

Binary to Decimal Machine



Submitted by:
Dennis Schmitt
Curtis Stovall
Seth Bell

ME307
Mechatronics
Colorado State University
December 9, 2003

Table of Contents

Table of Contents.....	i
List of Figures.....	ii
List of Tables.....	iii
List of Tables.....	iii
Design Summary.....	1
Design Details.....	4
Functional Diagram.....	6
Device Pictures.....	7
Software Flowchart.....	14
Electrical Schematics.....	15
Input/Output Map.....	17
Project Costs.....	18
Appendix A – Software.....	19
Appendix B – Mechanical Schematics.....	23

List of Figures

Figure 1 Binary to Decimal Machine.....	2
Figure 2 LED's and LCD with Initial Message	2
Figure 3 LCD Message After Start Button is Pushed.....	3
Figure 4 LCD Display with Decimal Guess	3
Figure 5 Functional Diagram.....	6
Figure 6 Inside Main Control Box	7
Figure 7 Close-up of Main Circuit Board.....	7
Figure 8 Rear View of Front Panel.....	8
Figure 9 Side View of Remote Board and LED's.....	8
Figure 10 Side View with Control Box in Place.....	9
Figure 11 Front Panel.....	9
Figure 12 Keypad.....	10
Figure 13 Keypad 15 Pin D-Sub Connector	10
Figure 14 Photo Sensor and Motor	11
Figure 15 Front View of Motor Drive Assembly	11
Figure 16 Top View of Motor and Dispensing Assembly.....	12
Figure 17 Motor Drive Assembly Components.....	12
Figure 18 Keypad, Control Box, and Motor/Sensor Connectors.....	13
Figure 19 Computer to Control Box Programming Interface.....	13
Figure 20 Software Flowchart	14
Figure 21 15-Pin D-Sub Connector for Keypad	16
Figure 22 Initial Concept of Front Panel	23
Figure 23 Initial Concept of Drive Mechanism	23
Figure 24 Schematic of Layout Drawing for Sheetmetal	24
Figure 25 Schematic of Layout for Front Panel.....	24
Figure 26 Schematic for Motor/Sensor/Drive Shaft Assembly	25
Figure 27 Custom Drive Shaft	25
Figure 28 Drive Gear Example	26

List of Tables

Table 1 Input/Output Map for Electrical Schematic.....	17
Table 2 Project Cost Sheet.....	18

Design Summary

The project is designed as a learning toy with a reward for determining the correct answer (see Figure 1). The object is to represent a random number, from 1 to 255, in binary form by using eight LEDs. The user enters the decimal form of the number using a keypad. The user's guess is displayed on an LCD as the guess is entered. Once the guess is entered, by pressing the enter key, the guess would then be compared to the actual number. If the guess matches the actual number, a motor activates, turning a shaft to the gumball machine that releases a gumball or candy. If the guess does not match the actual number, a digital sound module is activated and a sound is played through a speaker. The program then starts over and asks the user to press the start button.

The program is controlled by a Microchip brand PIC microcontroller. The PIC generates a random 16-bit number that is converted into an 8-bit number by the software. The program then turns on the corresponding outputs for the LEDs. This displays the number in binary (1 corresponding with a lighted LED and 0 with an unlighted LED). The LCD then displays a message prompting the user to enter a guess (see Figures 2 and 3). The user then enters a guess with the keypad (see Figure 4). The program allows for three digits to be entered and then looks for the enter button. It also allows the entered digits to be cleared from memory when the clear button is pressed. The program stores the number in an array as three separate digits and then turns them into one number when the enter button is pressed. This number is compared to the randomly generated number. As stated before, if the numbers match the motor for the gumball machine is activated. The motor is geared so that one revolution of the motor's shaft will make the gumball machine give out one gumball. This corresponds to a gearing ratio of 3:1; three revolutions of the motor shaft will produce one full revolution of the dispensing wheel. When the motor has made one revolution, a photo sensor is used to detect a metal flag attached to the shaft and this input to the PIC stops the motor. If the numbers entered do not match the stored number, a pulse is sent to the digital playback module, which will play the sound once.



Figure 1 Binary to Decimal Machine



Figure 2 LED's and LCD with Initial Message



Figure 3 LCD Message After Start Button is Pushed

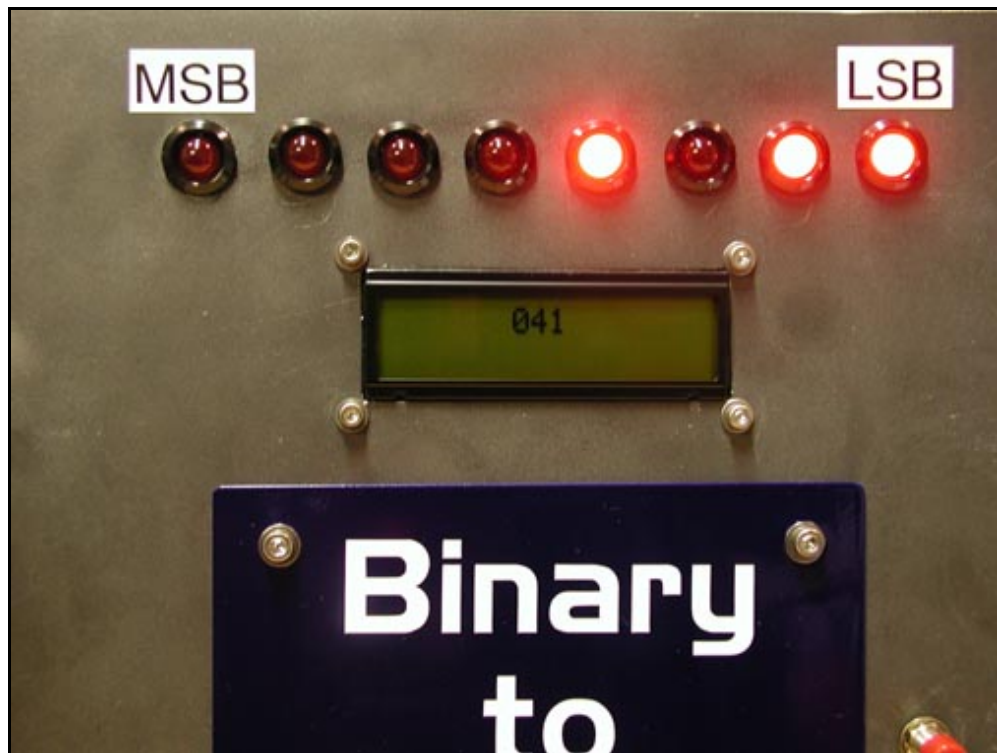


Figure 4 LCD Display with Decimal Guess

Design Details

As the functional diagram indicates (see Figure 5), there are two functional areas that input information to the PIC controller, the motor sensor and the keypad/start switch. The PIC outputs signals to control four functional areas, the motor, sound module, LCD display, and LED's. One of the most unique features of this project's design is the in-circuit programming capability. The computer is capable of programming directly to the PIC controller via an RS232 interface chip. This allows for very quick program checks and aids in reducing troubleshooting time.

The main circuit board, 24V power supply, and audio amplifier are located in the controller box (see Figures 6 and 7). The 24V power supply is converting the line voltage (120 V AC) to 24V DC. The main circuit board has the following major components:

- 18F452 PIC Microcontroller
- 10MHz External Oscillator
- Max233 RS232 Serial Communications Chip
- 5V Regulator
- 9V Regulator
- 12V Regulator
- Solid State Relays
- Digital Sound Module

The controller box has the power supply mounted at the bottom and the main circuit board is mounted in an inverted fashion on the aluminum lid (with nylon standoffs). The aluminum lid also has the main power switch, a power indicator (blue LED), a hardware reset switch, heartbeat LED, and a volume control (see Figure 10). The machine is interfaced with the control box via a 34-conductor ribbon cable. This cable connects the control box to the remote circuit board (see Figure 8).

The remote board has the final interface for the LCD, 8 LED's, start switch, motor, keypad, and photo sensor (see Figure 9). Ribbon cable connectors, header style connectors, and a 15-pin D-Sub connector were used for the motor and sensor wiring, the bank of LED's/LCD, and the keypad (see Figure 18). The start switch, LED's, and LCD are all mounted on the front panel

(see Figure 11). The keypad is connected remotely with a ribbon cable and the 15-pin D-Sub connector (see Figures 12 and 13).

