. Our device should therefore embody the Count-A-Dose’s basic design while increasing its functionality to encompass the stated shortcomings.

Load-Matic by Palco Labs: This product is priced at $49.95 and is very similar in appearance and aesthetics to the aforementioned Count-A-Dose. The design, however, is
far more complex; the Load-Matic includes increment settings of either 1 unit at a time or 10 units at a time, using a movable operating drill gear. Although this does have some advantages, it also leads to more error in dosing. The 10-unit lever is not easy to fully depress, especially for clients with limited motor functions, and therefore under dosing is a recurring problem. Also, like the Count-A-Dose this device fails to minimize the human mechanical interactions and therefore needlessly increasing the risk of error. Our device should continue with the Load-Matic’s trend toward a quicker dosing mechanism but at the same time minimize the devices complexity and the human mechanical interactions.

The Syringe Support by the Foundation Center Louise-Herbert in Canada: At $19.95, this product is perhaps best noted for its cost effectiveness. However, there is a definite detriment to its lower price. Its design lacks both audible and tactile indicators of dosing units. It also relies on a turn screw mechanism to draw the syringe back. This is a disadvantage because the patient must count, without indication, the number of turns the screw has been twisted; this may prove especially difficult for clients with limited motor functions. Furthermore, the device does not have an easy mechanism for mixing; the vial must be disengaged and then a new one fixed in place every time. Although this is a competitive product in terms of pricing, it does not meet product expectations in terms of accommodating disabilities, and is sizably susceptible to error (all information from http://www.nfb.org/vodold/inslmeas.htm, Insulin Measurement Devices).

III. DESIGN OPTIONS

During the course of the design process, three varying designs have been constructed to meet the constraints of this device. The equipment and analysis portions of the preliminary designs are included below, and a full description of the optimal design follows.

A. DESIGN I

Design I features a gear system to mechanically draw the syringe, and a simple logic circuit to process the user’s request and translate it into a given number of pulsations, each one corresponding to 0.01cc of medication. Several views of this design are shown below.
Syringe Dosing Circuit

DC Motor

INT

Free Sliding Track

FRONT S
The LDC display pictured above is the display that will be incorporated into the design.
Motor Selection:

- Note that this specific 6-pole rotor turns in a direction opposite the rotation of the stator field, a two-pole rotor inside the exact same field would rotate in the same direction.

- This illustration is based on half-step control, where alternate half steps involve one and two motor windings.

- It takes three complete cycles of the control system to turn this 6-pole rotor one revolution. A two-pole rotor would turn a full revolution per control system cycle.

Source: [http://www.engr.uconn.edu/cse/Courses/CSE210W/](http://www.engr.uconn.edu/cse/Courses/CSE210W/)

Of the various types of stepping motors we have chosen a unipolar for our design. The basic design and specifications are depicted to the right. Note that the main advantage of using a unipolar motor over a variable reluctance motor lies in the number of control signals required. As depicted below, the basic unipolar control circuit outputs to four winding control switches (for two windings) whereas the standard variable reluctance
motor outputs to three control switches (one for each of the three windings). Four control signals is much easier to control using a binary based controlled circuit than three control circuits simply because of the ease of converting from a base two to a base four number. In terms of our actual circuitry this means that we can easily distribute a given number of pulses between our four switches using a simple 8-bit counter unit (see the actual circuit diagram and accompanying materials for details). Besides the ease of designing a control circuit there are several other reasons why the unipolar motor was chosen for our design:

1. The unipolar motors are far cheaper than variable reluctance and bipolar motors.
2. Permanent magnet motors (in general) have a higher torque output per power input ratio than the variable reluctance motors.
3. As discusses earlier, by varying the number of poles (2, 4, 6 etc.) one can control the minimum step size (90°, 45°, 30° etc). \( S_{\text{min}} = 180/P \). Note that our current selection is a 10 polar, 18° motor.

**Control Circuit:**

The basic schematic for our control circuit is depicted below. The circuit inputs an 8-bit binary number corresponding to the number of units (1 unit = .01 cc) desired. It then outputs logical pulses to the four control switches in the driver circuit as specified previously. Since the minimum step size of the motor (18°) has been designed (through the gear system) to coincide with a 1 unit intake (see gear transfer section) the total number of pulses distributed is equal to the inputted binary number. This circuit is useful...
in minimizing the cost of our overall design by eliminating the need for a microprocessor. Our final design, however, may very well include a microprocessor which will likely replace some (or all) of this hardware.
The circuit functions as follows:
1. The first 8-bit counter hooked up to a basic 555 chip clock or the microprocessor inputs the requested number of units.
2. The counter enable input is directly linked to the “nor” value of the counter outputs (1 if all 0s). Thus the counter counts down from the requested value to 0 and stops.
3. A 3-way “and” gate is used to limit the total number of clock pulses passed to the second counter to the number requested (the number of pulses used to get the counter to 0).
4. The second counter counts up to the desired dosage (a safety measure) and then distributes the pulses to the 4 control switches. It does so using the principles of binary notation (see truth table) and four 2-way “and” gate.
5. The truth table for step 4 which determines the configuration of the and gates:

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q0</th>
<th>1a</th>
<th>1b</th>
<th>2a</th>
<th>2b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

6. The overall circuit is controlled by the Load function (LD) of first counter which serves to both confirm the entry prior to dosing (enter) and reset the system (reset).

The basic operation of the circuit for a given request can be grouped into 4 stages:
**Stage A.** The LD switch is activates (reset/enter) effectively initiating the motor to the 0 position by outputting a steady voltage through switch 1a. The user then uses a keypad to enter the requested dosage value.

**Stage B.** Stage B begins when the user once again toggles the LD switch, the equivalent of pressing enter. The circuit outputs the control signals to the motor and the motor moves the desired amount.

**Stage C.** The final control pulse is a constant signal to the final winding which holds the motor in position to ensure accurate dosage prior to removal of the syringe.

**Stage D.** After the syringe is removed the motor is returned to the 0 position and the system is reset initiating a constant voltage to control switch 1a and returning the cycle to Stage A. The following image and table depict the cycle for a request of 25 units.
<table>
<thead>
<tr>
<th>Time Pos, Duration</th>
<th>2b</th>
<th>2a</th>
<th>1b</th>
<th>1a</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stage A</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151068</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Stage B</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151113</td>
<td>20</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>151133</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>151153</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>151173</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>151193</td>
<td>20</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>151213</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>151233</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>151253</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>151273</td>
<td>20</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>151293</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>151313</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>151333</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>151353</td>
<td>20</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>151373</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>151393</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>151413</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>151433</td>
<td>20</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>151453</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>151473</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>151493</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>151513</td>
<td>20</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>151533</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>151553</td>
<td>20</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>151573</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Stage C</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151593</td>
<td>161</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>Stage D</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151754</td>
<td>176</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Part Schematics:
The following are the part schematics for all of the elements in the control circuit. Pricing estimate for all the parts is less than 1$ (far cheaper than a microprocessor).

**7425 Dual 4-Input NOR Gate (With Strobe)**
This device contains 2 independent 4-input NOR gates.

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>S A B C D Y</td>
<td></td>
</tr>
<tr>
<td>L L L L UNDEF</td>
<td></td>
</tr>
<tr>
<td>H X X L H</td>
<td></td>
</tr>
<tr>
<td>H X H X H</td>
<td></td>
</tr>
<tr>
<td>X X H X L</td>
<td></td>
</tr>
<tr>
<td>X X X H L</td>
<td></td>
</tr>
</tbody>
</table>

FUNCTION TABLE (each NOR gate)

Notes:
1. X is a don't care state (may be L or H).
2. The 5th input must be high to receive usable output.

**7404 Hex Inverter**
This device contains six independent inverters.

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Y</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
</tr>
</tbody>
</table>

FUNCTION TABLE (each inverter)

**7408 Quad 2-Input AND Gate**
This device contains 4 independent 2-input AND gates.

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B Y</td>
<td></td>
</tr>
<tr>
<td>L L L L</td>
<td></td>
</tr>
<tr>
<td>L H L L</td>
<td></td>
</tr>
<tr>
<td>H H L H</td>
<td></td>
</tr>
</tbody>
</table>

FUNCTION TABLE (each AND gate)
Motor and Gear Setup

Relevant equations:

Circumference = \pi D
R = (1/2)D

Circumference is calculated by multiplying the diameter of a circle by \pi. The radius of a circle is one half its diameter.

