

Final Project Report

ME 307

Section L02

Colorado State University

December 7, 2004

S.T.A.C.I.E.



David Bryant

Justin Hollingsworth

Tyson Rabani

Jonathan Reynolds

Table of Contents

List of Figures.....	2
Introduction	3
Design Summary.....	4
Design Details.....	8
Visual Output Display.....	8
Audio Output Device.....	10
Manual Data Input.....	11
Automatic Sensor Input.....	12
Actuators.....	14
Logic, Counting, and Control.....	16
Detailed Wiring Diagram.....	19
Software Flow Chart – Distance Sensor.....	20
Software Flow Chart – Remote.....	21
Project Cost Report.....	22
Appendix A – 3D Model Images.....	23
Appendix B – Engineering Drawings and Schematics.....	29
Appendix C –Software.....	33

List of Figures

Figure 1.....	3
Figure 2.....	3
Figure 3.....	4
Figure 4.....	4
Figure 5.....	4
Figure 6.....	5
Figure 7.....	6
Figure 8.....	7
Figure 9.....	8
Figure 10.....	9
Figure 11.....	10
Figure 12.....	11
Figure 13.....	12
Figure 14.....	13
Figure 15.....	14
Figure 16.....	15
Figure 17.....	16
Figure 18.....	17
Figure 19.....	17
Figure A1.....	23
Figure A2.....	23
Figure A3.....	24
Figure A4.....	24
Figure A5.....	25
Figure A6.....	25
Figure A7.....	26
Figure A8.....	26
Figure A9.....	27
Figure A10.....	27
Figure A11.....	28
Figure B1.....	29
Figure B2.....	30
Figure B3.....	31
Figure B4.....	32

Introduction

Research:

Every year, millions of dollars are spent on robotic research. Many interesting new designs have been made, with many different applications. Pictured below are two examples of new robotic designs that aren't very cost effective.



Figure 1 – Sony Orio



Figure 2 – Honda Asimo

There exists a need to develop a robot that demonstrates a great deal of functionality at a low cost. It is important that robotic innovation continue, and that creative people persist in creating useful designs. The intent is to develop a robot to serve as a stepping stone that will lead to future improvements in the robotic field.

Business:

The proposed robot device will be able to perform several useful tasks. One common problem in the business environment, is trying to find a conference room that has the right equipment for a presentation. In this day and age, nearly all presentations are done with PowerPoint, and it can be frustrating trying to find and reserve a conference room that has both a computer and a projector available. However, such things will no longer be a concern for the businessman that owns one of these robots. The proposed robot device will have a built-in projector that can display a PowerPoint presentation anywhere, as long as there is a white wall to point it at. In addition, the business associate will not have to carry a lot of bulky equipment, but will be able to drive it to the conference room with ease, through the use of our wireless remote control. Colleagues will be impressed by the level of professionalism that one is able to demonstrate during a presentation through the use of our robot.

Design Summary

The project is a robotic office assistant, which can be used to assist with duties around the office, such as multimedia presentations. The robot balances on two wheels, and can be driven around for mobility. The user can control where it goes through the use of a wireless remote. Once it has been steered to the appropriate position, the user can then deploy the third leg of the robot, allowing it to rest in a stable position rather than rocking back and forth while projecting. By pushing another button, the user can begin projecting a presentation onto a wall or screen, and play sound clips through the robot's built in speakers. The robot also has the capability to play pre-programmed messages through the use of an onboard voice synthesizing utility.

The robot is a Self-contained, Transliterating, Automated, Controllable, Integrated, Embarkment, also known as "S.T.A.C.I.E." It consists of a frame composed of Plexiglas® shelves held together with AllThread®. The left and right wheels each have a motor that is controlled by a PIC microcontroller. On the top shelf of frame, sits the projector, which is basically a color LCD screen that is projected through the proper alignment of mirrors and lenses. Mounted onto the sides of the projector apparatus, are the speakers for audio output. On the middle shelf of the frame is where all of our circuit elements and power supplies are located. All of these items (the motors, speakers, projector, voice synthesizer, etc.) are controlled by a series of PIC microcontrollers from Microchip Technology Inc™.

One additional design feature of S.T.A.C.I.E. is the ability to balance on two wheels, similar to a Segway® Human Transporter (see Figures 3 and 4). The Segway balances by utilizing a complicated combination of gyroscopes, accelerometers, and tilt sensors. Group 24 found gyroscopes and accelerometers to be fairly expensive, and did not consider this to be a cost effective way of accomplishing this feature. Instead, the group endeavored to prove that a self balancing device of the same style could be built by using sensors that are significantly less expensive, similar to what was done with the nBot (See Figure 5).



Figure 3 – Segway®



Figure 4 – Person riding a Segway®



Figure 5 – Balancing nBot



Figure 6 – Front/angled view of Robot



Figure 7 – Front View of Robot

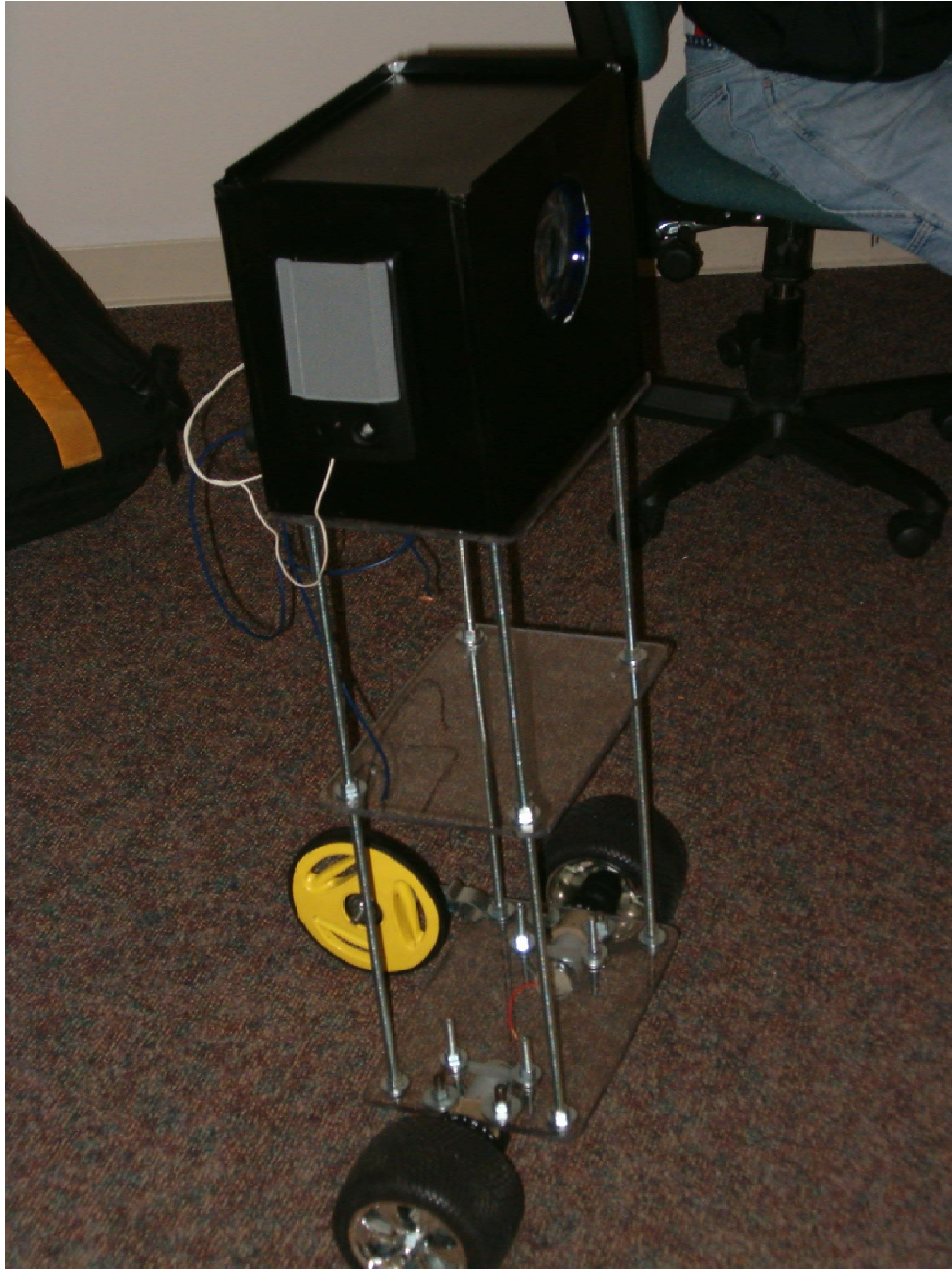


Figure 8 – Front/angled view of robot



Figure 9 - Projector

Design Details

The project demonstrates the following six functional element categories:

1. Visual Output Display
2. Audio Output Device
3. Manual Data Input
4. Automatic Sensor Input
5. Actuators
6. Logic, Counting, and Control

The following sections will describe, in detail, how each of these categories were accomplished.

Visual Output Display

The robot has a projector that is capable of projecting images from a video input. Rather than purchasing a projector that could be simply mounted onto the top of the robot, the group

chose to undergo significant amounts of research and planning in order to actually *build* a projector with a few simple components, most of which the group already had in their possession. This approach had obvious economic advantages as well, since a projector with this type of capability would usually cost several hundred dollars. S.T.A.C.I.E.'s projector cost about thirty dollars.



Figure 10 – Projector w/ speaker

The projector (see Figure 10) was built, using a color LCD screen, a mirror, and a lens, all housed in a sheet metal box. A small, color LCD screen receives video input from a basic video camera, and displays the images accordingly. The LCD screen (see Figures A9 – A11) lies in the bottom of the box, and a mirror that is suspended a few inches above the screen, at an angle, reflects the image towards the opening in the box. A lens is mounted in the opening in order to project the image onto a larger area. Two light bulbs, that are intended to be used for automobile headlights, were also mounted inside the projector to provide an image that was bright enough to still be visible when projected over such a large area.

S.T.A.C.I.E. is also equipped with a smaller LCD screen that displays pre-programmed messages. In addition, significant effort was employed, in order to produce a variety of complex animations that go beyond the usual display of simple words. These animations required a large amount of tedious programming.

Audio Output Device

The robot has speakers mounted onto the sides of the projector, which can play whatever is received from its audio input cable.

S.T.A.C.I.E. has the capability to play electronically voice synthesized audio, through the use of the Speakjet® Development Board (see Figure 11). The Speakjet® is a complex IC that uses mathematical sound algorithms to exploit a five channel sound synthesizer to ultimately imitate a person's voice, and a variety of sounds.