The motor, when activated, allows a gumball (or candy) to be released after one revolution of the motor shaft. The motor stops when the sensor flag interrupts the infra-red light beam being transmitted by the infra-red LED, located in the sensor housing (see Figure 14). A matching gear, bolted to the belt-driven shaft, rotates the internal disc of the machine (see Figures 15-17).

If a wrong answer is entered, then a digitally stored sound is turned on. This sound is stored on the digital sound module and then is amplified to an appropriate volume. The volume is controlled by a potentiometer located on the main control box.

To program the PIC microcontroller, a computer is used to interface directly. A laptop with all of the necessary programming and communications software enables the programming to be done with the chip in the circuit. The computer connects to the control box with an industry standard null modem serial cable (see Figure 19).

Functional Diagram

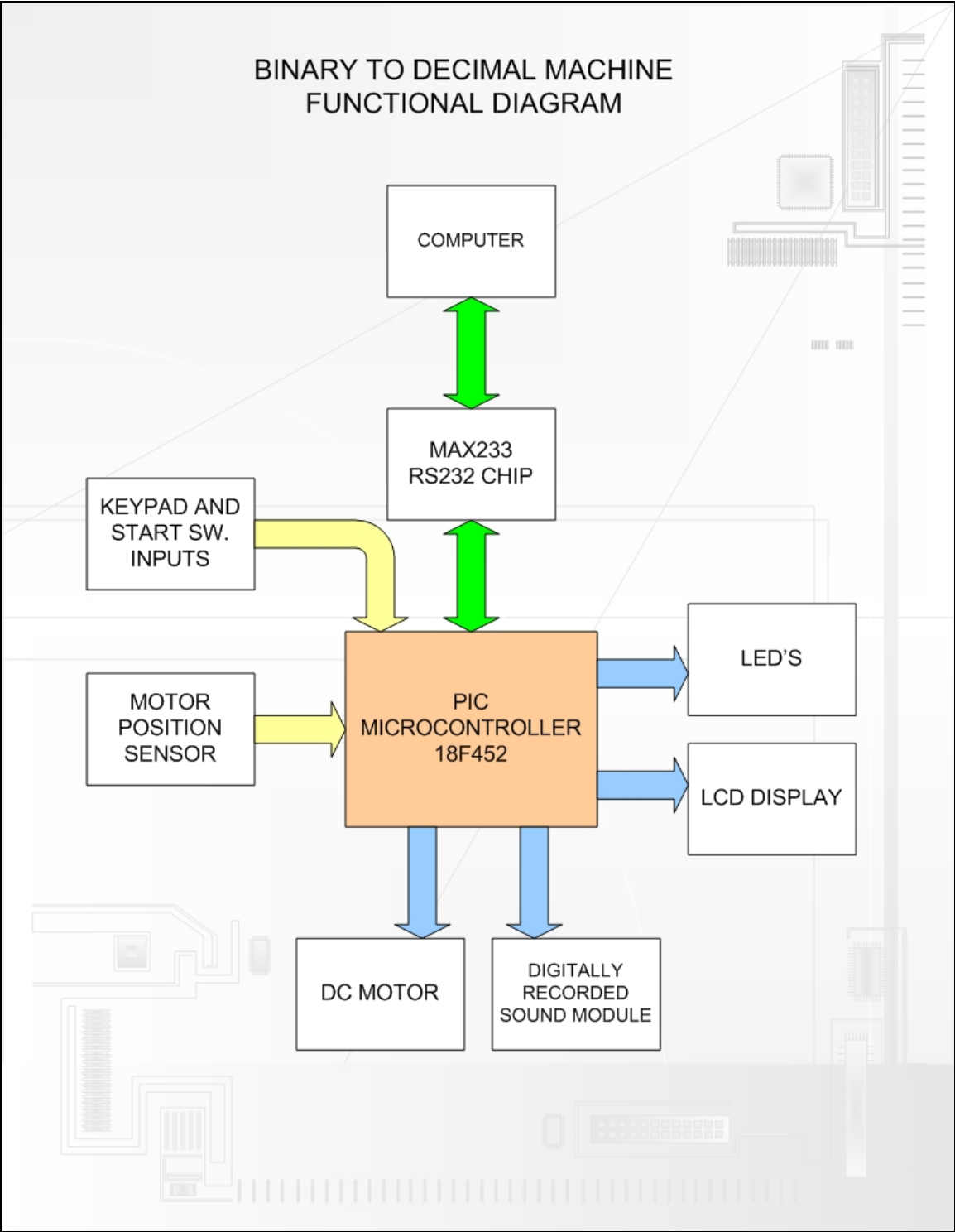


Figure 5 Functional Diagram

Device Pictures

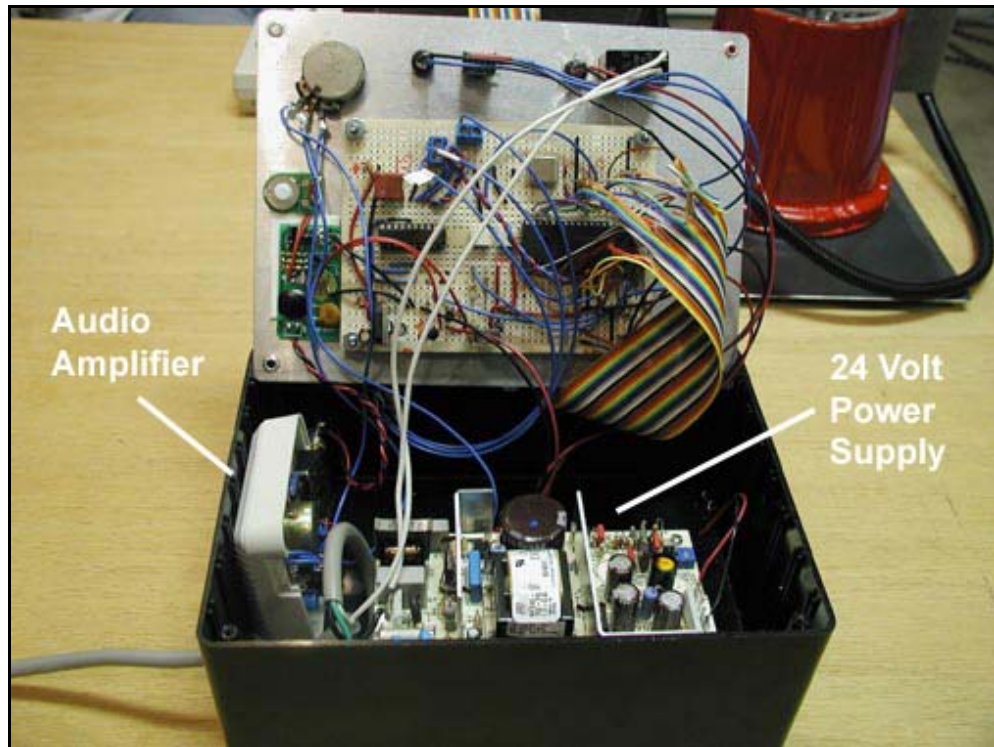


Figure 6 Inside Main Control Box

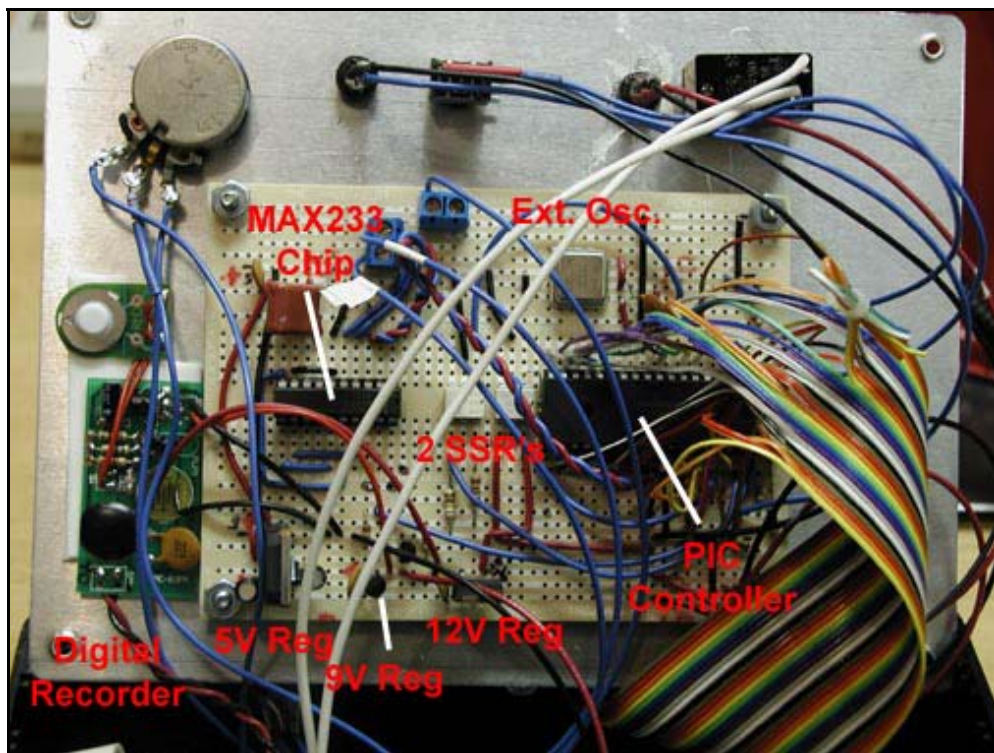


Figure 7 Close-up of Main Circuit Board

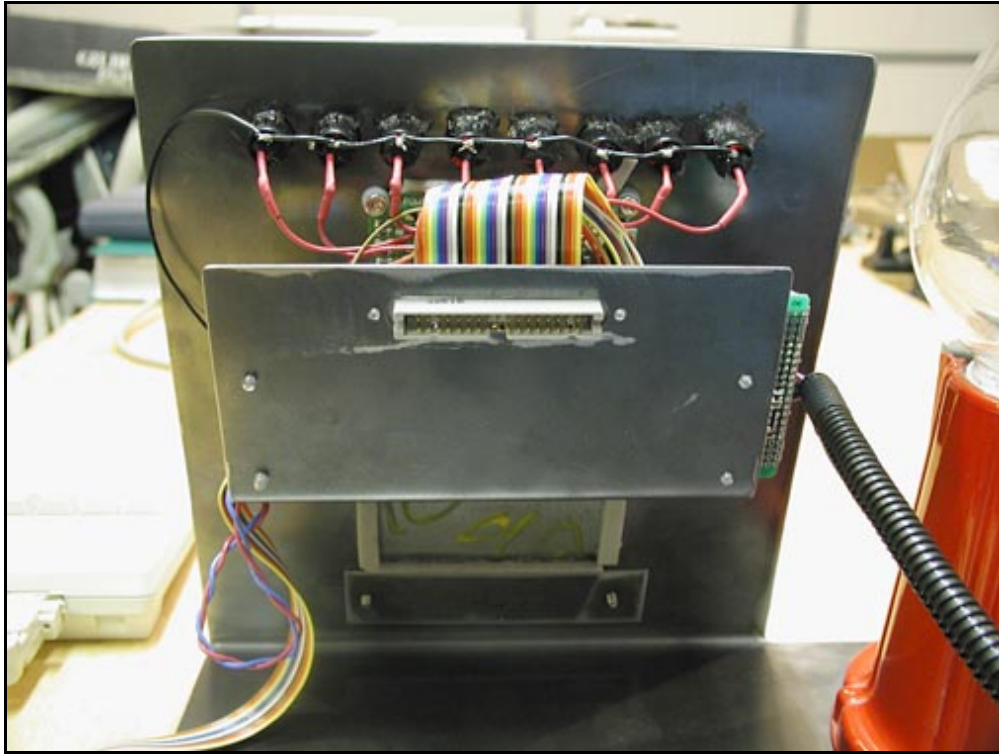