Syringe Calculations:

Type of syringe: Standard 1cc ¼” diameter insulin syringe
Overall Length of syringe tube: 7 cm
Total volume of syringe: 1cc  
Accuracy of syringe pump needed: 0.01cc  
Length of syringe that contains 0.01cc of volume:  
\[ \frac{1cc}{7cm} = \frac{0.01cc}{x} \]
\[ x = 0.07cm \]

The syringe being used in this project is a standard one cc ¼” diameter syringe. This is one of the most common syringes used for dosing insulin and heparin. The overall length of the tube of the syringe which contains the fluid is seven centimeters. The total volume of the syringe is one cubic centiliter. The accuracy of the syringe pumping mechanism must be to the nearest one hundredth of a cubic centiliter. The length that contains one hundredth of a cubic centiliter can be calculated by setting up a ratio between overall length and volume. The amount of distance the plunger must move in order to pull in one cubic centiliter of volume is calculated as seven hundredths of a centimeter.

**Motor Calculations:**

Type of Motor: Digikey Stepper Motor (403-1001-ND)  
Step Rotation: 18°  
Steps per full rotation: \[ \frac{360°}{18°} = 20 \]  
Percent Rotation: \[ \frac{18°}{360°} = 5\% \]

<table>
<thead>
<tr>
<th>Technical/Catalog Information</th>
<th>403-1001-ND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Package</td>
<td>30</td>
</tr>
<tr>
<td>Category</td>
<td>Motors &amp; Solenoids</td>
</tr>
<tr>
<td>Family</td>
<td>Stepper Motors</td>
</tr>
<tr>
<td>Vendor</td>
<td>Portescap Danaher Motion US, LLC</td>
</tr>
<tr>
<td>Voltage-Rated</td>
<td>5VDC</td>
</tr>
<tr>
<td>Size</td>
<td>20mm x 34.9mm x 1.5mm</td>
</tr>
<tr>
<td>Type</td>
<td>Unipolar</td>
</tr>
<tr>
<td>Holding Torque</td>
<td>7.8/1.1</td>
</tr>
<tr>
<td>Step Angle</td>
<td>18°</td>
</tr>
<tr>
<td>Steps per Revolution</td>
<td>20</td>
</tr>
<tr>
<td>Other Names</td>
<td>403-1001</td>
</tr>
</tbody>
</table>

Source: [www.digikey.com](http://www.digikey.com)

The syringe dosing unit will use a DC stepper motor. One step of this motor is approximately eighteen degrees of rotation. Dividing the full number of degrees in a rotation by eighteen degrees it can be seen that twenty steps are needed to make on full rotation. Dividing eighteen degrees into three hundred and sixty degrees gives the percent rotation per step. This value is five percent.
**Gear Calculations:**

Gear 1: Spur gear connected directly to motor
Circumference: 0.5cm
Diameter: \( \frac{0.5\text{cm}}{\pi} \)
Rotation in cm: \( 0.5\text{cm} \times 5\% = 0.025\text{cm} \)
Desired Rotation per step: 0.07cm

Gear 2: Another free spinning spur gear connected to spur gear of motor
Ratio Calculation: \( 0.025\text{cm} \times y = 0.07\text{cm} \) \( y = 2.8 \)
Size of gear 2: Size of gear 1 \( \times 2.8 = 1.4\text{cm} \) circumference

Gear 3: Geared shaft on a free sliding mechanism attached to roof of syringe dosing mechanism case.

---

Gear one is connected to the shaft of the motor and is a round spur gear. The circumference of this gear is one half of a centimeter. The diameter of the gear can be found by dividing the circumference by \( \pi \). The gears are being analyzed using their circumferences rather than their diameters to simplify calculations. The rotation of gear one when rotated five percent of a rotation is twenty five thousandths of a centimeter. Gear two is another spur gear directly connected in between the spur gear of the motor and a geared shaft. In order for the geared shaft to move seven hundredths of a
centimeter this gear must have a one and four tenths centimeter circumference. This will move the free sliding geared shaft seven hundredths of a centimeter for each step of the motor. The geared shaft will be directly connected to the plunger in order to pump insulin or heparin into the syringe.

B. DESIGN II

The Accessible Syringe Dosing Device will meet all specifications as described above. It will use a ball-screw or cam screw mechanism to linearly shift a movable stage and thus draw back the syringe. A stepper motor (combined with a geared reducer if necessary) would be used to rotate the screw and thus drive the ball alpine connected to the stage. A simple logic circuit could be used to process the user’s request and translate it into a control function for the stepper motor. It is important to note that the number of logical pulses produced is be proportional to the number of rotations of the screw mechanism and therefore the distance the stage and syringe move (see calculations in the Analysis section). Ideally a single unit of motion for the stepper motor should correspond to the dosing of one corresponding to 0.01cc of medication. Shown below are several views of our design. The primary modifications to the interface and aesthetics of the device from the first device are as follows:

- The (+ -) interface is replaced by a numerical keypad that will enable the user to enter in an exact amount with greater ease.
- The expanded button selection also enables the user to program and access many more presets than before. This could be extremely useful for the mixing of multiple drugs of multiple dosages.
- The LCD display will be replaced by and LCD character module which will assure a more user-friendly environment. This coupled with the inclusion of a microprocessor will accommodate the increasing functional load, minimize the circuit pieces necessary and add a personalized memory component. Since the display is no longer limited to numbers, the interface can be much more elaborate and include a help feature.
- The previous “in box” design where the syringe is completely encased will be replaced by a sliding half-box-cutaway (patent pending) mechanism. This will allow the user easier access to the syringe holding compartment.

All figures have been produced using Microsoft Visio.
Internal Views:

External Views:
3cm by 1cm

LCD Character Module
Motor Selection

- Note that this specific 6-pole rotor turns in a direction opposite the rotation of the stator field. A two-pole rotor inside the exact same field would rotate in the same direction.

- This illustration is based on half-step control, where alternate half steps involve one and two motor windings.

- It takes three complete cycles of the control system to turn this 6-pole rotor one revolution. A two-pole rotor would turn a full revolution per control system cycle.
Of the various types of stepping motors we have chosen a unipolar for our design. The basic design and specifications are depicted to the right. Note that the main advantage of using a unipolar motor over a variable reluctance motor lies in the number of control signals required. As depicted below, the basic unipolar control circuit outputs to four winding control switches (for two windings) whereas the standard variable reluctance motor outputs to three control switches (one for each of the three windings). Four control signals is much easier to control using a binary based controlled circuit than three control circuits simply because of the ease of converting from a base two to a base four number.

In terms of our actual circuitry this means that we can easily distribute a given number of pulses between our four switches using a simple 8-bit counter unit (see the actual circuit diagram and accompanying materials for details). Besides the ease of designing a control circuit there are several other reasons why the unipolar motor was chosen for our design:
4. The unipolar motors are far cheaper than variable reluctance and bipolar motors.
5. Permanent magnet motors (in general) have a higher torque output per power input ratio than the variable reluctance motors.
6. As discussed earlier, by varying the number of poles (2, 4, 6 etc.) one can control the minimum step size (90°, 45°, 30° etc). \( S_{\text{min}} = \frac{180}{P} \). Note that our current selection is a 10 polar, 18° motor.

**Control Circuit**

The basic schematic for our control circuit is seen in Design I. The circuit inputs an 8-bit binary number corresponding to the number of units (1 unit = .01 cc) desired. It then outputs logical pulses to the four control switches in the driver circuit as specified previously. Since the minimum step size of the motor (18°) has been designed (through the gear system) to coincide with a 1 unit intake (see gear transfer section) the total number of pulses distributed is equal to the inputted binary number. This circuit is useful in minimizing the cost of our overall design by eliminating the need for a microprocessor. Our final design, however, may very well include a microprocessor which will likely replace some (or all) of this hardware.