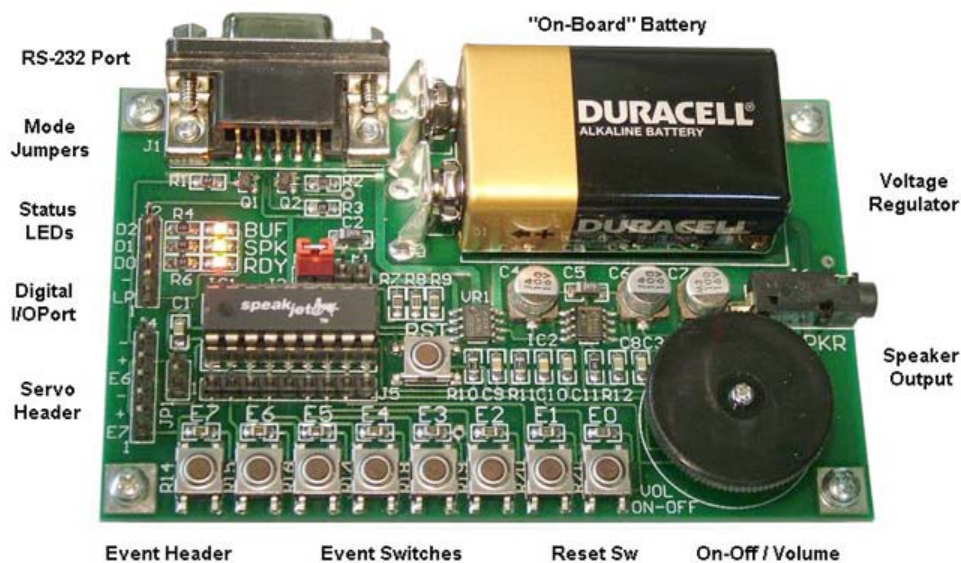


Figure 11 – Speakjet® Development Board

As a result of the above methods, the Speakjet® is capable of producing seventy-two allophones (speech elements), forty-three sound effects, and twelve DTMF tones. After a great deal of familiarization with the use of this IC, in addition to a great deal of trial and error, the group was able to give S.T.A.C.I.E. a voice and make her “talk.” After a large amount of independent research, the group learned how to program the chip to play the correct allophones in

the correct sequence, with the appropriate pauses, pitch, etc. to make it seem as if the robot was speaking. Plenty of effort was required to become familiar enough with the chip to be able to use it to its full capability. The Speakjet® is a powerful device, and does indeed have the potential to accurately synthesize a voice and a wide range of other sounds.

Manual Data Input

Manual data input for the robot is accomplished with a wireless RF remote. The remote has buttons for each of the following functions:

- Moving Forward: The robot will move in the forward direction.
- Moving Backward: The robot will move in the reverse direction.
- Turning Left: The robot will rotate to the left.
- Turning Right: The robot will rotate to the right.
- Projector Power: The projector will turn on/off.
- Projector Slide Change: The next slide will be displayed by the projector.
- Tripod Extension: The third wheel of the robot will be extended to make the robot more stable while projecting.



Figure 12 - Laipac Technologies™ RF Transmitter (left) and Receiver (right)

When one of these buttons is pressed, it causes the input pin of the 16F876 PIC in the remote to be high. The PIC then outputs, through a different pin, which button is pressed to the wireless transmitter via a serial signal. The serial signal is then transmitted to the robot, where the signal is obtained by an onboard receiver. The receiver inputs this signal to an 18F452 PIC in the robot. After the PIC has this input, telling it which button was pressed, the PIC carries out

whatever action is implied by the respective button being pressed, according to the software programmed into the PIC.

Automatic Sensor Input

The robot has two functions that utilize automatic sensor inputs: the RF receiver that receives the serial signal from the remote, and the IR range sensor that is used for balancing.

S.T.A.C.I.E. is equipped with the capability to balance on two of its wheels, in a fashion similar to that of the Segway® Human Transporter. The Segway® uses a combination of five gyroscopes and several accelerometers. Group 24 decided to design a robot that could balance on two wheels, like the Segway, but would not require such expensive components.



Figure 13 – Sharp GP2D120 Sensor

A GP2D120 infrared distance sensor (see Figure 13) is employed to accomplish balancing. The sensor is aimed at the ground, and continuously measures the distance from the robot to the ground. If the distance is getting shorter, it implies that it is falling forward, and the robot's motors are subsequently activated to make the proper correction. The reverse, of course, happens if the distance is getting larger. It is interpreted as an indication that the robot is falling backwards, and the motors are instructed by the PIC to compensate by moving the robot backwards, "underneath" it to catch the falling robot.

The GP2D120 sensor is an analog sensor, and as a result A/D conversion was necessary in order for it to be used by the PIC microcontroller. Interpreting the output of the sensor as a

distance required a significant amount of independent research. The distance measured by the sensor is continuously reported as a voltage, which is converted by the 18F452 PIC to a digital number. Points had to be plotted in order to find the relationship between the reported voltage and the distance observed by the sensor. By referencing the data sheet for this sensor (see Figure 14) and developing equations for the best fit curve, the group was eventually able to use the sensor to report actual distances.

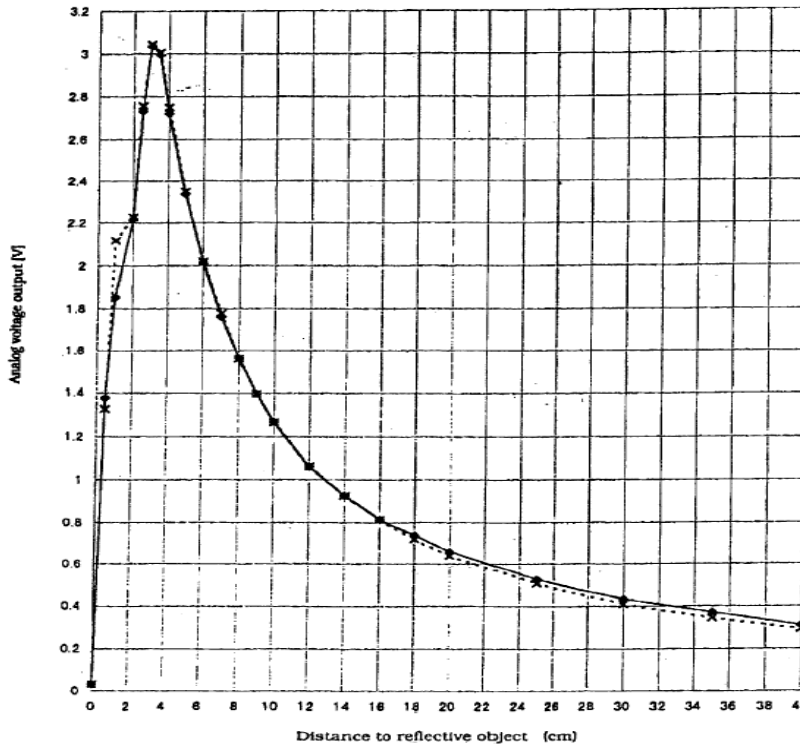


Figure 14 – Example Graph from GP2D120 Sensor Data Sheet

The other sensor used by S.T.A.C.I.E. is our wireless receiver. It is used to detect when a RF signal is being transmitted from the remote. The group had to make sure that the receiver was interfacing properly with both the PIC and the transmitter. In addition it had to be guaranteed that the detected signal was indeed an accurate indication of which button was being pressed on the remote. From the wireless signal, it is able to interpret what it receives into a digital signal that it outputs to a PIC 18F452 in the robot. This allows the robot to know which button was pushed, and allows it to proceed by carrying out the corresponding function.

Actuators

S.T.A.C.I.E.'s actuators are the two motors attached to the left and right wheels. These motors allow the robot to be mobile, which adds greatly to the convenience inherent in its functionality.

Rather than browse through endless catalogues trying to find the perfect motors for S.T.A.C.I.E., the group chose to instead buy two inexpensive 18V cordless drills. The group then took these drills apart, removed their motors and any useful corresponding components, and used them for the robot (See Figures 15 – 16, A1 – A8).

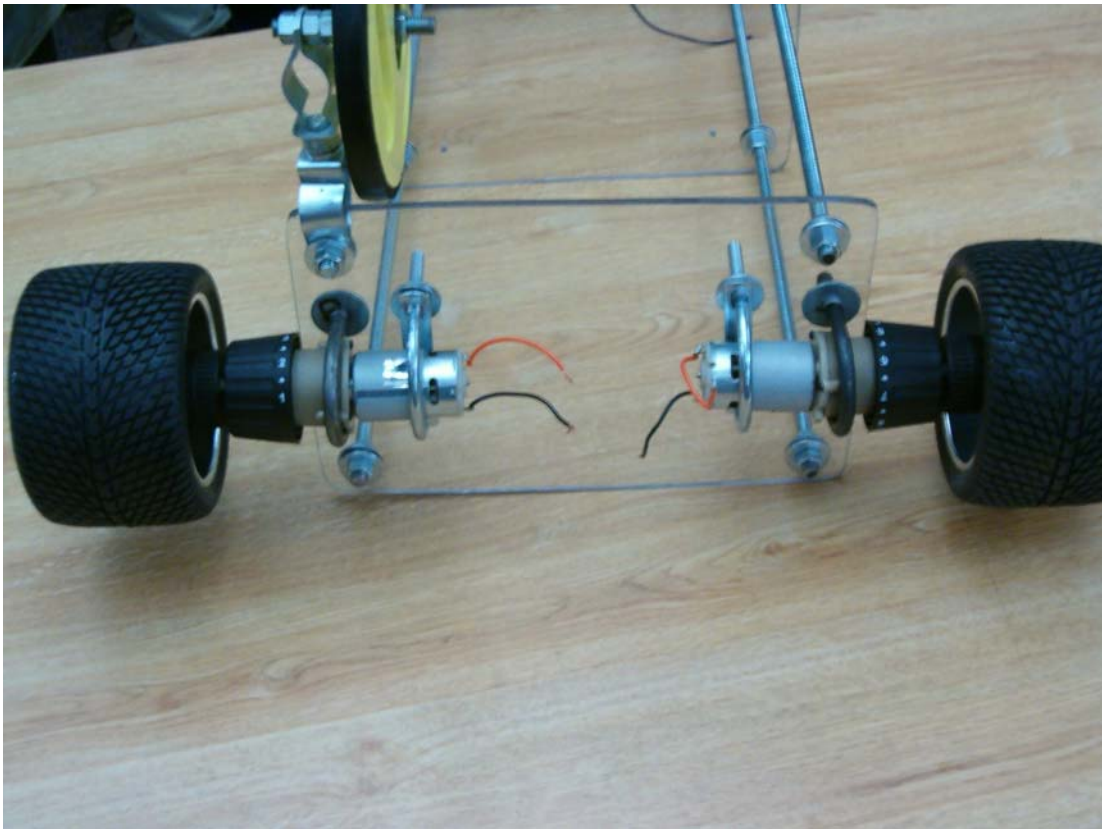


Figure 11 – Wheels and motors



Figure 12 – Close-up of motor

There are several reasons why this was convenient for this project. Firstly, purchasing twenty dollar cordless drills was just as cheap as buying the motors separately. This is especially true when one considers that the drills come with eighteen volt batteries, which can be conveniently used for the power source of the robot. The drills are also already designed to run forwards and backwards. In addition, the drills have bearings that are designed to support the relatively large amounts of axial stress that a drill undergoes when drilling a hole. This feature made it unnecessary to develop a gear box for the motors, which allowed us to simply “chuck” them directly to the axel of the wheels (See Figures 16, A5). This saved a great deal of time and effort, and was yet another convenient reason to use drill motors.