Figure 8 Rear View of Front Panel

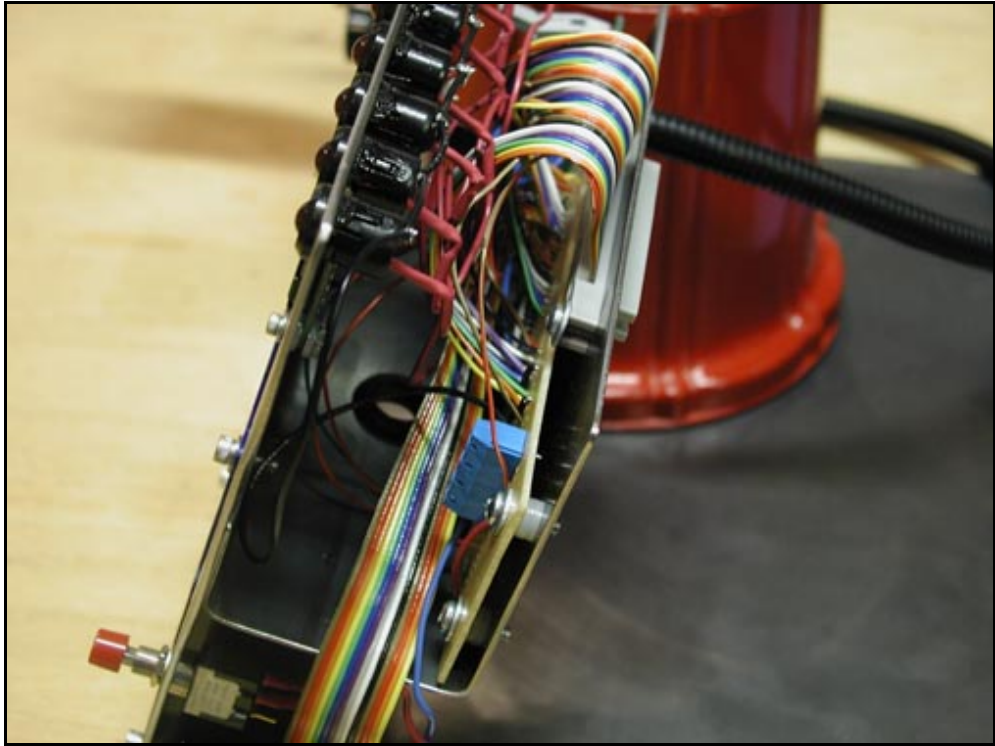


Figure 9 Side View of Remote Board and LED's

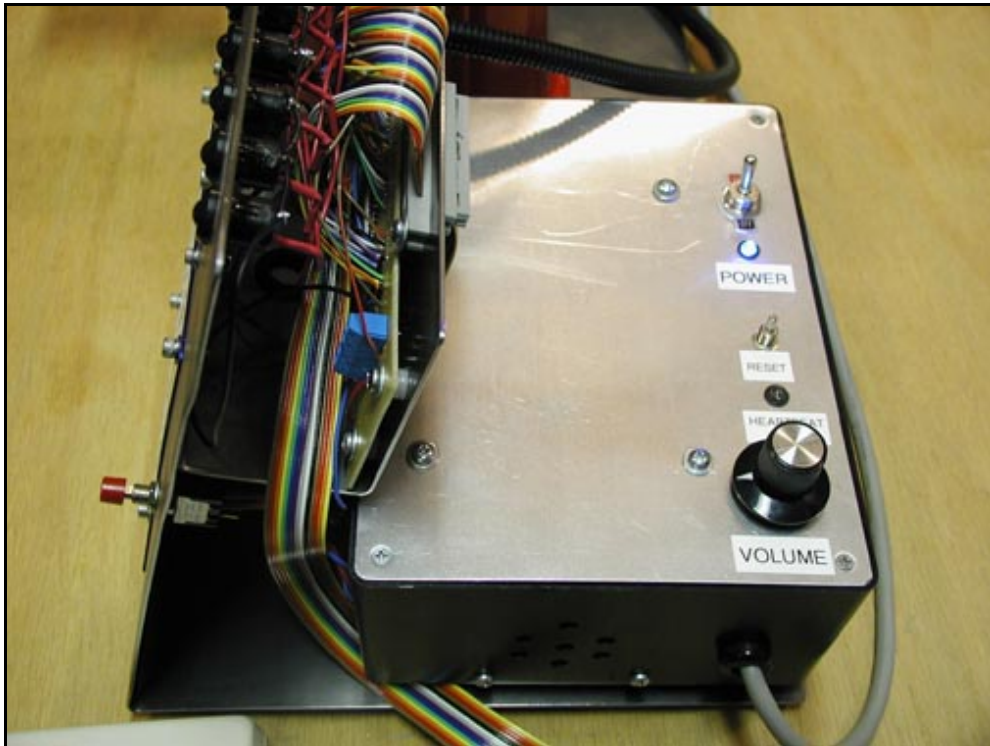


Figure 10 Side View with Control Box in Place



Figure 11 Front Panel



Figure 12 Keypad



Figure 13 Keypad 15 Pin D-Sub Connector

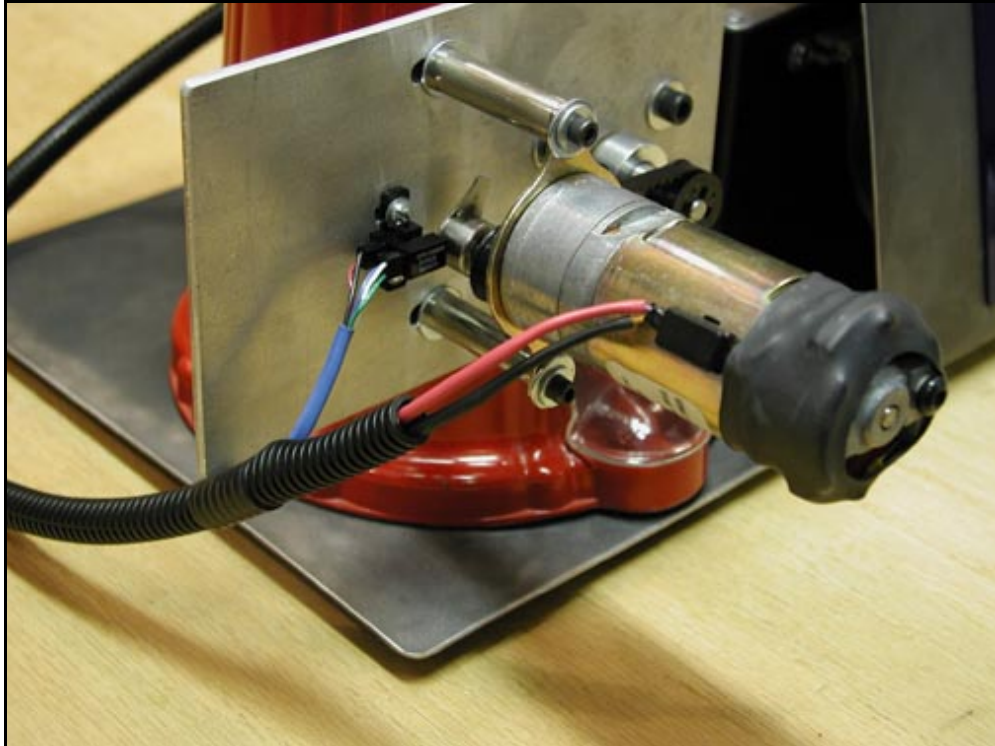


Figure 14 Photo Sensor and Motor

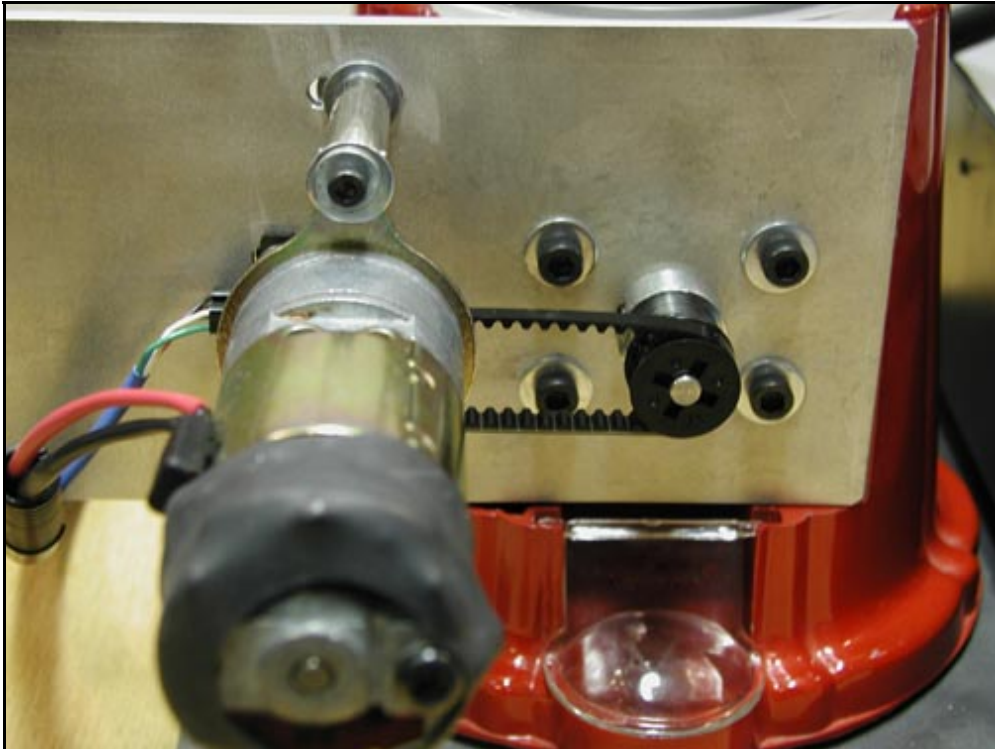


Figure 15 Front View of Motor Drive Assembly

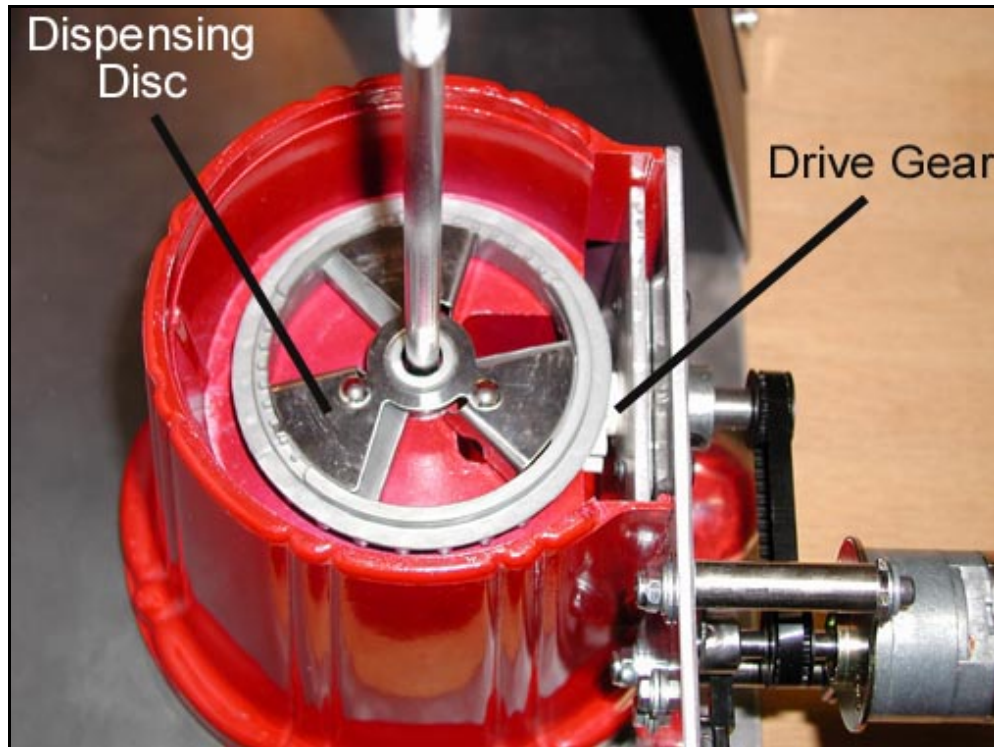


Figure 16 Top View of Motor and Dispensing Assembly



Figure 17 Motor Drive Assembly Components

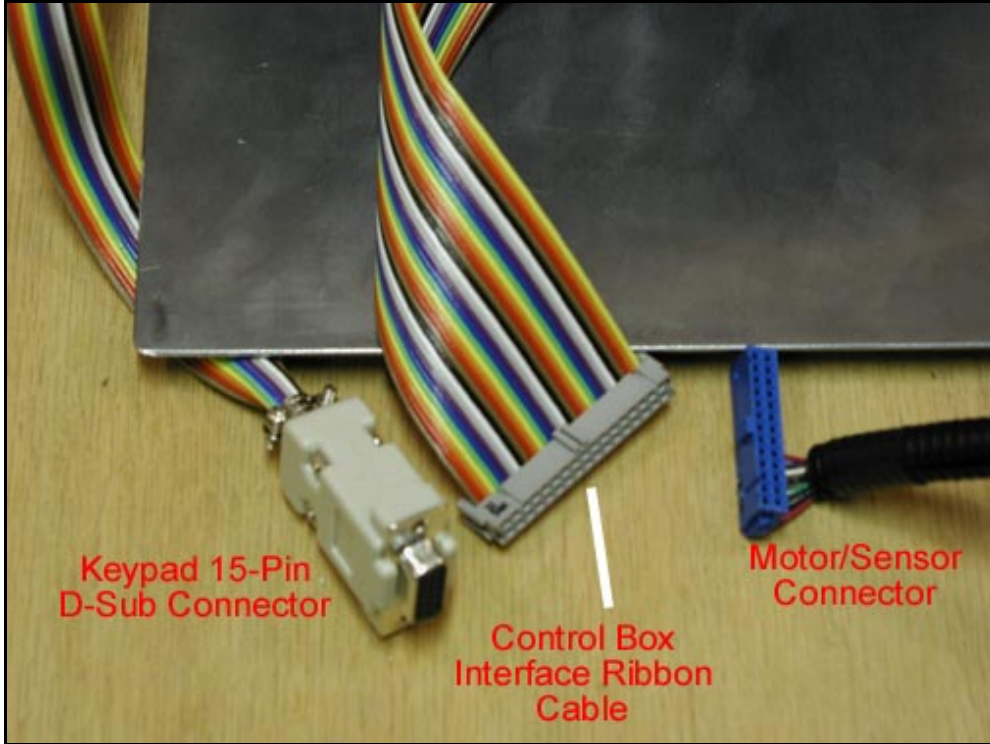