The circuit functions as follows:
7. The first 8-bit counter hooked up to a basic 555-chip clock or the microprocessor inputs the requested number of units.
8. The counter enable input is directly linked to the “nor” value of the counter outputs (1 if all 0s). Thus the counter counts down from the requested value to 0 and stops.
9. A 3-way “and” gate is used to limit the total number of clock pulses passed to the second counter to the number requested (the number of pulses used to get the counter to 0)
10. The second counter counts up to the desired dosage (a safety measure) and then distributes the pulses to the 4 control switches. It does so using the principles of binary notation (see truth table) and four 2-way “and” gate.
11. The truth table for step 4 which determines the configuration of the and gates:

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q0</th>
<th>1a</th>
<th>1b</th>
<th>2a</th>
<th>2b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
12. The overall circuit is controlled by the Load function (LD) of first counter which serves to both confirm the entry prior to dosing (enter) and reset the system (reset).

The basic operation of the circuit for a given request can be grouped into 4 stages:

**Stage A.** The LD switch is activated (reset/enter) effectively initiating the motor to the 0 position by outputting a steady voltage through switch 1a. The user then uses a keypad to enter the requested dosage value.

**Stage B.** Stage B begins when the user once again toggles the LD switch, the equivalent of pressing enter. The circuit outputs the control signals to the motor and the motor moves the desired amount.

**Stage C.** The final control pulse is a constant signal to the final winding which holds the motor in position to ensure accurate dosage prior to removal of the syringe.

**Stage D.** After the syringe is removed the motor is returned to the 0 position and the system is reset initiating a constant voltage to control switch 1a and returning the cycle to Stage A. The following image and table depict the cycle for a request of 25 units.

---

**Ball-Screw and Stage Assembly**

Sliding Stages:

---

#4 T-Series – Our choice

Low cost, highest performance/cost ratio, industry standard envelope:
Patented AccuGlide T-Series linear ball guides offer lower purchase price and installation cost with the high load capacity of traditional ProfileRail linear guides.

Ball-Screw:

Ball screw assembly consists of a screw with a precision ground or rolled helical groove, a nut (the outer race) with an internal groove, and a circuit of precision steel balls that recirculate in the grooves between the screw and nut. This anti-friction design converts torque to thrust as either the screw or nut turns and the other component moves in a linear direction.

Example of the Assembly in its entirety:

Source for figures on this page: www.netmotion.com

Part Schematics

The following are the part schematics for all of the elements in the control circuit.

555 Timer Chip

7425 Dual 4-Input NOR Gate (With Strobe)

This device contains 2 independent 4-input NOR gates.
**7404 Hex Inverter**

This device contains six independent inverters.

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
</tr>
</tbody>
</table>

**FUNCTION TABLE**

(each inverter)

**7408 Quad 2-Input AND Gate**

This device contains 4 independent 2-input AND gates.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>

**FUNCTION TABLE**

(each AND gate)

**7411 Triple 3-Input AND Gate**

This device contains 3 independent 3-input AND gates.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
</tbody>
</table>

**FUNCTION TABLE**

(each NAND gate)
Motor Ball-Shaft and Stage Mechanism Setup

Relevant equations

Circumference = πD
R = (1/2)D
Circumference is calculated by multiplying the diameter of a circle by pi. The radius of a circle is one half its diameter.

Syringe Calculations

Type of syringe: Standard 1cc ¼” diameter insulin syringe
Overall Length of syringe tube: 7 cm
Total volume of syringe: 1cc
Accuracy of syringe pump needed: 0.01cc
Length of syringe that contains 0.01cc of volume:
7cm = 1 * (0.01cc / 1 cc)
x = .07cm

The syringe being used in this project is a standard one cc ¼” diameter syringe. This is one of the most common syringes used for dosing insulin and heparin. The overall length of the tube of the syringe that contains the fluid is seven centimeters. The total volume of the syringe is one cubic centiliter. The accuracy of the syringe pumping mechanism must be to the nearest one hundredth of a cubic centiliter. The length that contains one one-hundredth of the total volume is therefore one one-hundredth of the length as calculated by the ratio between overall length and volume. The amount of
distance the plunger must move in order to pull in one cubic centiliter of volume is seven-hundredths of a centimeter.

**Ball-Screw Calculations**

\[ X = N_R \times D_T \]
Where:
\( N_R \) = the number of rotations
\( D_T \) = the distance between the threads on the screw
\( X \) = the distance the stage moves

**Motor Calculations**

Type of Motor: DigiKey Stepper Motor (403-1001-ND)
Step Rotation: 18°
Steps per full rotation: \( 360° / 18° = 20 \)
Percent Rotation: \( 18° / 360° = 5\% \)

<table>
<thead>
<tr>
<th>Technical/Catalog Information</th>
<th>403-1001-ND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Package</td>
<td>30</td>
</tr>
<tr>
<td>Category</td>
<td>Motors &amp; Solenoids</td>
</tr>
<tr>
<td>Family</td>
<td>Stepper Motors</td>
</tr>
<tr>
<td>Vendor</td>
<td>Portescap Danaher Motion US, LLC</td>
</tr>
<tr>
<td>Voltage-Rated</td>
<td>5VDC</td>
</tr>
<tr>
<td>Size</td>
<td>20mm x 34.9mm x 1.5mm</td>
</tr>
<tr>
<td>Type</td>
<td>Unipolar</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>Holding Torque</td>
<td>7.8/1.1</td>
</tr>
<tr>
<td>Step Angle</td>
<td>18°</td>
</tr>
<tr>
<td>Steps per Revolution</td>
<td>20</td>
</tr>
<tr>
<td>Other Names</td>
<td>403-1001</td>
</tr>
</tbody>
</table>

Source: [www.digikey.com](http://www.digikey.com)

The syringe dosing unit will use a DC stepper motor. One step of this motor is approximately eighteen degrees of rotation. Dividing the full number of degrees in a rotation by eighteen degrees it can bee seen that twenty steps are needed to make on full rotation. Dividing eighteen degrees into three hundred and sixty degrees gives the percent rotation per step. This value is five percent.

C. DESIGN III

The Accessible Syringe Dosing Device will meet all specifications as described above. It will use a ball-screw or cam screw mechanism to linearly shift a movable stage and thus draw back the syringe. A stepper motor (combined with a geared reducer if necessary) would be used to rotate the screw and thus drive the ball alpine connected to the stage. A microprocessor will be used to process the user’s request and translate it into a control function for the stepper motor. It is important to note that the number of logical pulses produced is be proportional to the number of rotations of the screw mechanism and therefore the distance the stage and syringe move (see calculations in the Analysis section). Ideally a single unit of motion for the stepper motor should correspond to the dosing of one corresponding to 0.01cc of medication.

Shown below are several views of our design. The primary modifications to the interface and aesthetics of the device from the first device are as follows:

- The (+/-) interface is replaced by a numerical keypad that will enable the user to enter in an exact amount with greater ease.
- The expanded button selection also enables the user to program and access many more presets than before.
- The LCD display will be replaced by and LCD character module which will assure a more user-friendly environment. This coupled with the inclusion of a microprocessor will accommodate the increasing functional load, minimize the circuit pieces necessary and add a personalized memory component. Since the display is no longer limited to numbers, the interface can be much more elaborate and include a help feature.
- The previous “in box” design where the syringe is completely encased will be replaced by a sliding half-box-cutaway mechanism. This will allow the user easier access to the syringe holding compartment.

All figures have been produced using Microsoft Visio.

**Internal Views:**
External Views: Control Circuit

Side View (C)

Control Circuit
Motor Selection

- Note that this specific 6-pole rotor turns in a direction opposite the rotation of the stator field, a two-pole rotor inside the exact same field would rotate in the same direction.

- This illustration is based on half-step control, where alternate half steps involve one and two motor windings.

- It takes three complete cycles of the control system to turn this 6-pole rotor one revolution. A two-pole rotor would turn a full revolution per control system cycle.
Of the various types of stepping motors we have chosen a unipolar for our design. The basic design and specifications are depicted to the right. Note that the main advantage of using a unipolar motor over a variable reluctance motor lies in the number of control signals required. As depicted below, the basic unipolar control circuit outputs to four winding control switches (for two windings) whereas the standard variable reluctance motor outputs to three control switches (one for each of the three windings). Four control signals is much easier to control using a binary based controlled circuit than three control circuits simply because of the ease of converting from a base two to a base four number. In terms of our actual circuitry this means that we can easily distribute a given number of pulses between our four switches using a simple 8-bit counter unit (see the actual circuit diagram and accompanying materials for details). Besides the ease of designing a control circuit there are several other reasons why the unipolar motor was chosen for our design:
7. The unipolar motors are far cheaper than variable reluctance and bipolar motors.
8. Permanent magnet motors (in general) have a higher torque output per power input ratio than the variable reluctance motors.
9. As discussed earlier, by varying the number of poles (2, 4, 6 etc.) one can control the minimum step size ($90^\circ$, $45^\circ$, $30^\circ$ etc). $S_{\text{min}}=180/P$. Note that our current selection is a 10 polar, $18^\circ$ motor.