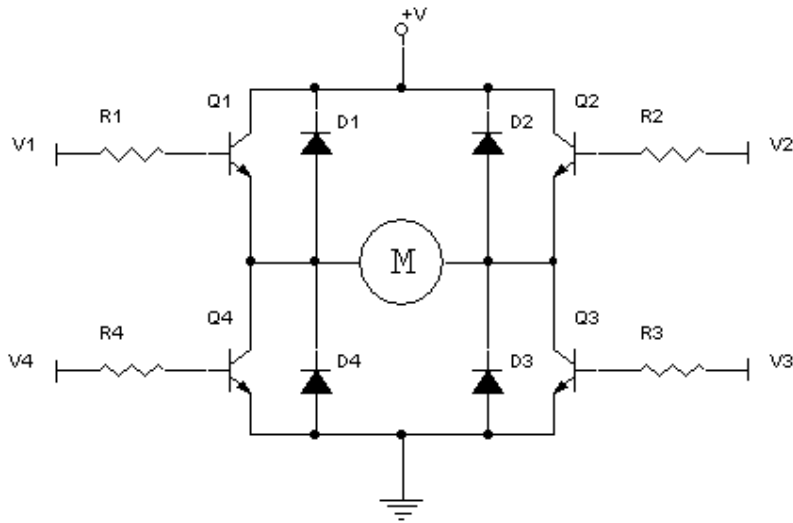


Figure 13 - H-Bridge Schematic

If S.T.A.C.I.E. was to balance, she would need to be able to move both forward and in reverse. This functionality implied that an H-bridge was necessary, to allow for the polarities of the motor to be reversed, resulting in the motor being able to move both forward and in reverse. Rather than purchasing this component from a vendor, the group chose to take the extra time to simply construct one, as this would provide extra knowledge and experience in addition to saving the money that would have been spent.

Logic, Counting, and Control

There are many aspects of S.T.A.C.I.E. that demonstrate the use of logic, counting and control. Almost every aspect of the code written for this robot uses some form of logic, counting, or control, and would fall under this category. When using PIC microcontrollers, it is all but guaranteed that one will have to use logic when writing the code that allows the PIC to function.

One of the most prominent examples is the project's balancing function. The PIC 18F452, based on inputs from the Sharp GP2D120 distance sensor, determines if the distance to the floor is getting smaller or larger (i.e. if the robot is falling forward or back). Based on that information, the PIC microcontroller is programmed to mitigate the robot's falling by moving its motors forward or backwards. This can be a fairly complex task to program, as it is difficult to quantify how fast the robot is falling in either direction, and requires a fair knowledge of calculus to

determine the rate of change of the input from the distance sensor. Furthermore, there are also vast difficulties associated with the process of programming the robot to recover from a falling position. It is too easy to over or under-compensate, making the job of balancing S.T.A.C.I.E. a challenging task indeed.

Another example of something that demonstrates functionality in this category is the input device. The PIC on the remote has to first determine if a button has been depressed. Once it has done that, it must output which button was pressed to the wireless transmitter, which will send the signal to the receiver on the robot. That signal is then input into another PIC. Based on the input, the PIC is programmed to carry out a certain function (e.g. move forward by turning on the motors, turn by turning on one motor, power on the projector, etc.)

The programming of the PIC for the projector is another example of the use of logic in this project. In order to be as economical as possible, the decision was made to not buy a projector, but to rather make one using a collection of components that the group already had. A PowerPoint presentation can be recorded on a video camera, which can be connected to the video inputs on S.T.A.C.I.E.'s color LCD projector. The camera will normally be paused, to allow a slide to be displayed for an indefinite period of time. When the user desires to view the next slide, they will press the appropriate button. Once this signal is transmitted and received by the robot, the PIC is programmed to make the video camera play for a few seconds, just long enough to allow the next slide to be shown, and then pause again. It is important to note that this is clearly not the most practical way to display a presentation if S.T.A.C.I.E. were going to be mass produced. However, for the purposes of this project, this was an inexpensive way to accomplish the goal, in addition to being a good way to demonstrate functionality in this category.



Figure 18 – 18F452 PIC Microcontroller

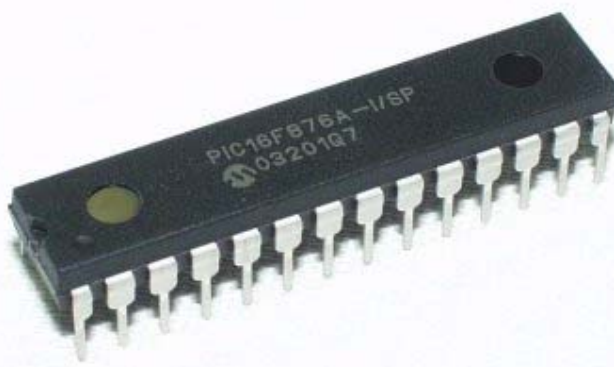
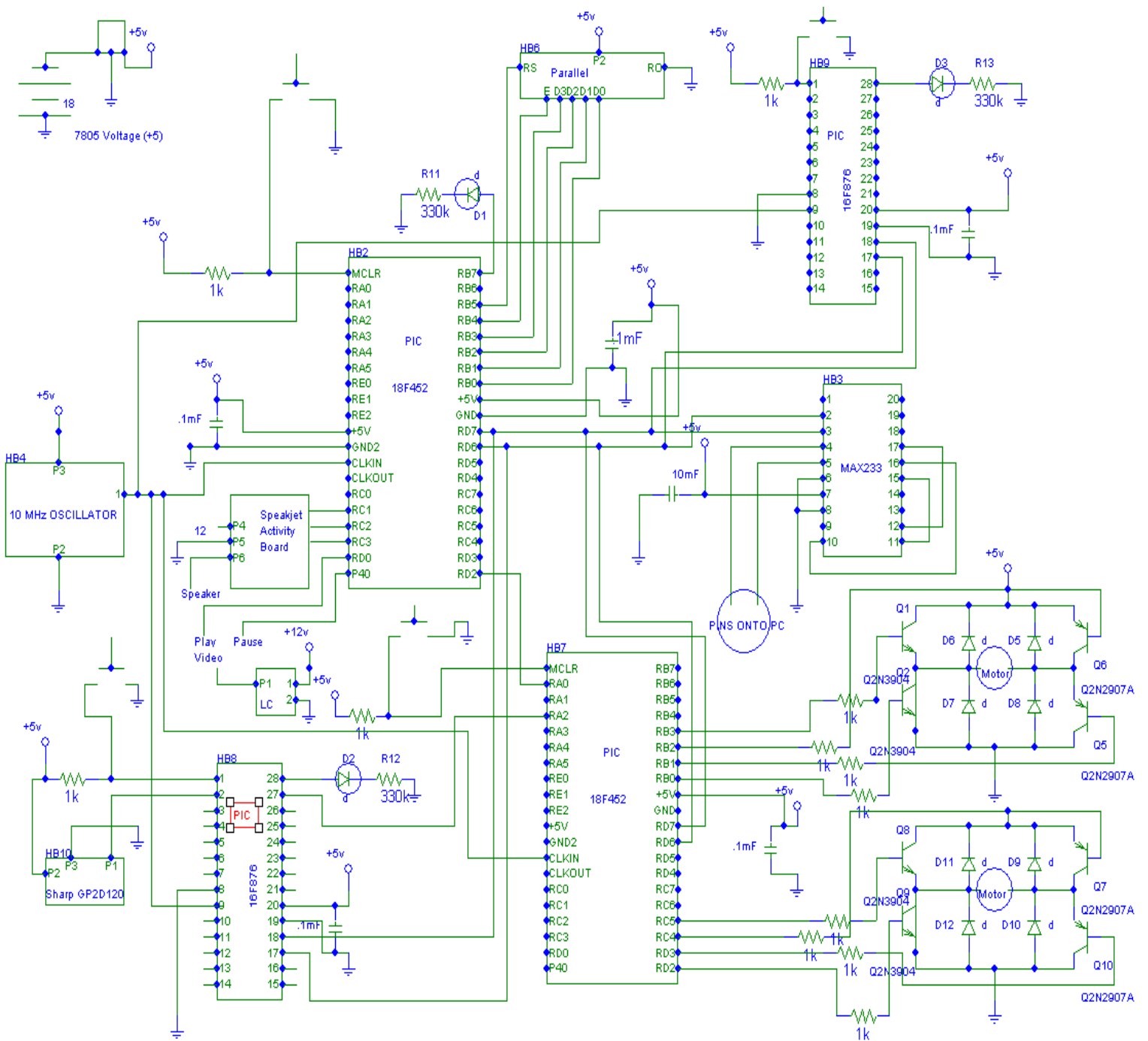


