unidrive®

FIRE FIGHTING ROBOT

Jake Bian & Dennis Bellinger

PREPARED FOR: C.CSANITS

JAN. 23, 2013

2 Table of Contents

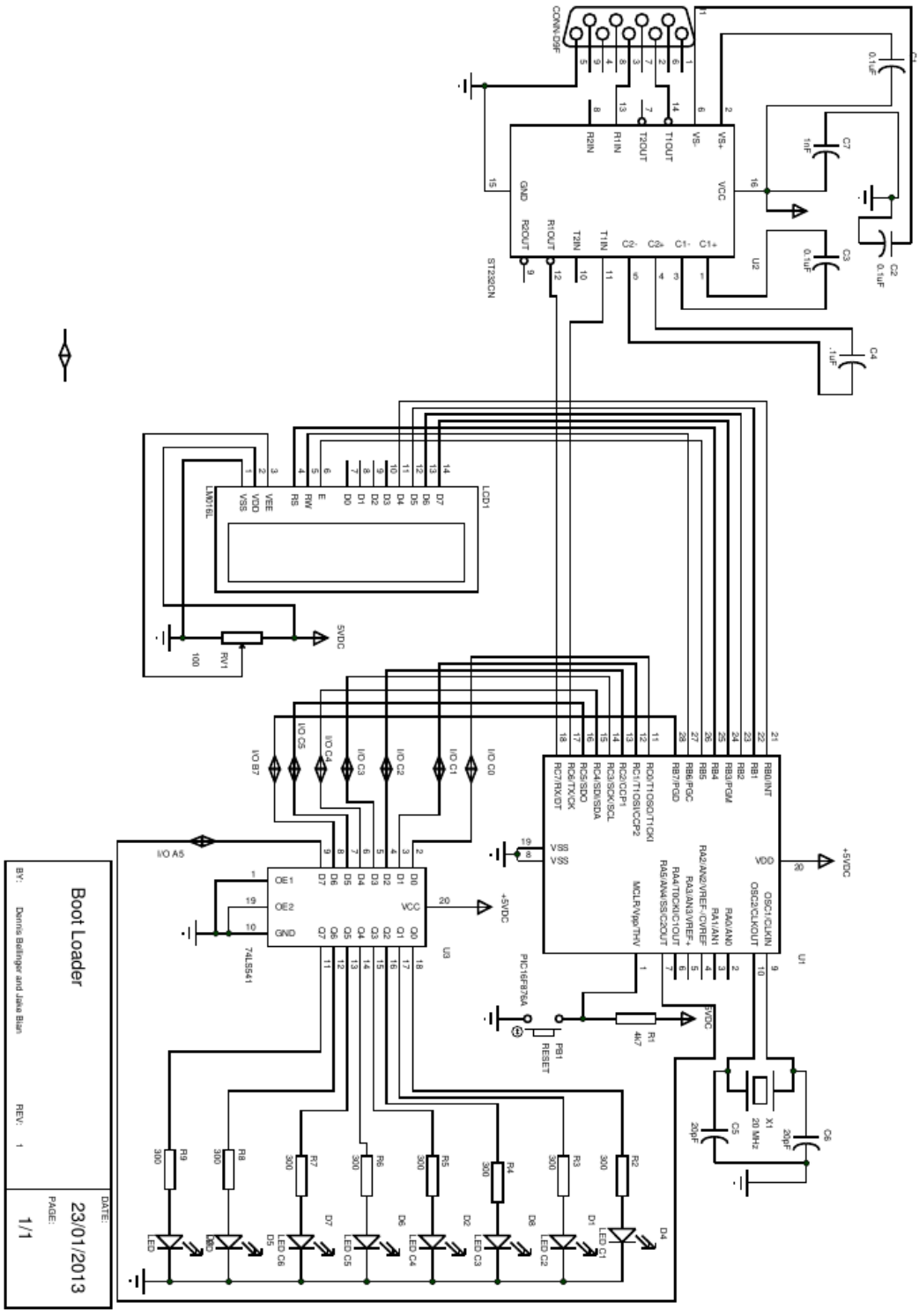
3 Introduction:

The idea of the Unidrive is the result of a pursuit for efficiency and precision in small robotic navigation systems. The team was given the task to design and build a robot capable of autonomous navigation in a small maze with the ultimate task of detecting and extinguishing a candle flame. The design offers an innovative method of locomotion which employs a single motorized wheel whose orientation is controlled by a servo fixed to the body of the robot. (See Section 4.0) The elimination of a turning radius as a direct result of this design allows the position of the robot to be easily computed. By its nature also eliminates the need to synchronize multiple wheels. The precise control of the direction of the linear motion also allows the robot to follow the most efficient path - a linear one, to the candle flame, or any other desired position. The project was initiated after the conception of the design in September 2012. Through the course of the construction and programming of the robot, despite the many unforeseen problems encountered and the subsequent failure to follow the planned timeline, most remaining aspects of the original hardware design were followed. Due to the time constraints and the magnitude of the project, the robot, as of January 2013, is incapable of performing the desired tasks. Nonetheless the knowledge and experience gained from the project have proved to be invaluable to its participants. This report aims to discuss the process of the design, construction, and programming of the robot, the problems encountered, and their respective solutions.