Figure 18 Keypad, Control Box, and Motor/Sensor Connectors



Figure 19 Computer to Control Box Programming Interface

Software Flowchart

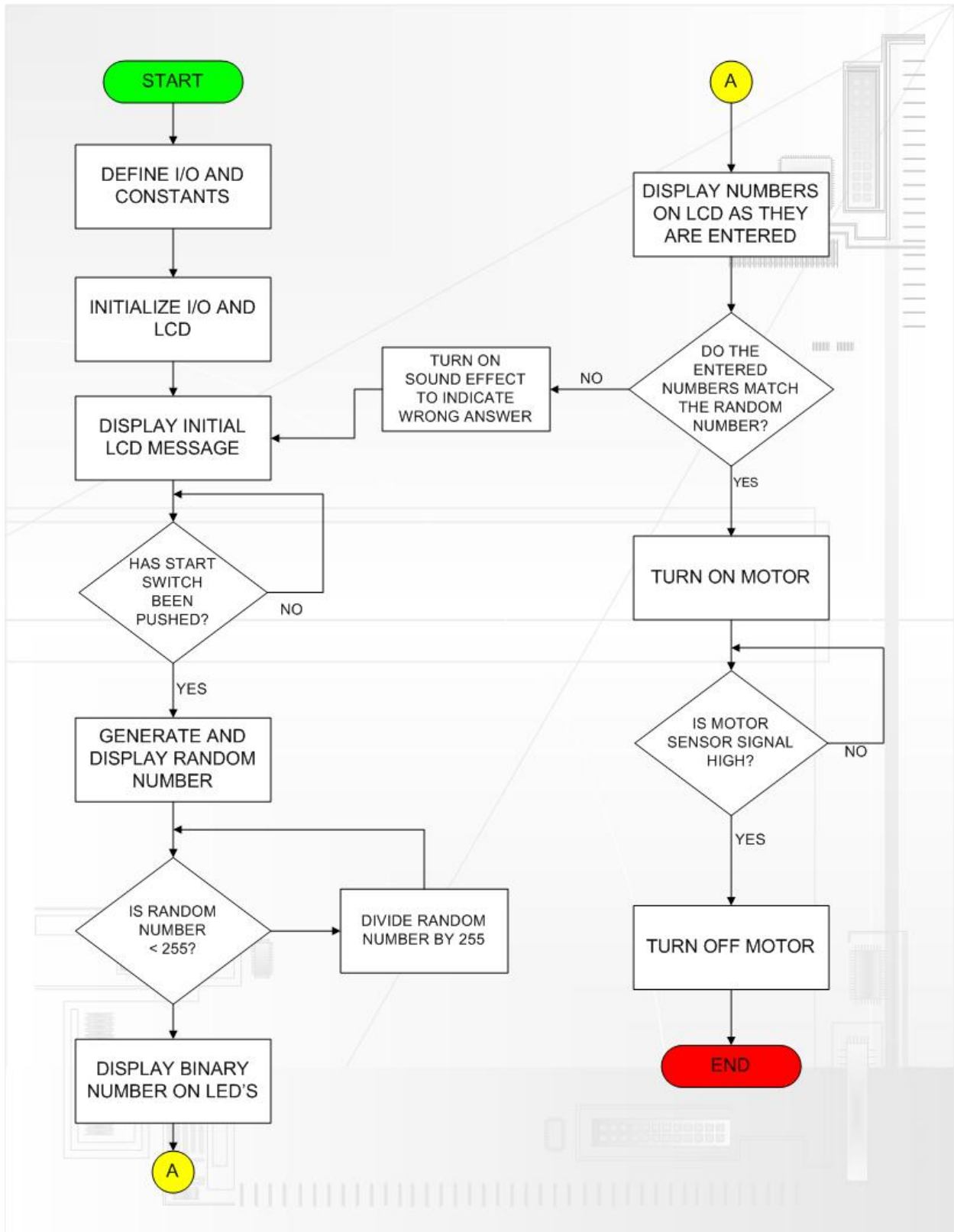


Figure 20 Software Flowchart

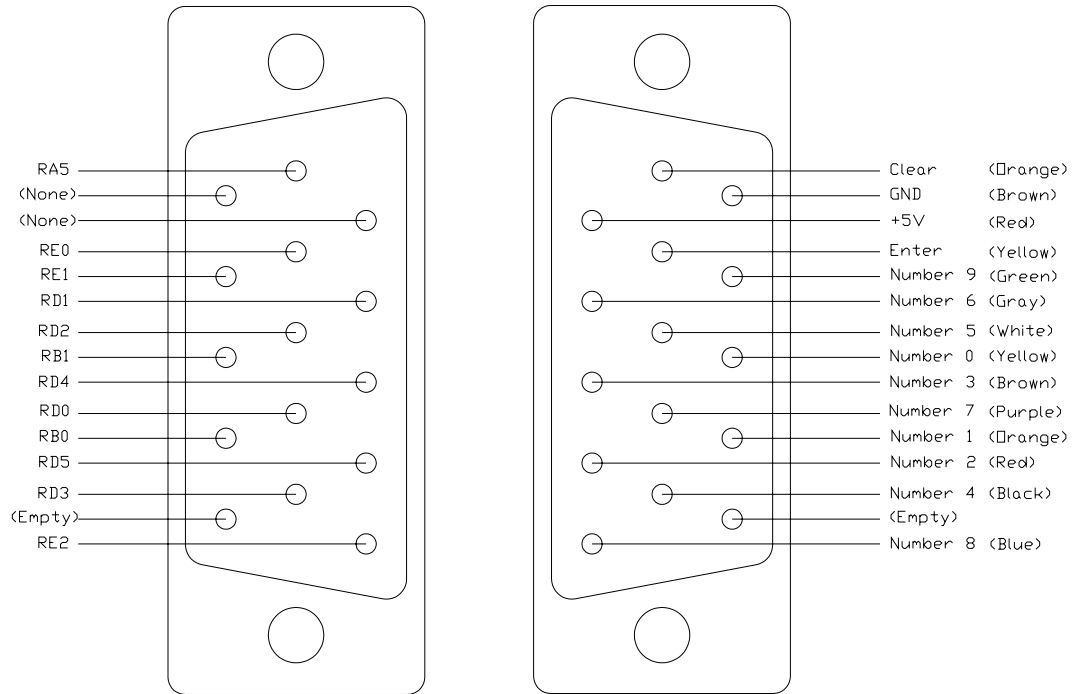


Figure 21 15-Pin D-Sub Connector for Keypad

Input/Output Map

This table is used to help navigate through the assembly process of the circuit board as well as assist the programmer. The wire color algorithm is such that the first wire (of the 34 conductor ribbon cable) is on the left, and in this case, it is the brown wire. The colors advance from left to right.