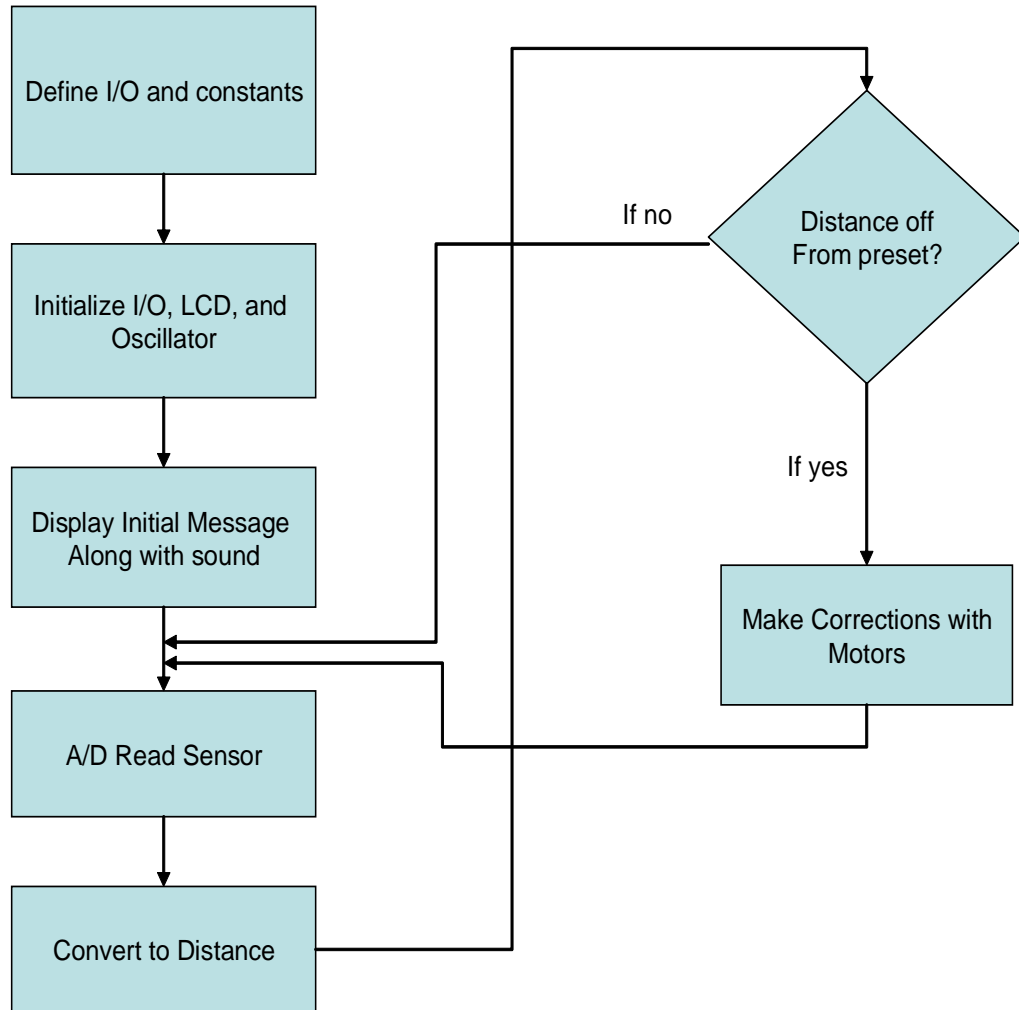
Figure 19 – 16F876 PIC Microcontroller

This project also utilizes several PICs, some of which are the 18F452 model while others are the 16F876. One of them operates using a 4 MHz crystal, while others use a 10 MHz oscillator. The task of integrating all of these different PICs with different frequencies for serial communication is another example that demonstrates functionality in this category.

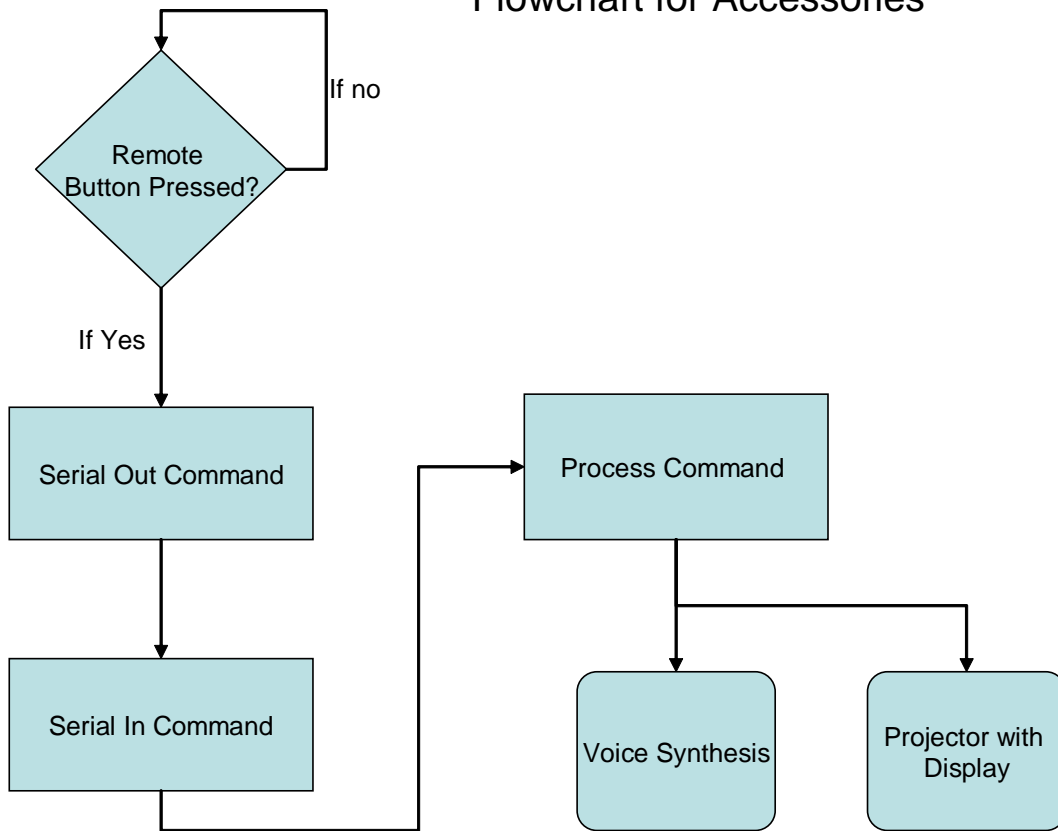
Detailed Wiring Diagram



Software Flowchart for Distance Sensor



Flowchart for Accessories



Project Cost Report:

Item	Quantity	Cost / Item	Total Cost
PIC 18F452	4	donated	\$0.00
PIC 16F876	2	donated	\$0.00
Max233 chip	1	donated	\$0.00
Drillmaster Drill	2	\$20.00	\$40.00
10 Mhz Oscillator	1	donated	\$0.00
4 Mhz Crystal	1	donated	\$0.00
Transistors	8	donated	\$0.00
Voltage regulators	3	donated	\$0.00
Speakjet® Development Board	1	\$50.00	\$50.00
LCD Screen	1	\$12.00	\$12.00
Color LCD Screen	1	\$20.00	\$20.00
Video Camera	1	donated	\$0.00
LED	7	donated	\$0.00
Speakers	2	donated	\$0.00
Road Rage 40 Series Tires (pair)	1	\$29.00	\$29.00
Velocity 6 40 Series Weld Wheels (pair)	1	\$22.00	\$22.00
Wireless Transmitter & Receiver	1	\$10.00	\$10.00
Pair of Headlights	1	\$11.00	\$11.00
Magnifying Glass	1	\$5.00	\$5.00
Sheet Metal Box Frame	1	donated	\$0.00
Plexiglas®	1	donated	\$0.00
Allthread/U-bolts	1	\$8.00	\$8.00
Sharp Sensor GP2D120	3	donated	\$0.00
Training Wheel	1	donated	\$0.00
Misc. Nuts, Bolts, and Wires	1	\$5.00	\$5.00
Serial Cable	1	\$5.00	\$5.00
Button Switches	10	donated	\$0.00
Breadboards	2	donated	\$0.00
Mirror	1	donated	\$0.00
Total Shipping Costs =			\$30.00
Grand Total =			\$247.00
Casey Industrial Sponsorship =			-\$300.00
Total Cost =			-\$53.00

Appendix A – 3D Model Images

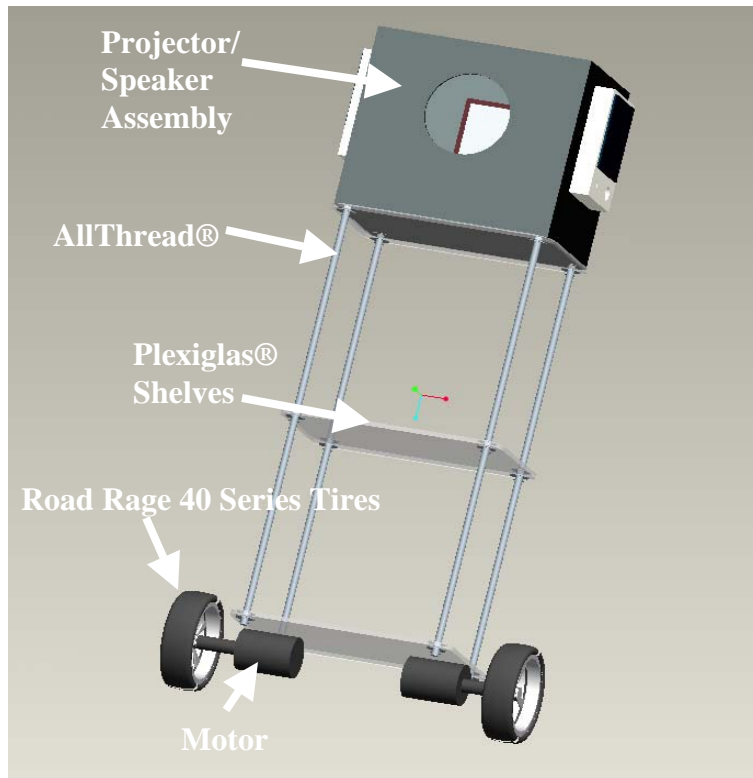


Figure A1 - Full front view of robot frame and projector

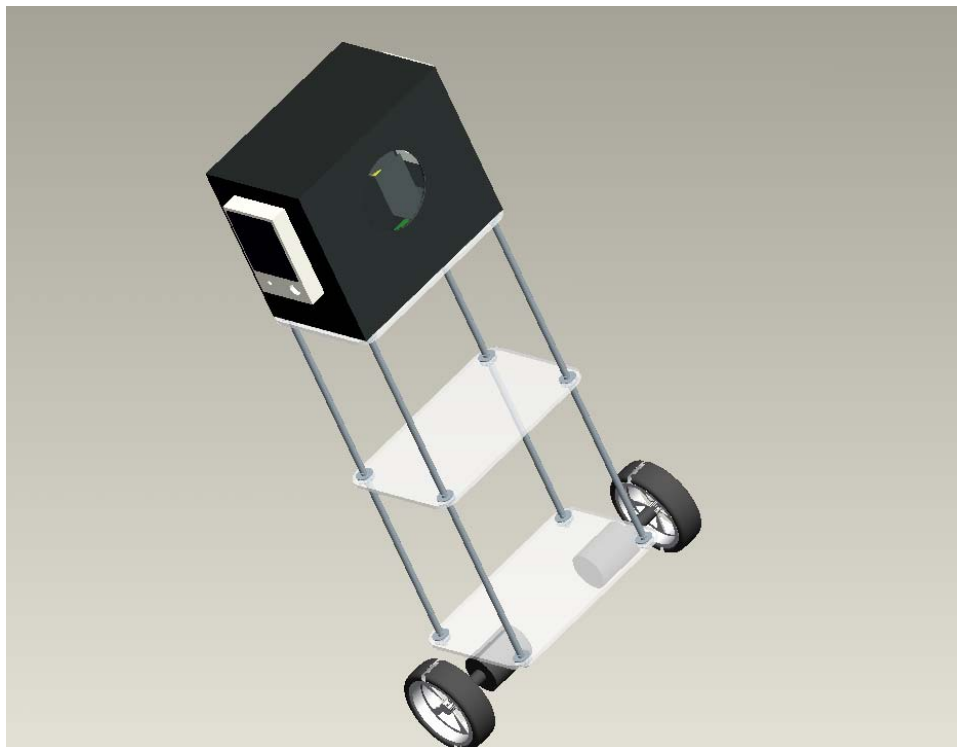


Figure A2 –Full frontal view of robot frame and projector (alternate angle)

