ACCESSIBLE SYRINGE DOSING DEVICE

Team #1
Megan Bowers
Karen Young
Bob Mock
Gabe Goldman
BME 290 Design III
December 8, 2004
OUTLINE

I. Introduction

II. Design
   A. Top View (Internal)
   B. Side View (Internal)
   C. User Interface Panel (External)
   D. Back View (External)
   E. Front View (External)
   F. Bottom View (External)
   G. Overall Flow Chart

III. Equipment and Analysis
   A. Motor Selection
   B. Microprocessor
   C. Ball-Screw and Stage Assembly
   D. Part Schematics
   E. Motor, Ball-Screw, and Stage Mechanism Setup
      a. Relevant Calculations
      b. Syringe Calculations
      c. Ball-Screw Calculations
      d. Motor Calculations
   F. LabVIEW Simulation

IV. Total Cost

V. Realistic Constraints and Safety Issues

VI. Conclusion
I. INTRODUCTION

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.

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.

II. DESIGN

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. 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:**

![Internal Views Diagram]
External Views:

Side View (Cross-Section)

Control Circuit
W FROM XX
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

- **Motor**
  - Used to rotate ball-screw mechanism

- **Ball -Shaft**

- **Stage**
  - Pulls to fill with insulin or heparin into syringe

- **C-LCD Display**
III. EQUIPMENT AND ANALYSIS

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/
Of the various types of stepping motors we have chosen a unipolar for our design. The basic design and specifications are depicted below. Note that the major 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 are much easier to control using a binary-based control circuit, rather than three control circuits, simply because of the ease of converting from a base-two to a base-four number. 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.

Source: [http://www engr uconn edu/cse/Courses/CSE210W/](http://www.engr.uconn.edu/cse/Courses/CSE210W/)
3. 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.

**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>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>44-TQFP</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>
**Example Code: PIC Code Relevant to Stepper-Motor Control**

The following program, taken from [http://www.emu8086.com/vb/index_asm.html](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.

```assembly
; 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
    db 011b
    db 000b

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

#include <16c74.h>
#fuses HS,NOWDT,NOPROTECT
#use delay(clock=20000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7) // Jumpers: 8 to 11, 7 to 12

#include <input.c>

#define FOUR_PHASE TRUE

#if #define FOUR_PHASE

byte const POSITIONS[4] = {0b0101, 0b1001, 0b1010, 0b0110};
#else
byte const POSITIONS[8] = {0b0101, 0b0001, 0b1001,
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("\r\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;
```c
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);

    drive_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.**
Part Schematics

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

Microprocessor

PDIP (40 pin)

PIC16F877A7/874A
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
Motor Ball-Shaft, and Stage Mechanism 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:
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
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 a full rotation. Dividing eighteen degrees into three hundred and sixty degrees gives the percent rotation per step. This value is five percent.
**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:**

![Syringe Dosing Simulation Diagram](image)

- **Plus Minus:**
  - 40

- **On/Off Button:**

- **Syringe 1cc total:**
  - 0 to 100

- **Bottle of Insulin (1cc total):**
  - 0 to 100
IV. TOTAL COST

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

<table>
<thead>
<tr>
<th>Part</th>
<th>Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter chip</td>
<td>0.60</td>
</tr>
<tr>
<td>Motor</td>
<td>18.00</td>
</tr>
<tr>
<td>Display (Curtis Instruments)</td>
<td>22.00</td>
</tr>
<tr>
<td>Keypad</td>
<td>19.53</td>
</tr>
<tr>
<td>Express PCB miniboard</td>
<td>51.00 for 3—17.00 each</td>
</tr>
<tr>
<td>PIC16F874A</td>
<td>8.23</td>
</tr>
<tr>
<td>Ball-shaft</td>
<td>15.27</td>
</tr>
<tr>
<td>Stage</td>
<td>12.73</td>
</tr>
<tr>
<td>TOTAL</td>
<td>113.36 (1 board); 147.36 (3 boards)</td>
</tr>
</tbody>
</table>

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

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.

Special thanks to the RERC for funding this and other projects through the 2004-2005 National Student Design Contest.