**Microprocessor**

The microprocessor we have chosen for this portion of the design is the PIC16F874A chip. This chip is ideal for our purposes because the functions of the microprocessor fit our needs nicely. Additionally, the hardware and software packages needed to program the chip are readily available to our team of designers. Specifically, the microprocessor will control the motor and thus, control the ball-shaft mechanism in the design. The program for the chip will be written in assembly code, using Microchip MPLAB ICD2 software. A sample of the type of program we may employ is given below the chip information seen here.

<table>
<thead>
<tr>
<th>Technical/Catalog Information</th>
<th>PIC16F874A-I/PT-ND</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standard Package</strong></td>
<td>160</td>
</tr>
<tr>
<td><strong>Category</strong></td>
<td>Integrated Circuits (IC's)</td>
</tr>
<tr>
<td><strong>Family</strong></td>
<td>Microcontrollers</td>
</tr>
<tr>
<td><strong>Vendor</strong></td>
<td>Microchip Technology</td>
</tr>
<tr>
<td><strong>Program Memory Size</strong></td>
<td>4K x 14</td>
</tr>
<tr>
<td><strong>RAM Size</strong></td>
<td>192 x 8</td>
</tr>
<tr>
<td><strong>Number of I/O</strong></td>
<td>33 (8 Ch. 10-Bit A/D)</td>
</tr>
<tr>
<td><strong>Package / Case</strong></td>
<td>44-TQFP</td>
</tr>
<tr>
<td><strong>Speed</strong></td>
<td>20MHz</td>
</tr>
<tr>
<td><strong>Controller Series</strong></td>
<td>PIC16F87x 8-Bit CMOS FLASH with 10-Bit A/D</td>
</tr>
<tr>
<td><strong>Oscillator Type</strong></td>
<td>Xtal/RC External</td>
</tr>
<tr>
<td><strong>Packaging</strong></td>
<td>Bulk</td>
</tr>
<tr>
<td><strong>Program Memory Type</strong></td>
<td>FLASH</td>
</tr>
<tr>
<td><strong>EEPROM Size</strong></td>
<td>128 x 8</td>
</tr>
</tbody>
</table>
Example Code: PIC Code Relevant to Stepper-Motor Control

The following program, taken from http://www.emu8086.com/vb/index_asm.html, will serve as a template for implementing a logical stepper-motor controller. Although the basic code was originally intended for a virtual microprocessor, it is most certainly applicable to our situation as well. The program however, is far from complete, as that it only writes the outputs and does not contain any user interface.

```
#make_bin#
#CS = 500#
#IP = 0#

; This is a sample of OUT instruction.
; It writes values to virtual I/O port
; that controls the stepper-motor.
; Try using datCCW, datCW_FS or datCCW_FS
; instead of datCW to get different
; behavior of the motor.
; set Data Segment to code:
MOV AX, CS
MOV DS, AX
MOV SI, 0
next_situation:
MOV AL, datCW[SI]
OUT 7, AL
INC SI
CMP SI, 4
JB next_situation
MOV SI, 0
JMP next_situation

; bin data for clock-wise
; half-step rotation:
datCW db 110b
db 100b
db 011b
db 010b

; bin data for counter-clock-wise
; half-step rotation:
datCCW db 011b
db 001b
db 110b
db 010b

; bin data for clock-wise
; full-step rotation:
datCW_FS db 001b
db 011b
db 110b
db 000b

; bin data for counter-clock-wise
; full-step rotation:
datCCW_FS db 100b
db 110b
```
Number Conversion

The second segment of code (taken from the same website) provides a convenient list of functions for converting between different number bases. This is extremely useful in facilitating communication between the user and the device since the device functions in binary and the user requires decimal.

#make_BIN#
; load binary value:
; (decimal: 5)
MOV AL, 00000101b
; load HEX value:
; (decimal: 10)
MOV BL, 0Ah
; load Octal value:
; (decimal: 8)
MOV CL, 10o
; 5 + 10 = 15 (0Fh)
ADD AL, BL
; 15 - 8 = 7
SUB AL, CL
HLT

Stepper Motor Interface:

The final segment of code will form the backbone of our program. The main reasons why this specific code is so useful is its preliminary setup of the motor positional logic codes for a Uni-Polar Stepper-Motor (in bin form) and its switch input option. http://www.testech-elect.com/ccs/examples.htm
/// permission. Derivative programs created using this software ///
/// in object code form are not restricted in any way. ///
///////////////////////////////////////////////////////////////////////
//
#include <16c74.h>
#define fuses HS,NOWDT,NOPROTECT
#define delay(clock=20000000)
#define rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7) // Jumpers: 8 to 11, 7
to 12
#include <input.c>
#define port_b = 6
#define FOUR_PHASE TRUE

byte const POSITIONS[4] = {0b0101,
0b1001,
0b1010,
0b0110};
else
byte const POSITIONS[8] = {0b0101,
0b0001,
0b1001,
0b1000,
0b1010,
0b0010,
0b0110,
0b0100};

drive_stepper(byte speed, char dir, byte steps) {
static byte stepper_state = 0;
byte i;
for(i=0;i<steps;++i) {
delay_ms(speed);
set_tris_b(0xf0);
port_b = POSITIONS[ stepper_state ];
if(dir!='R')
stepper_state=(stepper_state+1)&(sizeof(POSITIONS)-1);
else
stepper_state=(stepper_state-1)&(sizeof(POSITIONS)-1);
}
}
use_pot() {
byte value;
setup_adc(adc_clock_internal);
set_adc_channel( 1 );
printf("\n");
while( TRUE ) {
value=read_adc();
printf("%2X\r",value);
if(value<0x80)
drive_stepper(value,'R',8);
else if(value>0x80)
drive_stepper(128-(value-128),'F',8);
}
}
use_switch(byte speed, char dir) {

}
byte steps;

printf("\n\rSteps per press: ");
steps = gethex();
while(true) {
while(input(PIN_B7)) ;
drive_stepper(speed,dir,steps);
while(!input(PIN_B7)) ;
delay_ms(100);
}

main() {
byte speed,steps;
char dir;
setup_port_a(RA0_RA1_ANALOG);
while (TRUE) {
printf("\n\rSpeed (hex): ");
speed = gethex();
if(speed==0)
use_pot();
printf("\n\rDirection (F,R): ");
dir=getc()|0x20;
putc(dir);
printf("\n\rSteps (hex): ");
steps = gethex();
if(steps==0)
use_switch(speed,dir);
execute_stepper(speed,dir,steps);
}

**Once again this code is not representation of our entire program. The user interface itself, including the display output and input function, still needs to be written.**
The following are figures that represent the microprocessor we have chosen for our design.
FIGURE 1-2: PIC16F874A/877A BLOCK DIAGRAM

<table>
<thead>
<tr>
<th>Device</th>
<th>Program FLASH</th>
<th>Data Memory</th>
<th>Data EEPROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F874A</td>
<td>4K words</td>
<td>192 Bytes</td>
<td>128 Bytes</td>
</tr>
<tr>
<td>PIC16F877A</td>
<td>8K words</td>
<td>512 Bytes</td>
<td>320 Bytes</td>
</tr>
</tbody>
</table>
Ball-Screw and Stage Assembly

Sliding Stages:

#4 T-Series – Our choice

Low cost, highest performance/cost ratio, industry standard envelope:

Patented AccuGlide T-Series linear ball guides offer lower purchase price and installation cost with the high load capacity of traditional ProfileRail linear guides.

Ball-Screw:

Ball screw assembly consists of a screw with a precision ground or rolled helical groove, a nut (the outer race) with an internal groove, and a circuit of precision steel balls that recirculate in the grooves between the screw and nut. This anti-friction design converts torque to thrust as either the screw or nut turns and the other component moves in a linear direction.

Example of the Assembly in its entirety:

Source for figures on this page: www.netmotion.com
Part Schematics

The following are the part schematics for all of the elements in the control circuit.