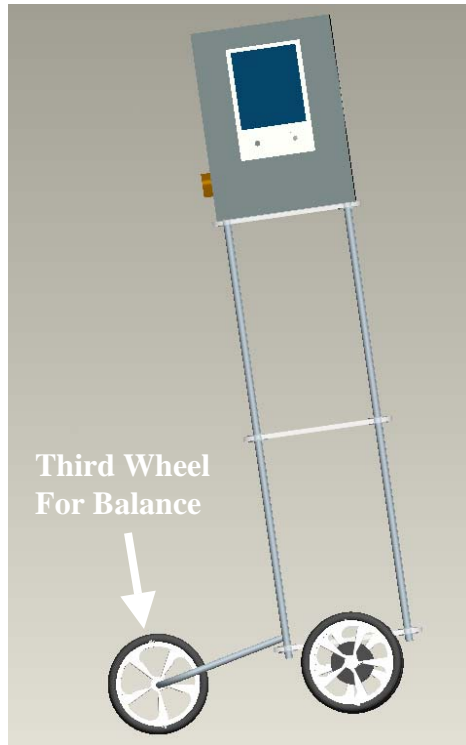


Figure A3 – Side View of Robot with Third Balancing Wheel



Figure A4 – Rear View of Robot with Third Wheel

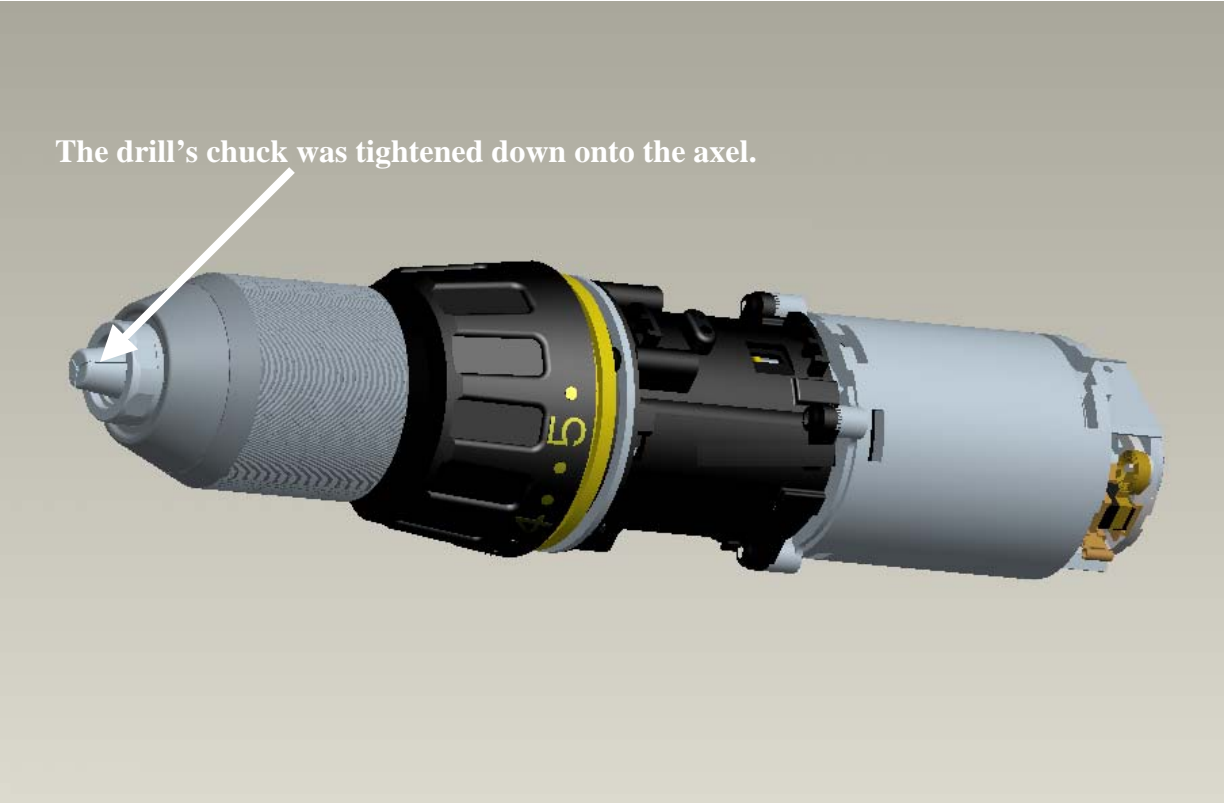


Figure A5 – Drill motor used to drive wheels

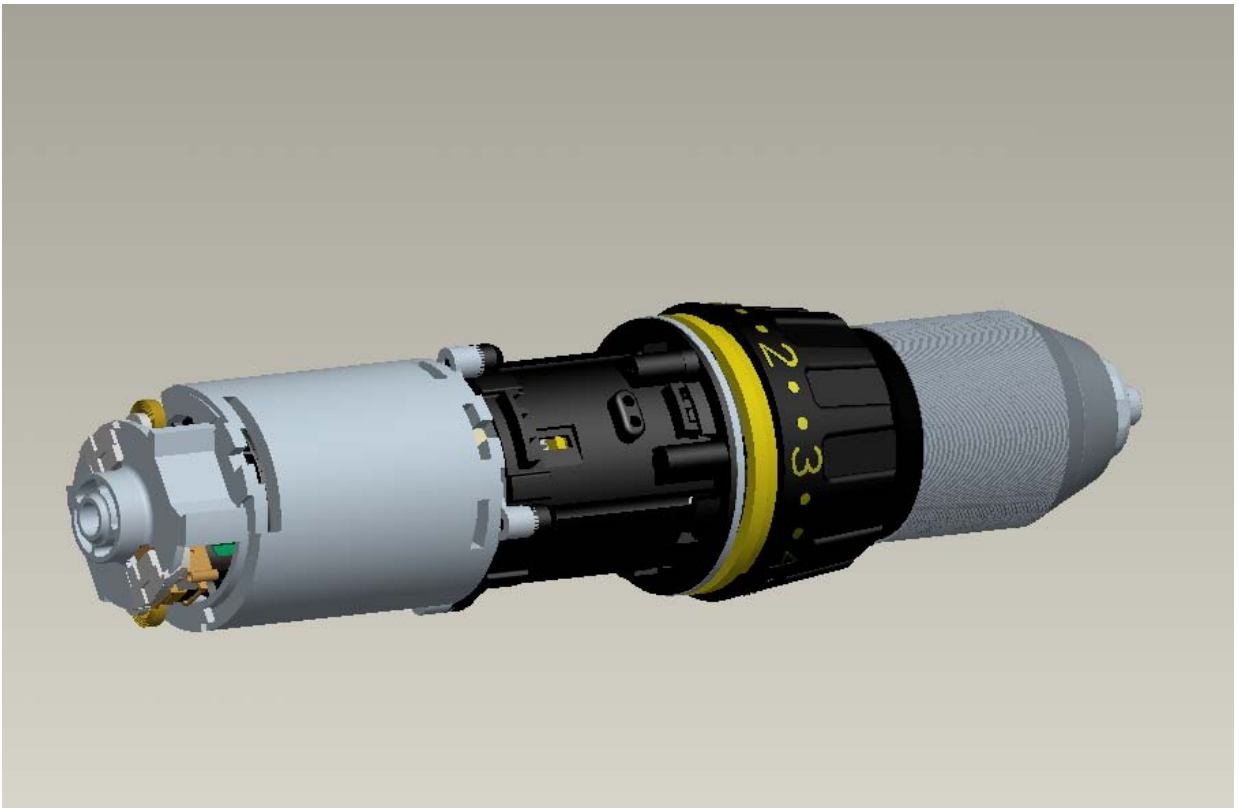


Figure A6 – Second view of drill motor



Figure A7 – Exploded view of drill motor

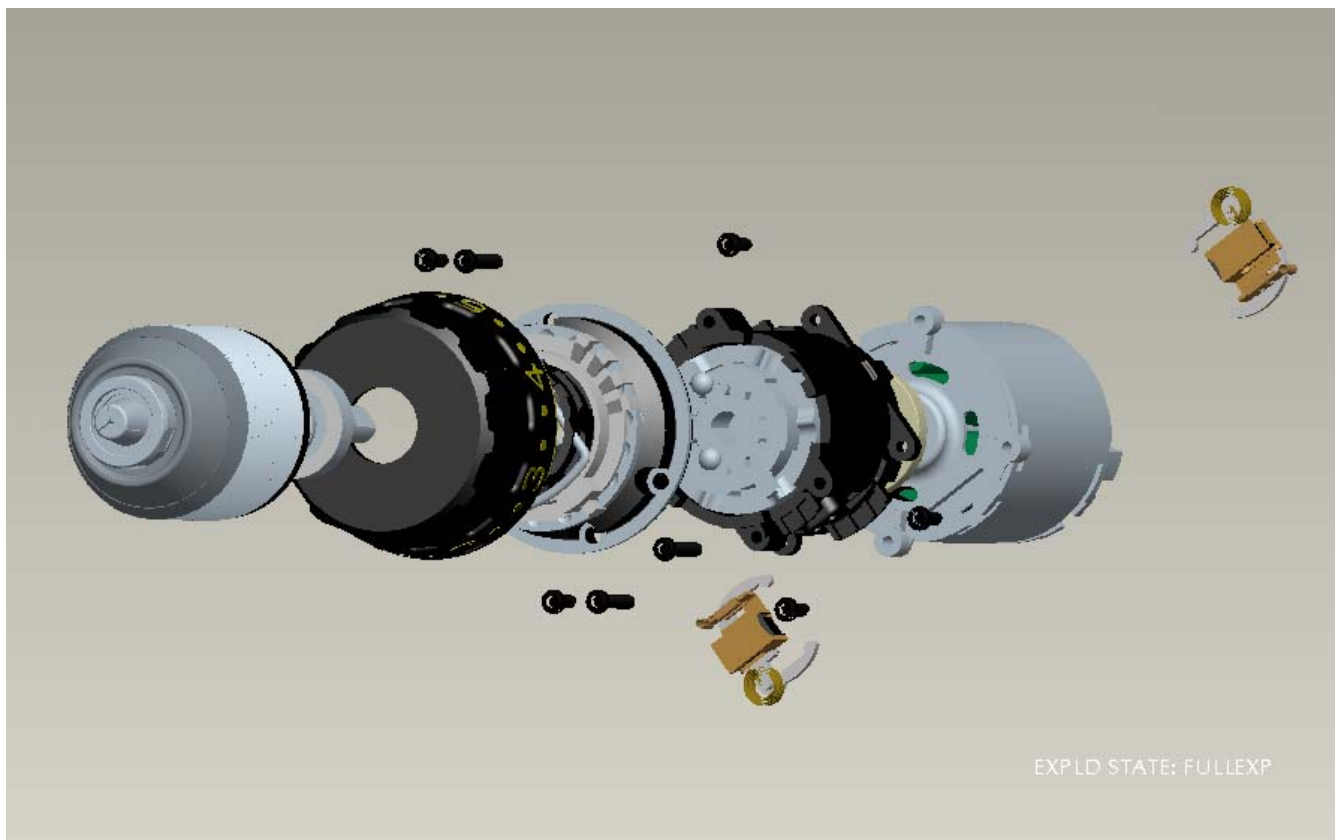


Figure A8 – Second exploded view of drill motor

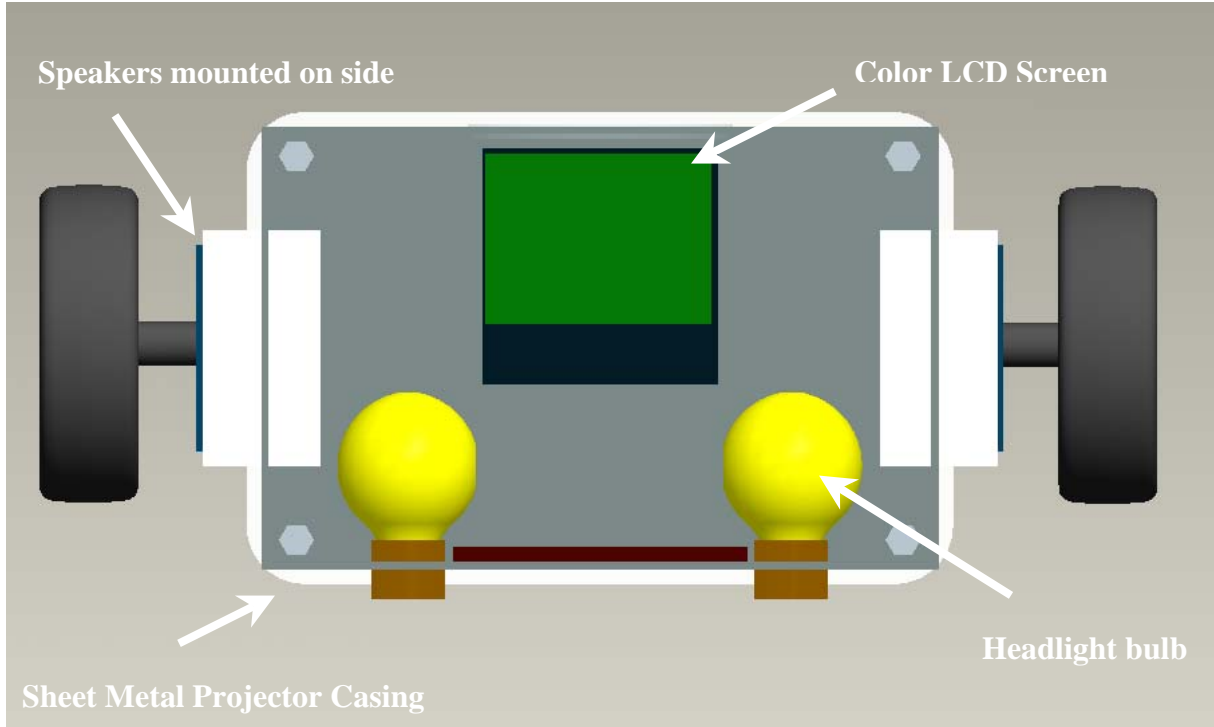


Figure A9 - Top View of Projector



Figure A10 - Speaker mounted on the side of projector

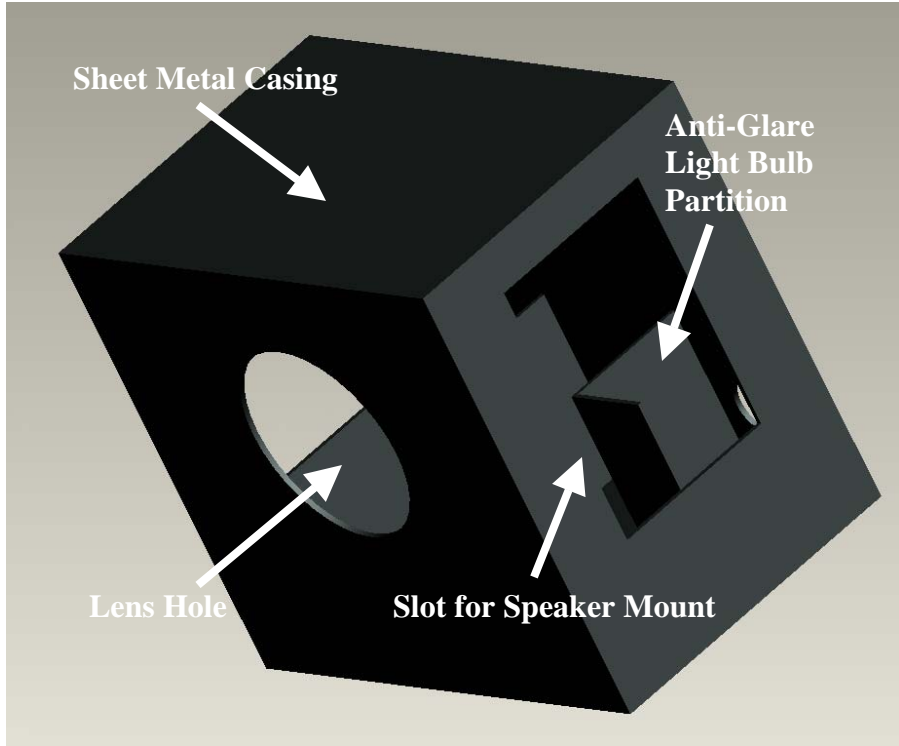


Figure A11 - Projector Box

Appendix B – Engineering Drawings and Specifications

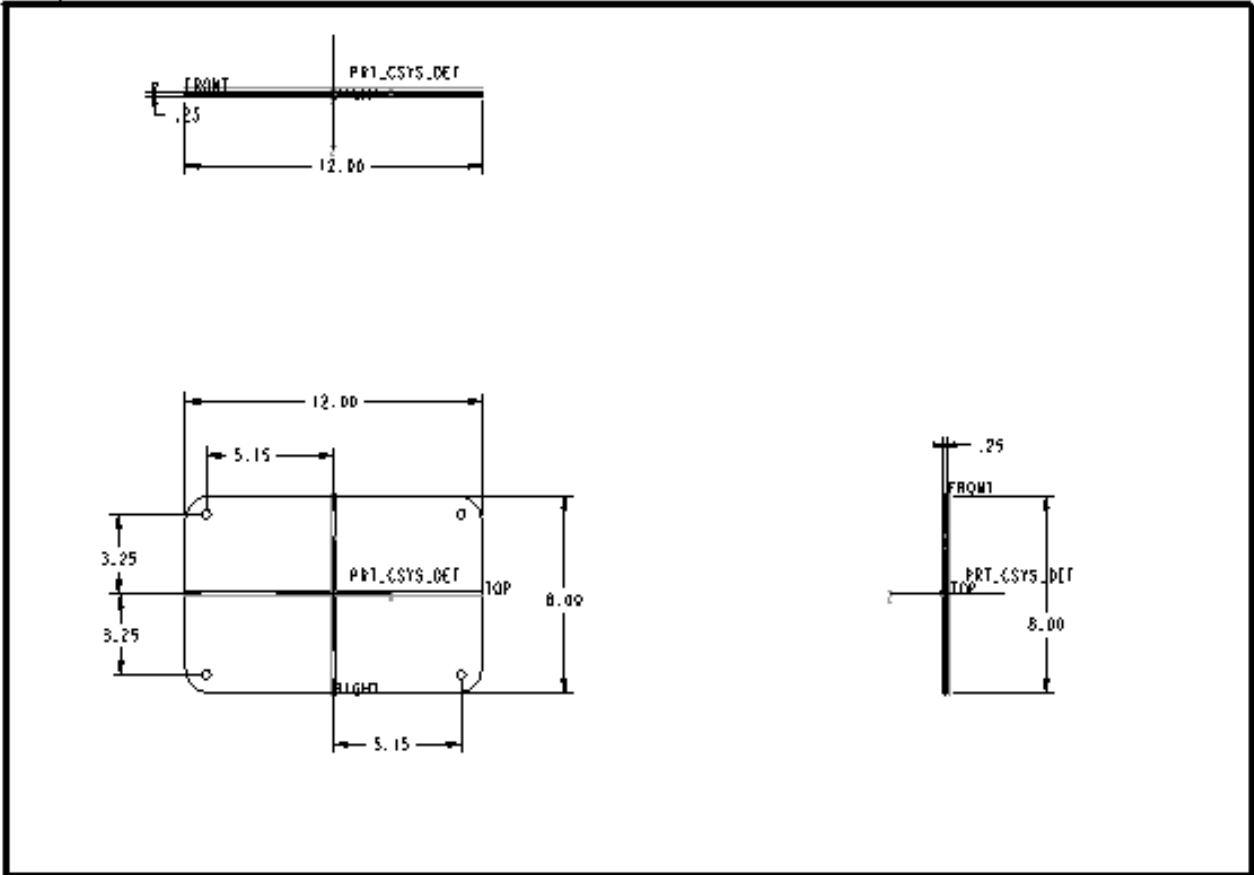


Figure B1 – Plexiglas® Shelf

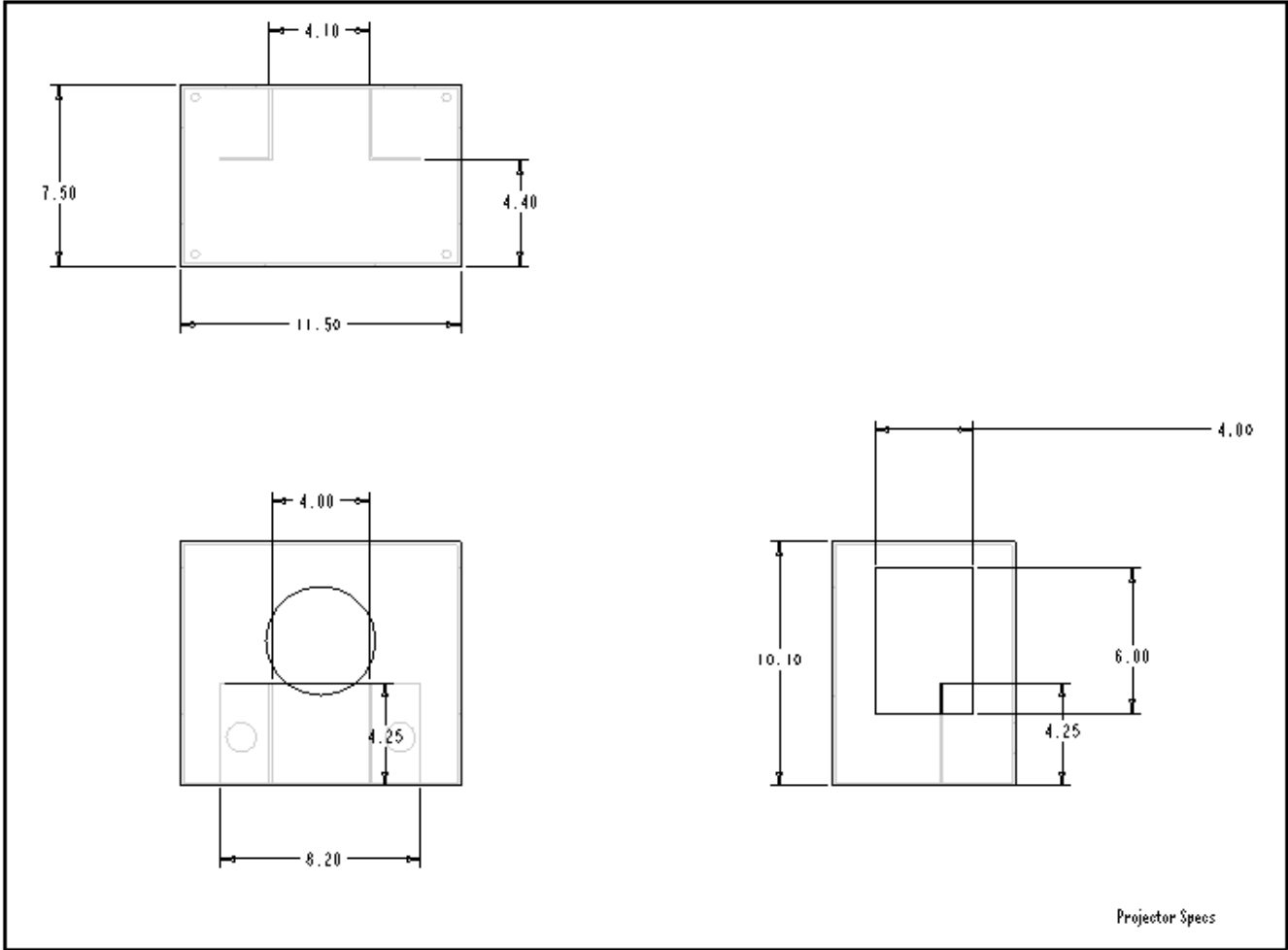


Figure B2 – Projector Box

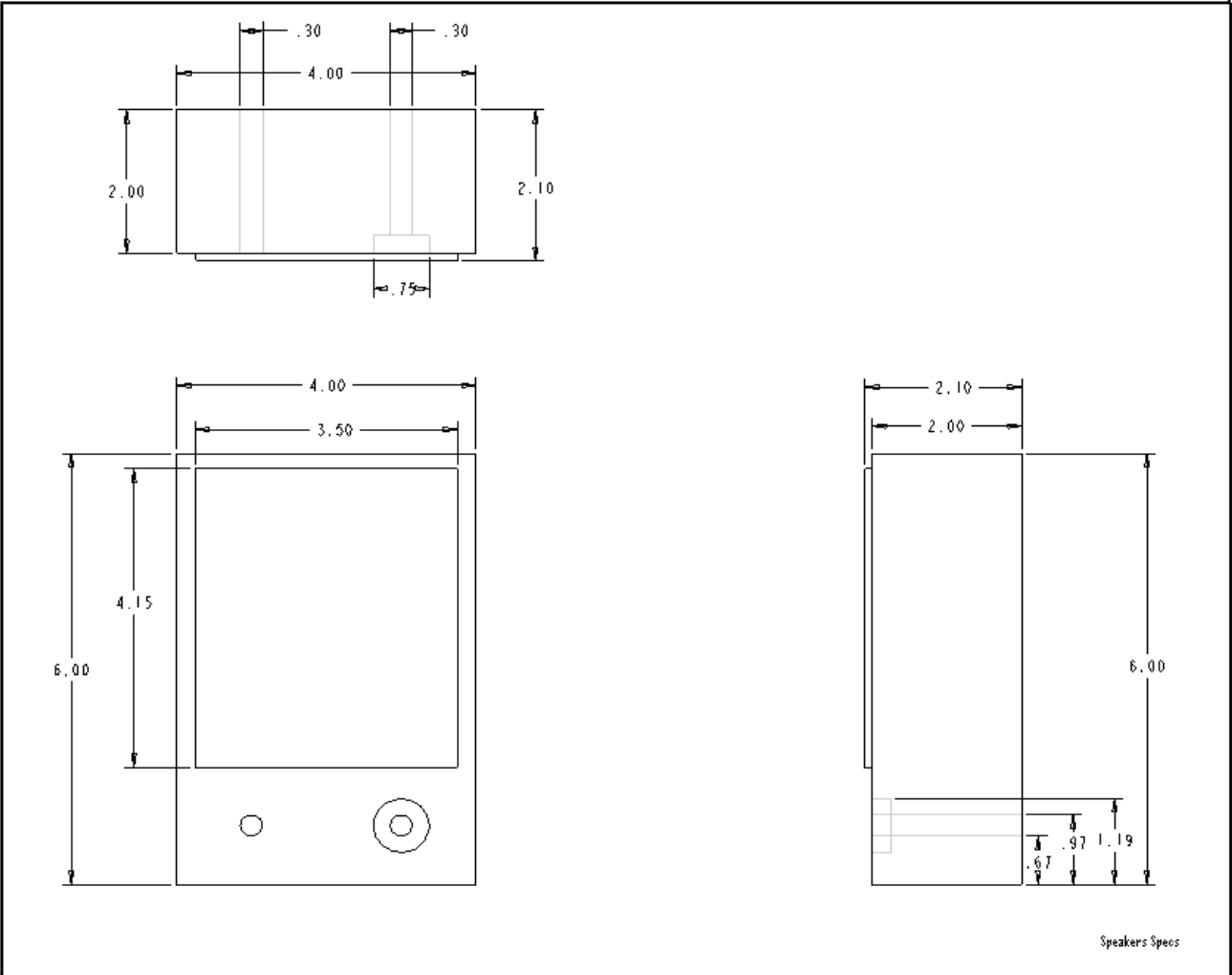


Figure B3 – Speaker Specifications

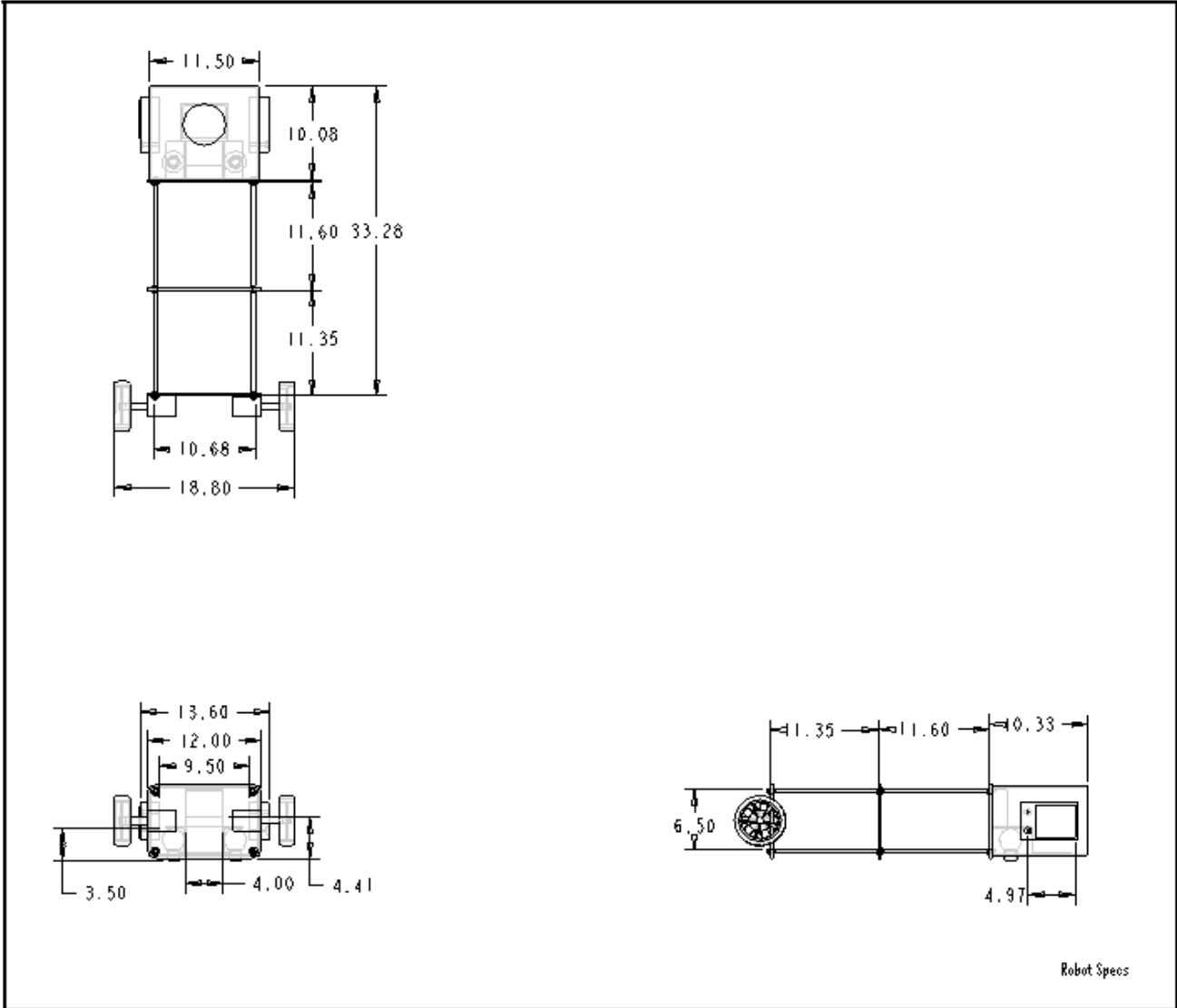


Figure B4 – Total Robot Assembly Specifications

Appendix C – Software

```
*****
'*  Name      : MAIN.BAS                                     *
'*  Author    : Jonathan Reynolds                           *
'*  Notice    : Copyright (c) 2004                          *
'*            : All Rights Reserved                         *
'*  Date      : 12/7/2004                                    *
'*  Version   : 1.0                                         *
'*  Notes     :                                             *
'*            :                                             *
*****

DEFINE OSC 10
DEFINE LCD_DREG  PORTB           'Set Data port of LCD
DEFINE LCD_DBIT  0               'Allocate B.0 to B.3 as pins
DEFINE LCD_RSREG PORTB
DEFINE LCD_RSBIT 5
DEFINE LCD_EREG  PORTB
DEFINE LCD_EBIT  4               'Set E bit

INCLUDE "MODEDEFS.BAS"
TRISD = %11111111

I VAR WORD
J VAR WORD
A0 VAR WORD
A1 VAR WORD
A1 = 0

HIGH PORTB.7
HIGH PORTE
LOW PORTC
LOW PORTE.0
PAUSE 100
HIGH PORTE.0
PAUSE 500
LCDOUT #1, $FE, 1, "ON"
PAUSE 500
LCDOUT $FE, 1

SPIN:

LCDOUT $FE, $80, "-", $FE, $80+15, "-"
PAUSE 100
LCDOUT $FE, $80, "X", $FE, $80+15, "X"
PAUSE 100