I/O Map			
Description	I/O	Wire Color	15 Pin D-Sub Connector Pin #
Start Sw	RB7	Brown	
+5 V		Red	1
N/C	RA4	Orange	
LCD Serial	RB3	Yellow	
N/C	RA0	Green	
N/C	RA1	Blue	
N/C	RA2	Purple	
N/C	RA3	Grey	
N/C		White	
DC Com		Black	11
Photo Sensor	RB5	Brown2	
Motor	RB2	Red2	
Sound	RB4	Orange2	
N/C		Yellow2	
Led (LSB)	RC0	Green2	
Led	RC1	Blue2	
Led	RC2	Purple2	
Led	RC3	Grey2	
Led	RC4	White2	
Led	RC5	Black2	
Led	RC6	Brown3	
Led (MSB)	RC7	Red3	
+ (CLR)	RA5	Orange3	6
Enter	RE0	Yellow3	7
9	RE1	Green3	12
8	RE2	Blue3	5
7	RD0	Purple3	9
6	RD1	Grey3	2
5	RD2	White3	8
4	RD3	Black3	10
3	RD4	Brown4	3
2	RD5	Red4	6
1	RB0	Orange4	14
0	RB1	Yellow4	13

Table 1 Input/Output Map for Electrical Schematic

Project Costs

This is the itemized list of components used to build the Binary to Decimal Machine. As can be seen, many of the higher cost items were donated for the project. Some of these donations came from the manufacturer and others were from a local company.

Component	Quantity	Price	Total Cost
Gumball Machine	1	dontated	\$0.00
PIC 18F452	3	dontated	\$0.00
LCD Display	1	dontated	\$0.00
Key Pad	1	dontated	\$0.00
DC Motor	1	\$7.50	\$7.50
LED's	10	\$0.68	\$6.80
Digital Sound Recorder	1	\$9.99	\$9.99
Photo Interrupter	2	dontated	\$0.00
MAX233 RS232 Chip	2	dontated	\$0.00
24V Power Supply	1	dontated	\$0.00
12V regulator	1	\$1.99	\$1.99
9V regulator	1	\$2.08	\$2.08
5V regulator	1	\$7.69	\$7.69
Sound Amplifier	1	dontated	\$0.00
Solid State Relay	2	dontated	\$0.00
10 MHz Oscillator	1	dontated	\$0.00
Serial Input Components	1	dontated	\$0.00
Button Switches	3	\$0.45	\$1.35
0.1 mF Capacitor	5	\$0.09	\$0.45
1.0 mF Capacitor	1	\$0.99	\$0.99
100W Resistor	2	\$0.15	\$0.30
330W Resistor	1	\$0.06	\$0.06
680W Resistor	3	\$0.15	\$0.45
1KW Resistor	5	\$0.06	\$0.30
330W Resistor Pack	2	\$0.60	\$1.20
680W Resistor Pack	2	\$0.60	\$1.20
2N3904 Transistor	1	donated	\$0.00
Proto Board (Small)	1	\$2.99	\$2.99
Proto Board (Large)	1	\$13.90	\$13.90
Metal	1	\$10.48	\$10.48
Fasteners	1	\$9.72	\$9.72
Dsub Connectors	1	\$5.52	\$5.52
Bronze Bushing	1	\$0.58	\$0.58
Gears	1	\$16.00	\$16.00
20 Pin IC Socket	1	\$1.04	\$1.04
40 Pin IC Socket	1	\$2.09	\$2.09
Total			\$104.67

Table 2 Project Cost Sheet

Appendix A – Software

```
*****
'* Name      : MAIN.BAS *
'* Author    : [set under view...options] *
'* Notice    : Copyright (c) 2003 [set under view...options] *
'*           : All Rights Reserved *
'* Date      : 7/25/2003 *
'* Version   : 1.0 *
'* Notes     : *
'*           : *
*****

    DEFINE LOADER_USED 1
    Include "Modedefs.Bas"
    Define OSC 10          ' Set Xtal Frequency @ 10mHZ
    ADCON1 = 7            ' Turn off the PIC A/D

' ** Define Variables **

    key0      var portb.1
    key1      var portb.0
    key2      var portd.5
    key3      var portd.4
    key4      var portd.3
    key5      var portd.2
    key6      var portd.1
    key7      var portd.0
    key8      var porte.2
    key9      var porte.1
    eqlbtn    var porte.0
    clrbtn    var porta.5
    allled    var portc
    startbtn  var portb.7
    heart     var portb.6
    photo     var portb.5
    speaker   var portb.4
    motor     var portb.2
    LCD       var portb.3
    guess     var byte
    number    var byte
    rnumber   var word
    ggray     var byte[3]
    num       var byte

' Define some constants that will be needed for the LCD
I          CON 254
CLR        CON 1
LINE1     CON 129
LINE2     CON 193
RIGHT     CON 20
```

```

'Define I/O Ports

    TRISA = %11111111
    TRISB = %10100011
    TRISC = %00000000
    TRISD = %10111111
    TRISE = %00000111

'-----*
'                ** THE MAIN PROGRAM STARTS HERE **
'*****
` sets all outputs low
low heart
low motor
low speaker
low LCD
low allled
allled = 0

pause 1000                'pause to initialize LCD

FRONT:

grray[0] = 15
grray[1] = 15                'initialize guess array (this prevents grray
grray[2] = 15                'from being entered twice later in the program)

    serout LCD,N9600,[I,CLR]
    serout LCD,N9600,[I,LINE1]                `display message on LCD
    serout LCD,N9600,["PUSH RED BUTTON"]
    SEROUT LCD,N9600,[I,LINE2]
    SEROUT LCD,N9600,["    TO START    "]

GO:
If startbtn == 1 Then        'see if user has responded, keeps checking
    Random number            'until button is pressed
    if number < 255 then
        goto displesds        `after button has been pressed, generates
    else                        `random 16-bit number and changes it to
        number = number / 255    `an 8-bit number by dividing it
    endif

    if number < 255 Then
        goto displesds
    else
        number = number / 255
        goto displesds
    endif
else
    goto GO
endif

DISPLEDS:
    allled = number            'display number on leds as a binary number