555 Timer Chip

7425 Dual 4-Input NOR Gate (With Strobe)

This device contains 2 independent 4-input NOR gates.

7404 Hex Inverter

This device contains six independent inverters.

7408 Quad 2-Input AND Gate

This device contains 4 independent 2-input AND gates.
Motor Ball-Shaft and Stage Mechanism Setup

**Relevant equations**

Circumference = \( \pi D \)

\[ R = \left(\frac{1}{2}\right)D \]

Circumference is calculated by multiplying the diameter of a circle by \( \pi \). The radius of a circle is one half its diameter.
**Syringe Calculations**

Type of syringe: Standard 1cc ¼” diameter insulin syringe  
Overall Length of syringe tube: 7 cm  
Total volume of syringe: 1cc  
Accuracy of syringe pump needed: 0.01cc  
Length of syringe that contains 0.01cc of volume:  
\[ 7\text{cm} = 1 \times (0.01\text{cc} / 1 \text{ cc}) \]  
\[ x = .07\text{cm} \]

The syringe being used in this project is a standard one cc ¼” diameter syringe. This is one of the most common syringes used for dosing insulin and heparin. The overall length of the tube of the syringe that contains the fluid is seven centimeters. The total volume of the syringe is one cubic centiliter. The accuracy of the syringe pumping mechanism must be to the nearest one hundredth of a cubic centiliter. The length that contains one one-hundredth of the total volume is therefore one one-hundredth of the length as calculated by the ratio between overall length and volume. The amount of distance the plunger must move in order to pull in one cubic centiliter of volume is seven-hundredths of a centimeter.

**Ball-Screw Calculations**

\[ X = N_R \times D_T \]  
Where:  
\( N_R = \) the number of rotations  
\( D_T = \) the distance between the threads on the screw  
\( X = \) the distance the stage moves
## Motor Calculations

Type of Motor: DigiKey Stepper Motor (403-1001-ND)
Step Rotation: 18°
Steps per full rotation: 360° / 18° = 20
Percent Rotation: 18° / 360° = 5%

<table>
<thead>
<tr>
<th>Technical/Catalog Information</th>
<th>403-1001-ND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Package</td>
<td>30</td>
</tr>
<tr>
<td>Category</td>
<td>Motors &amp; Solenoids</td>
</tr>
<tr>
<td>Family</td>
<td>Stepper Motors</td>
</tr>
<tr>
<td>Vendor</td>
<td>Portescap Danaher Motion US, LLC</td>
</tr>
<tr>
<td>Voltage-Rated</td>
<td>5VDC</td>
</tr>
<tr>
<td>Size</td>
<td>20mm x 34.9mm x 1.5mm</td>
</tr>
<tr>
<td>Type</td>
<td>Unipolar</td>
</tr>
<tr>
<td>Holding Torque</td>
<td>7.8/1.1</td>
</tr>
<tr>
<td>Step Angle</td>
<td>18°</td>
</tr>
<tr>
<td>Steps per Revolution</td>
<td>20</td>
</tr>
<tr>
<td>Other Names</td>
<td>403-1001</td>
</tr>
</tbody>
</table>

Source: [www.digikey.com](http://www.digikey.com)

The syringe dosing unit will use a DC stepper motor. One step of this motor is approximately eighteen degrees of rotation. Dividing the full number of degrees in a rotation by eighteen degrees it can bee seen that twenty steps are needed to make on full rotation. Dividing eighteen degrees into three hundred and sixty degrees gives the percent rotation per step. This value is five percent.
IV. OPTIMAL DESIGN

The optimal accessible syringe dosing device will meet all specifications as described above. It will use a linear actuator instead of a ball-screw mechanism to linearly shift a movable stage and thus draw back the syringe. A stepper motor will be used to rotate the screw and thus drive the linear actuator. A microprocessor will be used to process the user’s request and translate it into a control function for the stepper motor and actuator. It is important to note that the number of logical pulses produced must be proportional to the number of rotations of the linear actuator and therefore the distance the actuator and syringe must move (see calculations in the Analysis section). Ideally, a single unit of motion for the linear actuator should correspond to the dosing of one corresponding to 0.01cc of medication. Several views of the design are shown below. The primary attributes of the interface and aesthetics of the device are as follows:

- The interface will have a numerical keypad that will enable the user to enter an exact amount with ease.
- The button selection enables the user to store and access many presets.
- The LCD display will ensure a user-friendly interface. This, coupled with the inclusion of a microprocessor, will minimize the circuit pieces necessary and add a personalized memory component. Since the display is not limited to numbers, the interface can include a help feature.
Internal Views:

All figures have been produced using Microsoft Visio.
External Views:

User Interface
The block diagram below is a schematic of all the components of the device.

**Syringe Dosing Project Block Diagram**

- **User Interface**: Choose Proper Dose, Presets, Help Menu
- **Microprocessor**: Display on C-LCD and Operate Motor
- **Battery**: Power all components
- **Driver**
- **Encoder**
- **Linear Actuator**
- **C-LCD Display**
- **Syringe**
EQUIPMENT AND ANALYSIS

Microprocessor

The microprocessor we have chosen for this portion of the design is the PIC16F874A chip. The functions of this chip are ideal for our purposes. Additionally, the hardware and software packages needed to program the chip are readily available to our team of designers. Specifically, the microprocessor will control the linear actuator in the design. The program for the chip will be written in assembly code, using Microchip MPLAB ICD2 software. The microprocessor will be connected to a driver and an encoder for the linear actuator. The driver will amplify the voltage signal given by the microprocessor and it also creates a logical signal to control the motor. The encoder is a tracking device for the motor that can track motor speed, number of steps, and acts as a feedback mechanism to make sure the actuator is working properly.

<table>
<thead>
<tr>
<th>Technical/Catalog Information</th>
<th>PIC16F874A-I/PT-ND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Package</td>
<td>160</td>
</tr>
<tr>
<td>Category</td>
<td>Integrated Circuits (IC’s)</td>
</tr>
<tr>
<td>Family</td>
<td>Microcontrollers</td>
</tr>
<tr>
<td>Vendor</td>
<td>Microchip Technology</td>
</tr>
<tr>
<td>Program Memory Size</td>
<td>4K x 14</td>
</tr>
<tr>
<td>RAM Size</td>
<td>192 x 8</td>
</tr>
<tr>
<td>Number of I/O</td>
<td>33 (8 Ch. 10-Bit A/D)</td>
</tr>
<tr>
<td>Package / Case</td>
<td>40-DIP</td>
</tr>
<tr>
<td>Speed</td>
<td>20MHz</td>
</tr>
<tr>
<td>Controller Series</td>
<td>PIC16F87x 8-Bit CMOS FLASH with 10-Bit A/D</td>
</tr>
<tr>
<td>Oscillator Type</td>
<td>Xtal/RC External</td>
</tr>
<tr>
<td>Packaging</td>
<td>Bulk</td>
</tr>
<tr>
<td>Program Memory Type</td>
<td>FLASH</td>
</tr>
<tr>
<td>EEPROM Size</td>
<td>128 x 8</td>
</tr>
</tbody>
</table>
**Figure 1-2: PIC16F874A/877A Block Diagram**

- **Flash Program Memory**
- **Program Counter**
- **5 Level Stack (1296)**
- **RAM Pll Registers**
- **RAM Address (16)**
- **Address Multiplexer (9)**
- **Direct Address (7)**
- **Indirect Address (8)**
- **FBR Register 8 (FSR)**
- **Address Multiplexer (9)**
- **Status Register (7)**
- **Instruction Register**
- **Instruction Decode & Control**
- **Timing Generation**
- **OSC/CLOCK OSCICLKOUT**
- **Power-up Timer**
- **Watchdog Timer**
- **Brown-out Reset**
- **128k Circuit (Reset)**
- **Low-Voltage Programming**
- **MCLR (VDD, VSS)**
- **Timer0**
- **Timer1**
- **Timer2**
- **10-bit A/D**
- **Parallel Slave Port**
- **Data EEPROM**
- **CCP1, CCP2**
- **Synchronous Serial Port**
- **USART**
- **Comparator**
- **Voltage Reference**

<table>
<thead>
<tr>
<th>Device</th>
<th>Program Flash</th>
<th>Data Memory</th>
<th>Data EEPROM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIC16F874A</td>
<td>4K words</td>
<td>162 bytes</td>
<td>128 bytes</td>
</tr>
<tr>
<td>PIC16F877A</td>
<td>9K words</td>
<td>366 bytes</td>
<td>256 bytes</td>
</tr>
</tbody>
</table>
The user interface will display the following messages. The user will either enter the dosage or recall a preset dosage and then follow the prompts, which will be programmed by the microprocessor.