LCDOUT $FE, $80, "*", $FE, $80+15, "*"
PAUSE 100
LCDOUT $FE, $80, "|", $FE, $80+15, "|"
PAUSE 100
LCDOUT $FE, $80, "+", $FE, $80+15, "+"
PAUSE 1

RETURN
```

```

GOSUB SPIN
LCDOUT $FE, $80 + 2, "          "
GOSUB SPIN
GOSUB SPIN
LCDOUT $FE, $80 + 7, "M"
GOSUB SPIN
LCDOUT $FE, $80 + 7, "M7"
GOSUB SPIN
LCDOUT $FE, $80 + 6, "ME7"
GOSUB SPIN
LCDOUT $FE, $80 + 6, "ME07"
GOSUB SPIN
LCDOUT $FE, $80 + 5, "ME 07"
GOSUB SPIN
LCDOUT $FE, $80 + 5, "ME 307"
GOSUB SPIN
GOSUB SPIN

GOSUB SPIN
LCDOUT $FE, $C0 + 3, "          "
GOSUB SPIN
GOSUB SPIN
LCDOUT $FE, $C0 + 7, "G4"
GOSUB SPIN
LCDOUT $FE, $C0 + 6, "GR24"
GOSUB SPIN
LCDOUT $FE, $C0 + 5, "GRO 24"
GOSUB SPIN
LCDOUT $FE, $C0 + 4, "GROU# 24"
GOSUB SPIN
LCDOUT $FE, $C0 + 3, "GROUP # 24"
GOSUB SPIN
GOSUB SPIN
GOSUB SPIN
GOSUB SPIN
LCDOUT $FE, $80 + 1, "          ", $FE, $80 + 2, "S.T.A.C.I.E."
GOSUB SPIN
GOSUB SPIN
LCDOUT $FE, $80 + 1, "          ", $FE, $80 + 1, "S.T.A.C.I.E."
GOSUB SPIN
GOSUB SPIN
LCDOUT $FE, $80 + 1, "          ", $FE, $80 + 2, "S.T.A.C.I.E."
GOSUB SPIN
GOSUB SPIN
LCDOUT $FE, $80 + 1, "          ", $FE, $80 + 3, "S.T.A.C.I.E."
GOSUB SPIN
GOSUB SPIN
LCDOUT $FE, $80 + 1, "          ", $FE, $80 + 2, "S.T.A.C.I.E."
GOSUB SPIN
GOSUB SPIN

FOR I = 15 TO 0 STEP -1
LCDOUT $FE, $C0, "          ", $FE, $C0 + I, "SELF BALANCING"
IF I=15 OR I=12 OR I=9 OR I=6 OR I=3 OR I=0 THEN
    GOSUB SPIN
ENDIF
NEXT I

```

```

GOSUB SPIN

FOR I = 15 TO 0 STEP -1
LCDOUT $FE, $C0, "                ", $FE, $C0 + I, "TRANSLITERATING"
IF I=15 OR I=12 OR I=9 OR I=6 OR I=3 OR I=0 THEN
    GOSUB SPIN
ENDIF
NEXT I
GOSUB SPIN

FOR I = 15 TO 0 STEP -1
LCDOUT $FE, $C0, "                ", $FE, $C0 + I, "AUTOMATED"
IF I=15 OR I=12 OR I=9 OR I=6 OR I=3 OR I=0 THEN
    GOSUB SPIN
ENDIF
NEXT I
GOSUB SPIN

FOR I = 15 TO 0 STEP -1
LCDOUT $FE, $C0, "                ", $FE, $C0 + I, "COMPUTERIZED"
IF I=15 OR I=12 OR I=9 OR I=6 OR I=3 OR I=0 THEN
    GOSUB SPIN
ENDIF
NEXT I
GOSUB SPIN

FOR I = 15 TO 0 STEP -1
LCDOUT $FE, $C0, "                ", $FE, $C0 + I, "IMAGING"
IF I=15 OR I=12 OR I=9 OR I=6 OR I=3 OR I=0 THEN
    GOSUB SPIN
ENDIF
NEXT I
GOSUB SPIN

FOR I = 15 TO 0 STEP -1