```

```

DGTCHK:
SEROUT LCD,N9600,[I,CLR]           'displays 2nd message on LCD,
SEROUT LCD,N9600,[I,LINE1]        'telling user to enter a 3 digit
SEROUT LCD,N9600,["TYPE A 3 DIGIT"] 'number and then press enter
SEROUT LCD,N9600,[I,LINE2]
SEROUT LCD,N9600,["GUESS & ENTER"]
pause 2000
SEROUT LCD,N9600,[I,CLR]
SEROUT LCD,N9600,[I,LINE1 + 5]    'clear lcd and move cursor to middle of
                                  'first line

For num = 0 to 2                   'lets user enter guess
  GOSUB checkkey
  pause 200                        'debounce
  SEROUT LCD,N9600,[gsray[num] + 48] 'display typed number
next
goto chkentr

CHKENTR:
If eqlbtn == 0 then
  guess = (100 * gsray[0]) + (10 * gsray[1]) + gsray[2] 'convert from
  goto chkguess                                         'an array to a number
endif                                                    'when enter is pressed

if clrbtn == 0 then                                     'checks for clrbtn if enter
  gsray[0] = 15                                         'button hasn't been pressed
  gsray[1] = 15
  gsray[2] = 15
  SEROUT LCD,N9600,[I,CLR]
  SEROUT LCD,N9600,[I,LINE1]
  goto DGTCHK
Else
  goto chkentr

endif

CHKGUESS:
If guess == number then 'compares the guess number to the generated
  goto motoron          'number and tells PIC which output to activate
else
  goto soundfx
endif

MOTORON:
pause 100                'activates motor and tells it to wait so
high motor               'sensor flag can get clear, then tells it to
pause 500                'start looking for the flag
low motor
  while (photo <> 0)
    high motor
  wend
low motor

allled = 0               'turns LEDs off
guess = 0                'resets guess
goto FRONT

```

```

SOUNDFX:
  high speaker          'sends pulse to speaker to activate it
  pause 100
  low speaker
  allled = 0
  guess = 0
  goto FRONT
end

'*****
'subroutines
'*****

CHECKKEY:
'checks to see which button is being pressed and then stores the value

  While gsray[num] == 15    'makes sure the gosub happens only once

    If key0 == 0 THEN gsray[num] = 0
    If key1 == 0 THEN gsray[num] = 1
    If key2 == 0 THEN gsray[num] = 2
    If key3 == 0 THEN gsray[num] = 3
    If key4 == 0 THEN gsray[num] = 4
    If key5 == 0 THEN gsray[num] = 5
    If key6 == 0 THEN gsray[num] = 6
    If key7 == 0 THEN gsray[num] = 7
    If key8 == 0 THEN gsray[num] = 8
    If key9 == 0 THEN gsray[num] = 9

    if clrbtn == 0 then
      gsray[0] = 15
      gsray[1] = 15
      gsray[2] = 15
      serout LCD,N9600,[I,CLR]
      SEROUT LCD,N9600,[I,LINE1]
      goto dgtchk
    endif
  wend
Return