Designation

Welcome1

Welcome2

MainMenu0

MainMenu1

MainMenu2

MainMenu3

DosingMenu1

Dosing Screen

Syringe Check

Confirmation

Screen

Dosing Now

Modify1

Modify2

Safe

PresetsMenu0

PresetsMenu1

PresetsMenu2

PresetsMenu3
Scrol Addition

Additional Notes

Display on startup
Pause

FOR ASSISTANCE

Scrol Left
Pause

Wait for selection, v = MainMenu[+1], ^ = MainMenu[-1], ENTER = Enter Menu[X], Note: MainMenu[3] +1 = MainMenu[0] or vice versa

After enter selection

Dosing
Pause wait for ENTER
S1S0 = dosage amount/patient lable, M1M0 = "CC" for 2nd = 0 and " " for 2nd = 1, N1N0 = "0." for 2nd = 0 and "P#" for 2nd = 1, ENTER selects, CLEAR backspaces
Wait for selection (see above)

IN PLACE

Scrol Left
Wait for ENTER to confirm

Maybe Scrol the … for effect
note that s1s2 are if 2nd = 0 if 2nd = 1 skip to Safe screen
Switch back and forth
Switch back and forth
Wait for enter to return to MainMenu0 if 2nd = 0 or to StoreMenu1 if 2nd = 1 and patient has "store option" set (to 1)

MOVE THE SYRINGE

Driver

The driver that will be used is a DCM8028 micro stepping chopper driver. Using a driver will simplify the circuit because the driver will incorporate an amplifier that we will not have to design ourselves. This driver is also specified for the linear actuator model that we will be using in our design.
Linear Actuator

The linear actuator that will be incorporated into the design is a 35000 series captive linear actuator size 14, by Haydon Switch and Instrument, Inc. This is a hybrid linear actuator with a resolution of 0.003175 mm to .048768 mm (0.003175-0.00192 inches) per step. It can deliver a thrust up to fifty pounds which is extremely more force than needed for a syringe. The linear actuator will rotate internally and pull the syringe backwards linearly. A linear actuator will provide greater precision, higher resolution, and less machining than did the linear mechanisms presented in the other designs.
Syringe Calculations

Type of syringe: Standard 1cc ¼” diameter insulin syringe  
Overall Length of syringe tube: 5.75 cm  
Total volume of syringe: 1cc  
Accuracy of syringe pump needed: 0.01cc  
Length of syringe that contains 0.01cc of volume:  
5.75cm = 1 * (0.01cc / 1 cc)  
x = .575 mm

The syringe being used in this project is a standard one cc ¼” diameter syringe. This is one of the most common syringes used for dosing insulin and heparin. The overall length of the tube of the syringe that contains the fluid is seven centimeters. The total volume of the syringe is one cubic centiliter. The accuracy of the syringe pumping mechanism must be to the nearest one hundredth of a cubic centiliter. The length that contains one one-hundredth of the total volume is therefore one one-hundredth of the length as calculated by the ratio between overall length and volume. The amount of distance the plunger must move in order to pull in one hundredth cubic centiliter of volume is 0.575 millimeters.
**LabVIEW™ Simulation:**

This program allows the user to input a value of insulin up to one hundred units. This is done by using the plus-minus key. The on/off button is used as an enter key to pull the insulin into the syringe after the correct dose is entered. The blue represents the amount of insulin in the bottle and the red represents the amount of insulin in the syringe. The LCD display shows the amount of units pulled into the syringe.

**Front Panel:**

![Diagram of Syringe Dosing Simulation](image-url)
Block Diagram:

- **On/Off Button**
- **Plus Minus**
- **Syringe 1cc total**
- **Bottle of Insulin (1cc total)**
- **True**
- **Value: 100**
V. REALISTIC CONSTRAINTS AND SAFETY ISSUES

There are some constraints and limitations that will affect the accuracy of this syringe dosing project. First, the ball-screw mechanism must be somewhat sheltered and lubricated in order to assure the correct dosing of insulin. If a small particle somehow loges itself in the threads it could throw the rotation off and thus lead to an incorrect dosing. Since the second design incorporates a half-box-cutaway box system, the motor and gears must be enclosed completely. Testing is very important; the threads must stay perfectly in line with multiple usage and not wear this over time. The high precision of this device is essential for its success. This device must always be in operable condition. As a good measure of safety, a warning light and sound circuit must be incorporated to the device in order to warn the user that the motor needs a new battery in order to operate at full function.

There are no other major safety issues at this time; however, it is possible that more will arise as we move into the building phase. Currently, we predict that the potential for a high-risk electrical shock is near impossible since the maximum voltages in the motor circuit will not exceed twelve volts. A plastic case will be used with rounded edges to prevent injury when dropped. The syringe dosing unit is designed to be used with disposable needles, preventing the need for sterilization before each use. The main circuit will also be in a separate compartment to prevent tampering. Care should be taken not to have food or drink around the open syringe dosing apparatus. Liquids and food spilled anywhere near the device and its components may destroy its function.

VI. TECHNICAL SPECIFICATIONS

Technical Specifications

Electrical Parameters
Battery life: 6 months

Display:
Number of characters 15 minimum
Significant Figures 4
Height 1.3in
Width 6.5in
Illumination backlight to accommodate elderly and vision impaired

Motor:
Voltage 3-9 volts

Power Usage
User Interface Keypad and LCD Display

Volatges
Accuracy 0.01cc
Precision

**Mechanical Parameters**
- **Buttons**: 16 (0-9, Up, Down, Clear, Enter, Help, 2nd)
- **Button Size**
- **Weight**: 2kg maximum
- **Vibration**: Operates normally with trembling hand
- **Durability**: No adverse effects after 1m drop

**Environmental Parameters**
- **Operating Temperature**: 20-90 degrees C
- **Storage Temperature**: 20-90 degrees C

**Hardware and Software Parameters**
- Microprocessor:
  - **Programming**: programmed using MPLAB ICD2
  - **Memory**: 5 M minimum

Machine Dependent:
- **Wait times**: 2 min maximum
- **Increase/Decrease Dose**: 2 buttons
- **Preset**: 1 button
- **On/Off**: 1 button
VII. TIMELINE

The following timeline is representative of the tasks that will be completed during the spring of 2005.