LCDOUT $FE, $C0, "                ", $FE, $C0 + I, "EMBARKMENT"
IF I=15 OR I=12 OR I=9 OR I=6 OR I=3 OR I=0 THEN
    GOSUB SPIN
ENDIF
NEXT I
GOSUB SPIN

LCDOUT $FE, $C0, "                "

GOSUB SPIN
GOSUB SPIN
GOSUB SPIN

LOW PORTE.1
PAUSE 100
HIGH PORTE.1
PAUSE 1000
LOW PORTE.2
PAUSE 100
HIGH PORTE.2

MAIN:

```

```
IF PORTD.0 = 1 THEN 'FORWARD
    HIGH PORTC.0
    HIGH PORTC.2
    HIGH PORTC.5
    HIGH PORTC.7
    PAUSE 100
    LOW PORTC
ENDIF

IF PORTD.1 = 1 THEN 'REVERSE
    HIGH PORTC.1
    HIGH PORTC.3
    HIGH PORTC.4
    HIGH PORTC.6
    PAUSE 100
    LOW PORTC
ENDIF

IF PORTD.2 = 1 THEN 'LEFT
    HIGH PORTC.0
    HIGH PORTC.2
    HIGH PORTC.4
    HIGH PORTC.6
    PAUSE 100
    LOW PORTC
ENDIF

IF PORTD.3 = 1 THEN 'RIGHT
    HIGH PORTC.1
    HIGH PORTC.3
    HIGH PORTC.5
    HIGH PORTC.7
    PAUSE 100
    LOW PORTC
ENDIF

GOTO MAIN

END
```

```

'*****
'*  Name      : MOTORS.BAS
'*  Author    : Jonathan Reynolds
'*  Notice    : Copyright (c) 2004
'*            : All Rights Reserved
'*  Date      : 12/7/2004
'*  Version   : 1.0
'*  Notes     :
'*            :
'*****

```

```

DEFINE OSC 10
INCLUDE "MODEDEFS.BAS"

```

```

A0 VAR WORD

```

```

LOW PORTB
LOW PORTC
LOW PORTD

```

```

LOOP:

```

```

SERIN PORTA.0, N2400, A0

```

```

IF A0 = "F" THEN
  HIGH PORTB.3
  HIGH PORTB.1
  HIGH PORTC.5
  HIGH PORTD.3
  PAUSE 200
  LOW PORTB.3
  LOW PORTB.1
  LOW PORTC.5
  LOW PORTD.3
ENDIF

```

```

IF A0 = "B" THEN
  HIGH PORTB.2
  HIGH PORTB.0
  HIGH PORTC.4
  HIGH PORTD.2
  PAUSE 200
  LOW PORTB.2
  LOW PORTB.0
  LOW PORTC.4
  LOW PORTD.2
ENDIF

```

```

IF A0 = "L" THEN
  HIGH PORTB.3
  HIGH PORTB.1
  HIGH PORTC.4
  HIGH PORTD.2
  PAUSE 200
  LOW PORTB.3
  LOW PORTB.1
  LOW PORTC.4

```

```
    LOW PORTD.2
    ENDIF

IF A0 = "R" THEN
    HIGH PORTB.2
    HIGH PORTB.0
    HIGH PORTC.5
    HIGH PORTD.3
    PAUSE 200
    LOW PORTB.2
    LOW PORTB.0
    LOW PORTC.5
    LOW PORTD.3
    ENDIF

PAUSE 50

GOTO LOOP

END
```



```

'*****
'*  Name      : REMOTE.BAS
'*  Author    : Jonathan Reynolds
'*  Notice    : Copyright (c) 2004
'*            : All Rights Reserved
'*  Date      : 12/7/2004