```

Appendix B – Mechanical Schematics

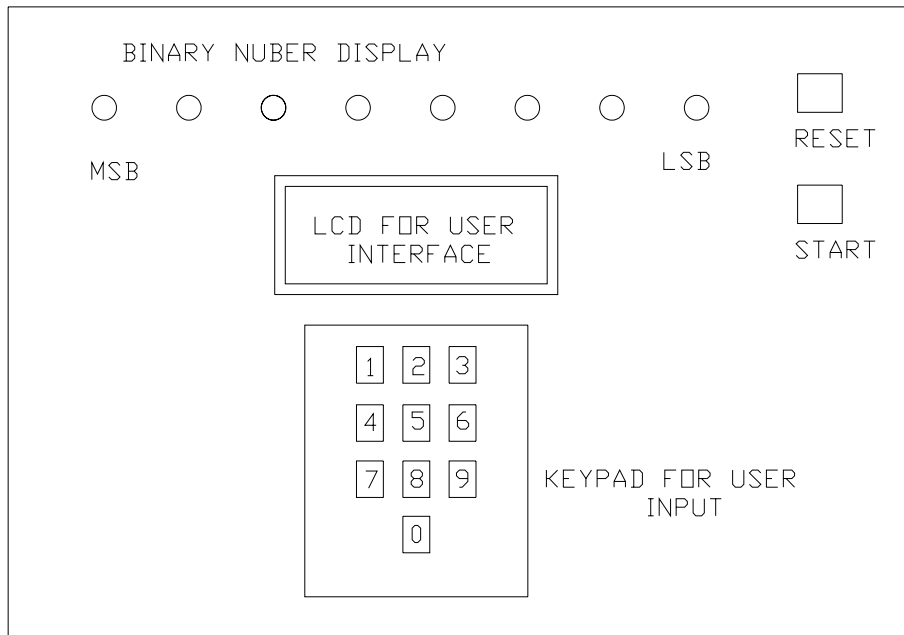


Figure 22 Initial Concept of Front Panel

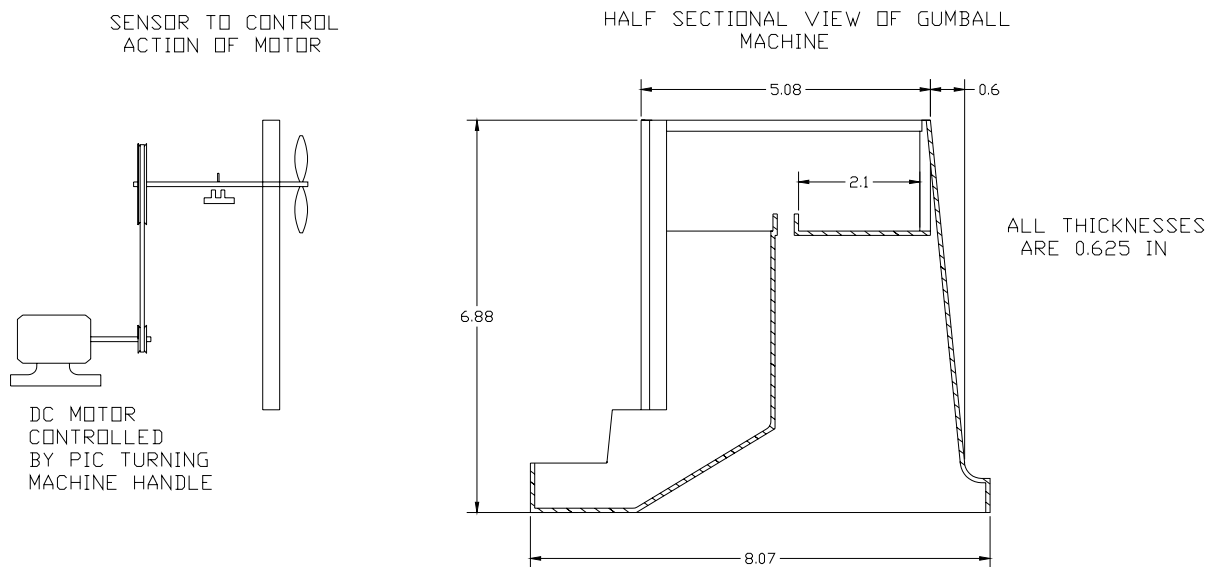


Figure 23 Initial Concept of Drive Mechanism

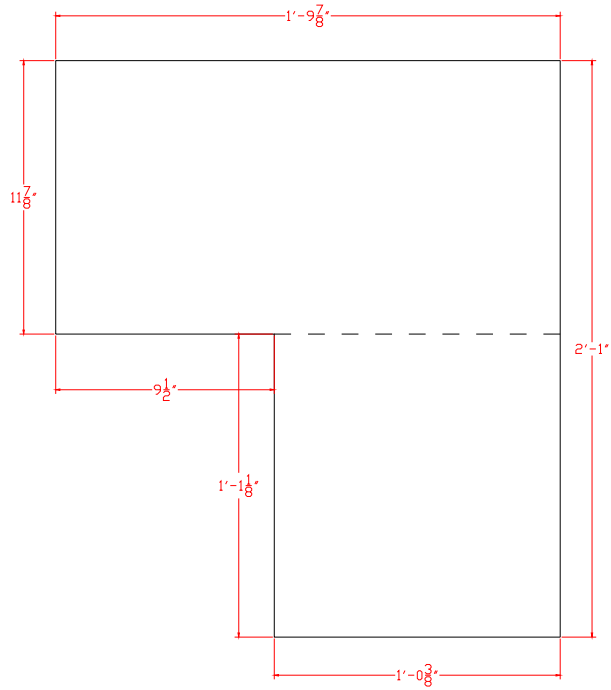


Figure 24 Schematic of Layout Drawing for Sheetmetal

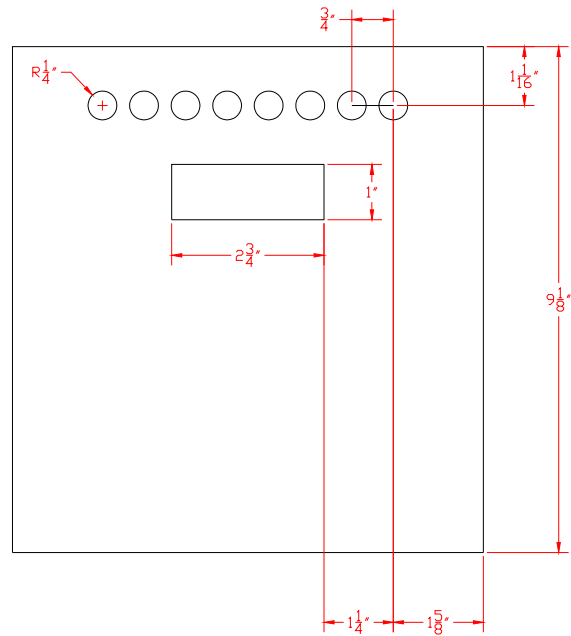


Figure 25 Schematic of Layout for Front Panel

FRONT PLATE ASSEMBLY

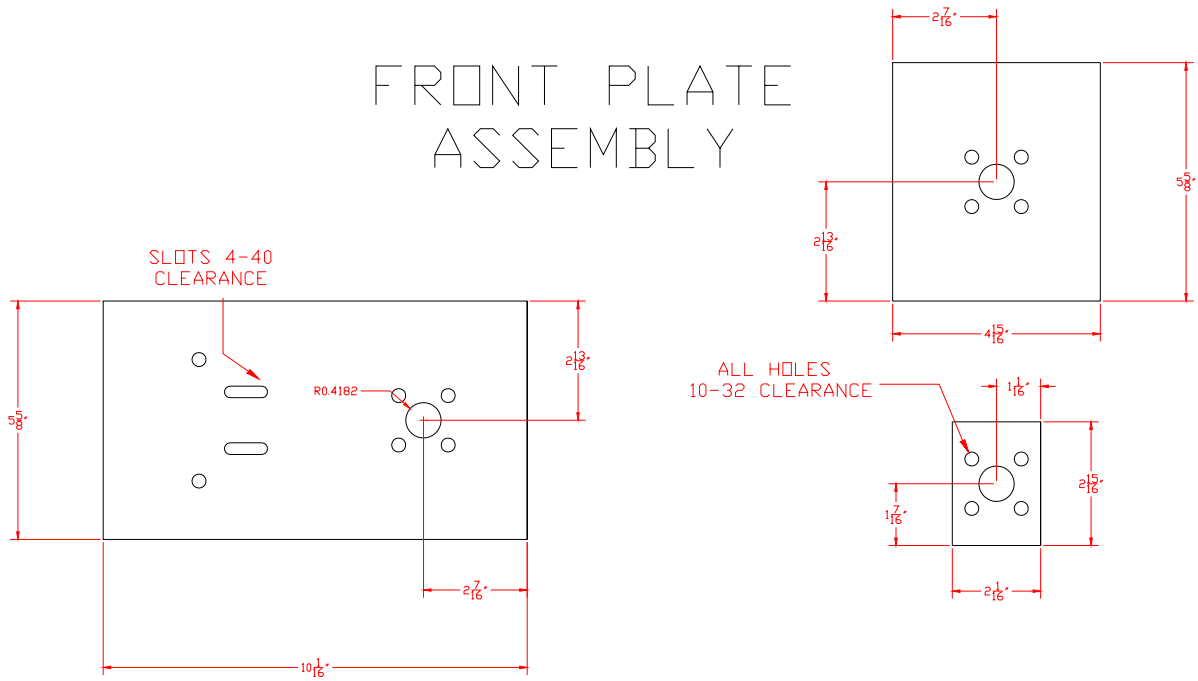


Figure 26 Schematic for Motor/Sensor/Drive Shaft Assembly

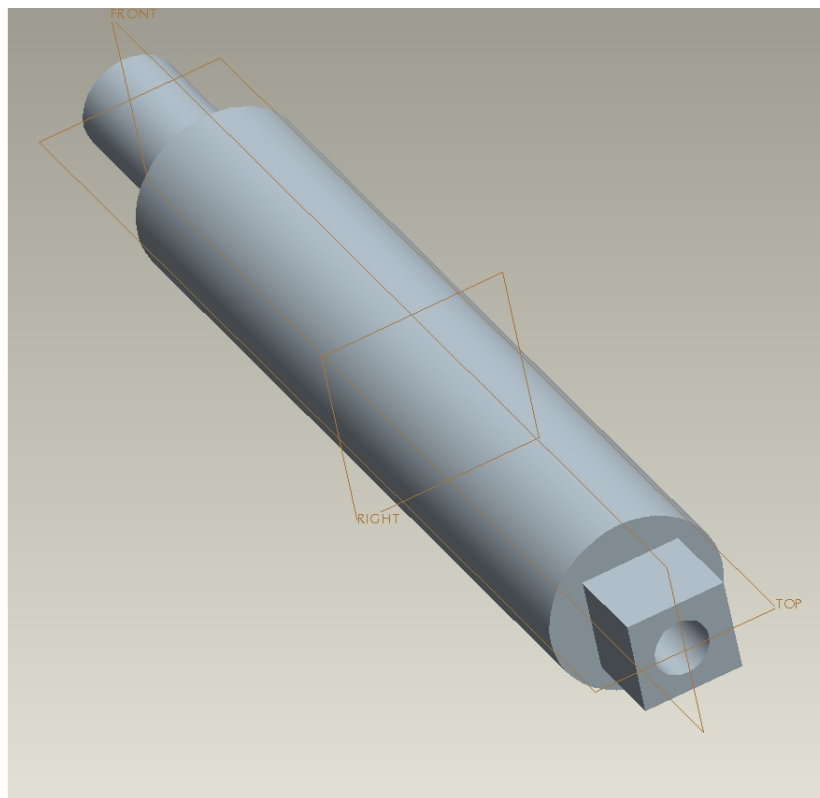


Figure 27 Custom Drive Shaft

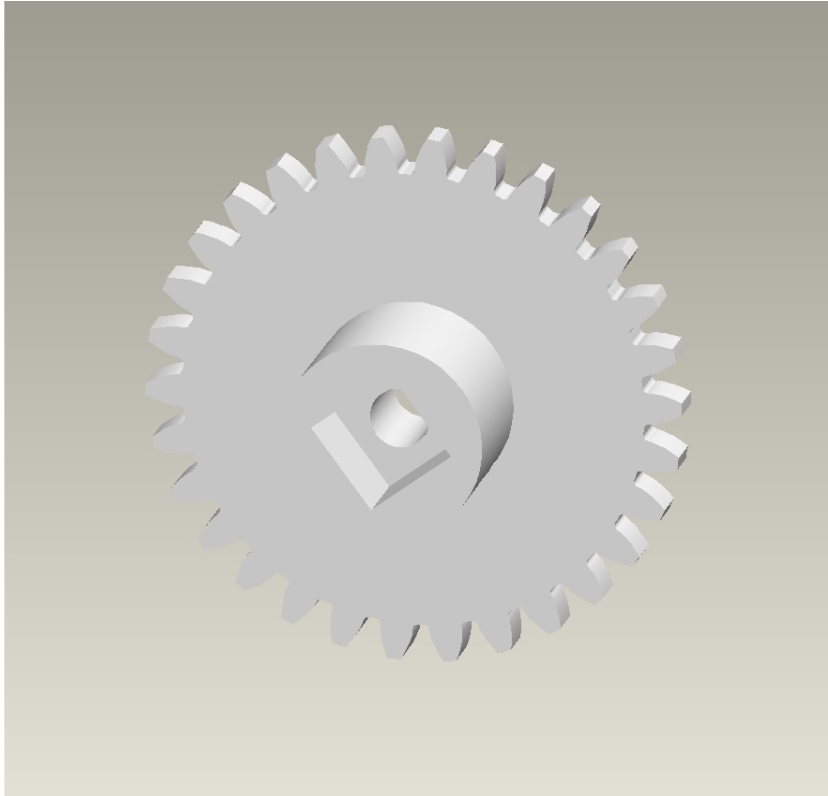


Figure 28 Drive Gear Example