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>Predecessors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Evaluation of Final Report</td>
<td>5 days</td>
<td>Mon 1/3/05</td>
<td>Fri 1/7/05</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Optimal design review</td>
<td>2 days</td>
<td>Mon 1/3/05</td>
<td>Tue 1/4/05</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Parts orders submitted</td>
<td>2 days</td>
<td>Tue 1/4/05</td>
<td>Wed 1/5/05</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Karen and Megan Machine shop certification</td>
<td>2.5 days</td>
<td>Mon 1/10/05</td>
<td>Wed 1/12/05</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Parts orders revised and completed</td>
<td>2 days</td>
<td>Wed 1/15/05</td>
<td>Thu 1/16/05</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Linear actuator price quote requested from vendor</td>
<td>1 day</td>
<td>Fri 1/14/05</td>
<td>Fri 1/14/05</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>Begin spreadsheet of expenses</td>
<td>1 day</td>
<td>Mon 1/17/05</td>
<td>Mon 1/17/05</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>Talk to Serge about case material</td>
<td>1 day?</td>
<td>Tue 1/18/05</td>
<td>Tue 1/18/05</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>Update website</td>
<td>1 day</td>
<td>Tue 1/18/05</td>
<td>Tue 1/18/05</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>Organize weekly report structure</td>
<td>1 day</td>
<td>Tue 1/18/05</td>
<td>Tue 1/18/05</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>Receive keypad and LCD display</td>
<td>1 day</td>
<td>Wed 1/19/05</td>
<td>Wed 1/19/05</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Finalize microprocessor order</td>
<td>1 day</td>
<td>Wed 1/19/05</td>
<td>Wed 1/19/05</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Receive Count-A-Dose product</td>
<td>1 day</td>
<td>Wed 1/19/05</td>
<td>Wed 1/19/05</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Review design constraints</td>
<td>1 day</td>
<td>Wed 1/19/05</td>
<td>Wed 1/19/05</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Rough drawings of case</td>
<td>2 days</td>
<td>Wed 1/19/05</td>
<td>Thu 1/20/05</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Finalize linear actuator order with express shipping</td>
<td>1 day</td>
<td>Fri 1/21/05</td>
<td>Fri 1/21/05</td>
<td>12</td>
</tr>
<tr>
<td>17</td>
<td>Construct model using cardboard</td>
<td>1 day</td>
<td>Fri 1/21/05</td>
<td>Fri 1/21/05</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Finalize dimensions of case</td>
<td>1 day</td>
<td>Fri 1/21/05</td>
<td>Fri 1/21/05</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Solder wires into display</td>
<td>1 day</td>
<td>Fri 1/21/05</td>
<td>Fri 1/21/05</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Find vendor for case material</td>
<td>2 days</td>
<td>Fri 1/21/05</td>
<td>Mon 1/24/05</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Make notes for user's manual</td>
<td>2 days</td>
<td>Mon 1/24/05</td>
<td>Tue 1/25/05</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Talk to Serge and Rich about case construction</td>
<td>1 day</td>
<td>Mon 1/24/05</td>
<td>Mon 1/24/05</td>
<td>4</td>
</tr>
<tr>
<td>23</td>
<td>Purchase order for case material (polycarbonate)</td>
<td>1 day</td>
<td>Tue 1/25/05</td>
<td>Tue 1/25/05</td>
<td>23</td>
</tr>
<tr>
<td>24</td>
<td>Order materials for case construction</td>
<td>2 days</td>
<td>Tue 1/25/05</td>
<td>Wed 1/26/05</td>
<td>22</td>
</tr>
<tr>
<td>25</td>
<td>Receive materials for case</td>
<td>1 day</td>
<td>Thu 1/27/05</td>
<td>Thu 1/27/05</td>
<td>24</td>
</tr>
<tr>
<td>26</td>
<td>Begin reverse engineering - take careful notes</td>
<td>3 days</td>
<td>Wed 1/26/05</td>
<td>Fri 1/28/05</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Communicate with HSI vendor regarding our application</td>
<td>2 days</td>
<td>Fri 1/21/05</td>
<td>Mon 1/24/05</td>
<td>13</td>
</tr>
<tr>
<td>28</td>
<td>Receive stepper motor</td>
<td>1 day</td>
<td>Wed 1/26/05</td>
<td>Wed 1/26/05</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Receive microprocessor</td>
<td>1 day</td>
<td>Wed 1/26/05</td>
<td>Wed 1/26/05</td>
<td>23</td>
</tr>
<tr>
<td>30</td>
<td>Start circuit board design</td>
<td>5 days</td>
<td>Thu 1/27/05</td>
<td>Wed 2/2/05</td>
<td>29</td>
</tr>
<tr>
<td>31</td>
<td>Add to notes for user's manual</td>
<td>2 days</td>
<td>Tue 2/1/05</td>
<td>Wed 2/2/05</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Begin microprocessor programming</td>
<td>5 days</td>
<td>Mon 1/31/05</td>
<td>Fri 2/4/05</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Task Name</td>
<td>Duration</td>
<td>Start</td>
<td>Finish</td>
<td>Predecessors</td>
</tr>
<tr>
<td>----</td>
<td>-----------------------------------------------</td>
<td>----------</td>
<td>-------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>33</td>
<td>Work on motor-linear actuator interface</td>
<td>6 days</td>
<td>Mon 1/31/05</td>
<td>Fri 2/3/05</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Finalize circuit needed for PC Board</td>
<td>7 days</td>
<td>Thu 2/3/05</td>
<td>Fri 2/11/05</td>
<td>30</td>
</tr>
<tr>
<td>35</td>
<td>Continue notes for user's manual</td>
<td>2 days</td>
<td>Mon 2/7/05</td>
<td>Tue 2/8/05</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Order PC Boards</td>
<td>2 days</td>
<td>Mon 2/14/05</td>
<td>Tue 2/15/05</td>
<td>34</td>
</tr>
<tr>
<td>37</td>
<td>Encode linear actuator to function with program microprocessor</td>
<td>5 days</td>
<td>Mon 2/7/05</td>
<td>Fri 2/11/05</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>Include keypad in programming</td>
<td>3 days</td>
<td>Wed 2/9/05</td>
<td>Fri 2/11/05</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>Draw plans for construction of case</td>
<td>5 days</td>
<td>Mon 2/7/05</td>
<td>Fri 2/11/05</td>
<td>24</td>
</tr>
<tr>
<td>40</td>
<td>Continue programming microprocessor</td>
<td>5 days</td>
<td>Mon 2/7/05</td>
<td>Fri 2/11/05</td>
<td>32</td>
</tr>
<tr>
<td>41</td>
<td>Add to notes for user's manual</td>
<td>2 days</td>
<td>Mon 2/14/05</td>
<td>Tue 2/15/05</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>Begin construction of case in machine shop</td>
<td>5 days</td>
<td>Mon 2/4/05</td>
<td>Fri 2/18/05</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>Continue motor-linear actuator interface work</td>
<td>5 days</td>
<td>Mon 2/14/05</td>
<td>Fri 2/18/05</td>
<td>37</td>
</tr>
<tr>
<td>44</td>
<td>Continue notes for user's manual</td>
<td>2 days</td>
<td>Mon 2/21/05</td>
<td>Tue 2/22/05</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>Receive PC Boards</td>
<td>1 day</td>
<td>Mon 2/21/05</td>
<td>Mon 2/21/05</td>
<td>36</td>
</tr>
<tr>
<td>46</td>
<td>Practice soldering as required</td>
<td>2 days</td>
<td>Mon 2/21/05</td>
<td>Tue 2/22/05</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>Begin construction and analysis of circuit</td>
<td>5 days</td>
<td>Mon 2/21/05</td>
<td>Fri 2/25/05</td>
<td>40</td>
</tr>
<tr>
<td>48</td>
<td>Continue case construction in machine shop</td>
<td>5 days</td>
<td>Mon 2/21/05</td>
<td>Fri 2/25/05</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>Add to notes for user's manual</td>
<td>2 days</td>
<td>Mon 2/28/05</td>
<td>Tue 3/1/05</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>Integrate all tested parts into working circuit</td>
<td>5 days</td>
<td>Mon 2/28/05</td>
<td>Fri 3/4/05</td>
<td>47</td>
</tr>
<tr>
<td>51</td>
<td>Order any additional parts that may be needed</td>
<td>4 days</td>
<td>Sat 3/5/05</td>
<td>Wed 3/9/05</td>
<td>50</td>
</tr>
<tr>
<td>52</td>
<td>Continue case construction</td>
<td>5 days</td>
<td>Mon 2/28/05</td>
<td>Fri 3/4/05</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>Review notes for user's manual</td>
<td>5 days</td>
<td>Sat 3/5/05</td>
<td>Thu 3/10/05</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>Spring Break</td>
<td>10 days</td>
<td>Sat 3/5/05</td>
<td>Thu 3/17/05</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>Organize documentation for work completed</td>
<td>5 days</td>
<td>Sat 3/5/05</td>
<td>Thu 3/10/05</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>Re-evaluate timeline and make necessary adjustments</td>
<td>3 days</td>
<td>Sat 3/5/05</td>
<td>Tue 3/8/05</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>Extensively troubleshoot working prototype</td>
<td>5 days</td>
<td>Mon 3/14/05</td>
<td>Fri 3/18/05</td>
<td>56</td>
</tr>
<tr>
<td>58</td>
<td>Continue notes for user's manual</td>
<td>2 days</td>
<td>Mon 3/14/05</td>
<td>Tue 3/15/05</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>Finish case construction</td>
<td>3 days</td>
<td>Mon 3/14/05</td>
<td>Wed 3/16/05</td>
<td>52</td>
</tr>
<tr>
<td>60</td>
<td>Debug, debug, debug</td>
<td>6 days</td>
<td>Mon 3/14/05</td>
<td>Fri 3/16/05</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>Paint case</td>
<td>3 days</td>
<td>Mon 3/21/05</td>
<td>Wed 3/23/05</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>Add to notes for user's manual</td>
<td>2 days</td>
<td>Mon 3/21/05</td>
<td>Tue 3/22/05</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>Test prototype against project specifications</td>
<td>2 days</td>
<td>Mon 3/21/05</td>
<td>Tue 3/22/05</td>
<td>59</td>
</tr>
<tr>
<td>64</td>
<td>Make necessary adjustments to motor-linear actuator</td>
<td>3 days</td>
<td>Wed 3/23/05</td>
<td>Fri 3/25/05</td>
<td>43</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>Predecessors</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>Make sure functioning prototype will fit in case</td>
<td>2 days</td>
<td>Thu 3/31/05</td>
<td>Fri 4/1/05</td>
<td>66</td>
</tr>
<tr>
<td>66</td>
<td>Make necessary adjustments to case</td>
<td>3 days</td>
<td>Mon 3/28/05</td>
<td>Wed 3/30/05</td>
<td>59</td>
</tr>
<tr>
<td>67</td>
<td>Finalize working prototype</td>
<td>5 days</td>
<td>Mon 4/4/05</td>
<td>Fri 4/8/05</td>
<td>65</td>
</tr>
<tr>
<td>68</td>
<td>Add to notes for user's manual</td>
<td>2 days</td>
<td>Mon 3/28/05</td>
<td>Tue 3/29/05</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>Begin writing final report</td>
<td>5 days</td>
<td>Mon 3/28/05</td>
<td>Fri 4/1/05</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>Begin constructing final presentation</td>
<td>5 days</td>
<td>Mon 4/4/05</td>
<td>Fri 4/8/05</td>
<td>69</td>
</tr>
<tr>
<td>71</td>
<td>Continue any necessary testing and adjustments on prototype</td>
<td>5 days</td>
<td>Mon 4/4/05</td>
<td>Fri 4/8/05</td>
<td>60</td>
</tr>
<tr>
<td>72</td>
<td>Continue writing final report</td>
<td>5 days</td>
<td>Mon 4/4/05</td>
<td>Fri 4/8/05</td>
<td>69</td>
</tr>
<tr>
<td>73</td>
<td>Update website with pictures of construction and prototype</td>
<td>2 days</td>
<td>Wed 2/8/05</td>
<td>Thu 2/7/05</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>Begin compiling users manual for prototype</td>
<td>5 days</td>
<td>Mon 4/11/05</td>
<td>Fri 4/15/05</td>
<td>72</td>
</tr>
<tr>
<td>75</td>
<td>Finish final report</td>
<td>5 days</td>
<td>Mon 4/11/05</td>
<td>Fri 4/15/05</td>
<td>72</td>
</tr>
<tr>
<td>76</td>
<td>Finish final presentation</td>
<td>5 days</td>
<td>Mon 4/11/05</td>
<td>Fri 4/15/05</td>
<td>75</td>
</tr>
<tr>
<td>77</td>
<td>Finish posting information on website</td>
<td>5 days</td>
<td>Mon 4/11/05</td>
<td>Fri 4/15/05</td>
<td>75</td>
</tr>
<tr>
<td>78</td>
<td>Continue compilation of users manual</td>
<td>4 days</td>
<td>Mon 4/18/05</td>
<td>Thu 4/21/05</td>
<td>74</td>
</tr>
<tr>
<td>79</td>
<td>Finish user's manual</td>
<td>1 day</td>
<td>Fri 4/22/05</td>
<td>Fri 4/22/05</td>
<td>75</td>
</tr>
<tr>
<td>80</td>
<td>Finishing touches to project</td>
<td>2 days</td>
<td>Mon 4/18/05</td>
<td>Tue 4/19/05</td>
<td></td>
</tr>
</tbody>
</table>
## VIII. TOTAL COST