'*  Version   : 1.0
'*  Notes     :
'*            :
'*****

```

```

DEFINE OSC 10
INCLUDE "MODEDEFS.BAS"
TRISA = %11111111

```

```

A0 VAR WORD
I VAR WORD

```

```

HIGH PORTB.7
LOW PORTA

```

```

LOOP:

```

```

A0 = "Q"

```

```

IF PORTA.0 = 1 THEN
  A0 = "F"
  GOTO SEND
ENDIF

```

```

IF PORTA.1 = 1 THEN
  A0 = "B"
  GOTO SEND
ENDIF

```

```

IF PORTA.2 = 1 THEN
  A0 = "L"
  GOTO SEND
ENDIF

```

```

IF PORTA.3 = 1 THEN
  A0 = "R"
  GOTO SEND
ENDIF

```

```

SEND:

```

```

FOR I = 0 TO 20
SEROUT PORTC.3, N2400, [A0]
PAUSE 100
NEXT I

```

```

GOTO LOOP:

```

```

END

```

```

'*****
'* Name      : SENSOR.BAS                               *
'* Author    : Jonathan Reynolds                       *
'* Notice    : Copyright (c) 2004                     *
'*           : All Rights Reserved                     *
'* Date      : 12/7/2004                               *
'* Version   : 1.0                                     *
'* Notes     :                                         *
'*           :                                         *
'*****

DEFINE OSC 10
INCLUDE "MODEDEFS.BAS"

' Define ADCIN parameters
DEFINE ADC_BITS      10      ' Set number of bits in result
DEFINE ADC_CLOCK     3      ' Set clock source (3=rc)
DEFINE ADC_SAMPLEUS  8      ' Set sampling time in uS

TRISA = %11111111
TRISD = %11111111
TRISC = %00000000

adval  VAR      WORD      ' Create adval to store result
I      VAR WORD
A0     VAR WORD
A1     VAR WORD
V      VAR WORD
D      VAR WORD
M      VAR WORD
A0 = 0
V = 0
D = 0
M = 0

ADCON1 = %10000010      ' Set PORTA analog and right
justify result

HIGH PORTB.7

loop:

    PAUSE 100

    FOR I = 1 TO 100

        ADCIN 0, adval      ' Read channel 0 to adval

        A1 = A1 + adval / 2
        A0 = A1

    NEXT

    A1 = A0 / 100
    A0 = A1 * 2

```

```

V = (48 * A0 + 19) / 10
IF (V < 2800) && (V > 800) THEN
GOSUB CURVE2
ENDIF
IF (V < 800) THEN
GOSUB CURVE3
ENDIF

SEROUT PORTB.6, N2400, [#D]

GOTO loop ' Do it forever

CURVE2:
    D = 3510 + 2600 * (SIN (V / 10 + 5 / 2) ) + 1300 * (COS (V / 10 + 5 / 2)
)
    RETURN

CURVE3:
    D = (1156 - V) / 22
    RETURN

END

```