The total cost for the components of this design is illustrated in the table below.

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Description</th>
<th>Quantity</th>
<th>Unit Price</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jameco Electronics</td>
<td>20X1 Parallel LCD Display</td>
<td>1</td>
<td>$19.99</td>
<td>$19.99</td>
</tr>
<tr>
<td>Jameco Electronics</td>
<td>2X8 16-Button Keypad Switch</td>
<td>1</td>
<td>$16.95</td>
<td>$16.95</td>
</tr>
<tr>
<td>Jameco Electronics</td>
<td>Shipping and Handling</td>
<td>----</td>
<td>----</td>
<td>$5.00</td>
</tr>
<tr>
<td>Maxi Aids for Independent Living</td>
<td>Count-A-Dose 1/2cc</td>
<td>1</td>
<td>$49.95</td>
<td>$49.95</td>
</tr>
<tr>
<td>Maxi Aids for Independent Living</td>
<td>Shipping and Handling</td>
<td>----</td>
<td>----</td>
<td>$6.25</td>
</tr>
<tr>
<td>DigiKey</td>
<td>Unipolar Stepper Motor 5VDC 18Deg</td>
<td>1</td>
<td>$16.20</td>
<td>$16.20</td>
</tr>
<tr>
<td>DigiKey</td>
<td>PIC16F874A Microprocessor 40DIP</td>
<td>6</td>
<td>$7.15</td>
<td>$42.90</td>
</tr>
<tr>
<td>DigiKey</td>
<td>Shipping and Handling</td>
<td>----</td>
<td>----</td>
<td>$5.00</td>
</tr>
<tr>
<td>Mansfield Supply</td>
<td>Sq. Ft. 3/8” Plexiglass</td>
<td>2</td>
<td>$10.76</td>
<td>$21.51</td>
</tr>
<tr>
<td>Haydon Switches and Instruments</td>
<td>Linear Actuator with Encoder</td>
<td>1</td>
<td>$280.00</td>
<td>$280.00</td>
</tr>
<tr>
<td>Haydon Switches and Instruments</td>
<td>Driver</td>
<td>1</td>
<td>$168.20</td>
<td>$168.20</td>
</tr>
<tr>
<td>Haydon Switches and Instruments</td>
<td>Shipping and Handling</td>
<td>----</td>
<td>----</td>
<td>$20.00</td>
</tr>
<tr>
<td>Jameco Electronics</td>
<td>Pin Cap</td>
<td>1</td>
<td>$0.50</td>
<td>$0.50</td>
</tr>
<tr>
<td>Jameco Electronics</td>
<td>Header</td>
<td>10</td>
<td>$0.10</td>
<td>$1.00</td>
</tr>
<tr>
<td>Jameco Electronics</td>
<td>Shipping and Handling</td>
<td>----</td>
<td>----</td>
<td>$5.00</td>
</tr>
</tbody>
</table>

**TOTAL SPENT** $658.45

**AMOUNT UNDER BUDGET** $1,341.55

Note: the cost of production is expected to be far less than the cost of development.
IX. CONCLUSIONS

Products currently on the market do not accommodate patients that lack the fine motor skills necessary for correct use. This is perhaps the most important unique aspect of the project. Currently marketed products also lack a digital user interface, which the project described here will include. The product described here will provide a digital self-dosing device that will accommodate many of the physical limitations mentioned above, while remaining affordable and competitive in today’s market.

This design will be beneficial to patients in need of an accurately dosage syringe that is not labor intensive, does not require a lot of physical exertion. This is one of the major factors in the design, to have an easily readable and usable user interface that will accommodate both hearing and visual impaired along with patients that lack motor skills. This is why the design implements a control circuit which controls the movements of the electro-mechanical devices that will pull back the syringe filling it with the appropriate amount of medication.

This design of the product will be ready for testing by March 2005. The members of Team #1 are excited to be a part of such a meaningful and necessary product. Our team looks forward to the opportunity to move forward with the design described here.

X. REFERENCES

The following websites were instrumental in researching and designing our project:

http://www.emu8086.com/vb/index_asm.html
http://www.digikey.com
http://www.engr.uconn.edu/cse/Courses/CSE210W/
http://www.netmotion.com
http://www.nfb.org/vodold/inslmeas.htm
http://www.hsi-inc.com

XI. ACKNOWLEDGEMENTS

Special thanks to the following parties for supporting Team 1 during the design process:

- The 2004-2005 National Student Design Contest, funded by the Rehabilitation Engineering Research Center on Accessible Medical Instrumentation
- Dr. Enderle
- Chris Liebler
- Haydon Switch and Instrument, Inc.