Section 4: Drawings

The schematics and PCB designs for the circuit boards on the unidrive are reproduced below. Also included are a few original designs that lead to the creation of the Unidrive firefighting robot.

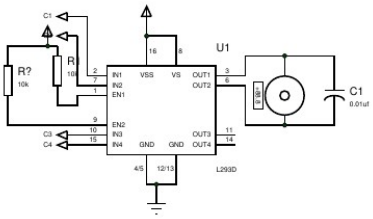
4.1 Motherboard Schematic



4.2 Motherboard PCB Design

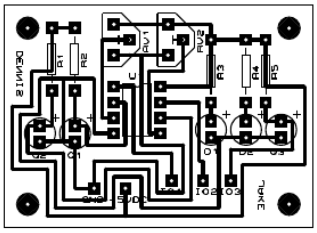
TITLE:	DATE:
UNIDRIVE BOT MOTHERBOARD	23-10-12
BY: DENNIS & JAKE REV:1	PAGE: 1

4.3 Motor Driver Schematic



TITLE: UNIDRIVE BOT MOTOR DRIVER BOARD BY: DENNIS & JAKE REV: 1	DATE: 23-10-12 PAGE: 1

4.4 Line and Candle Detection PCB Design

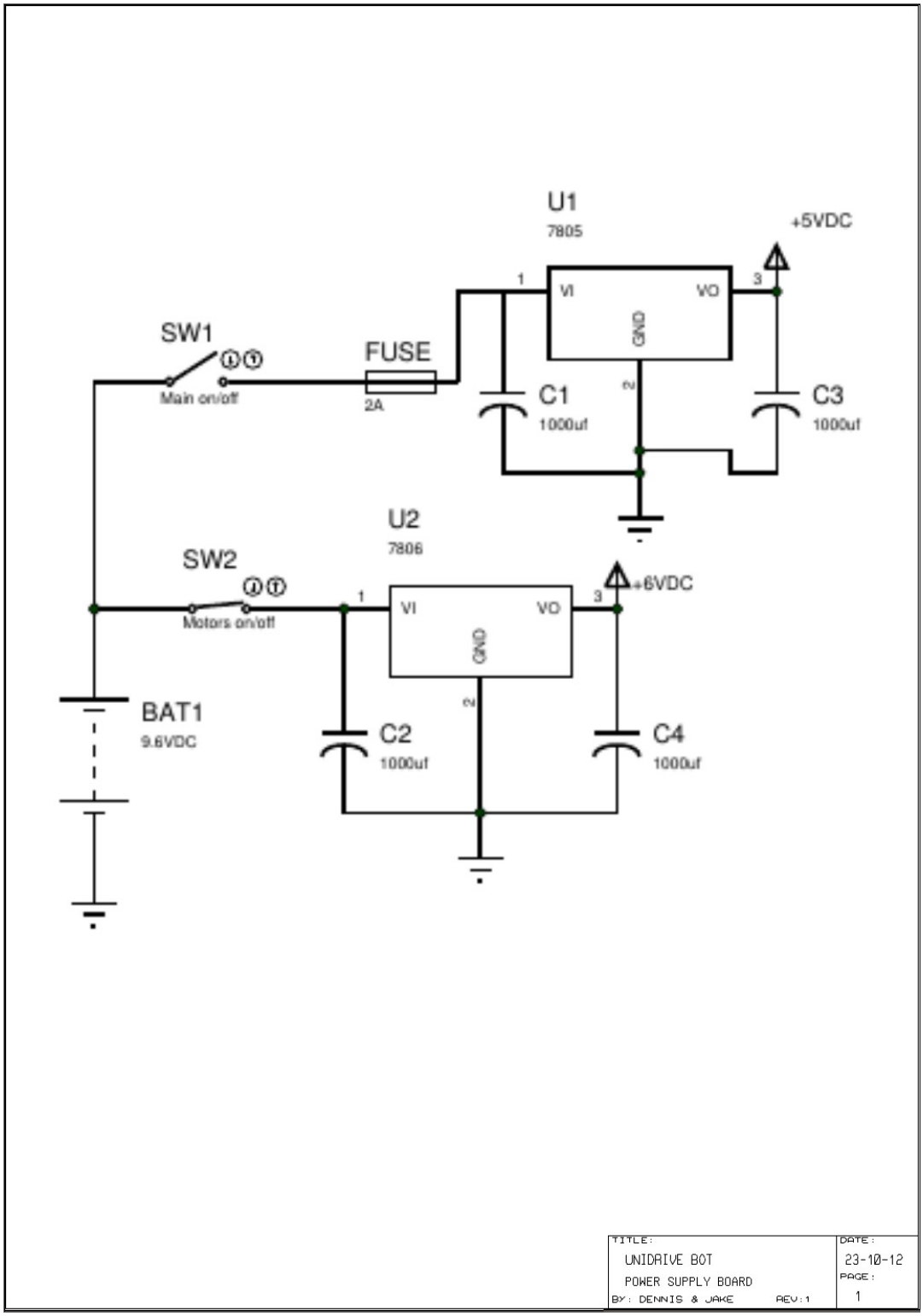


Bill of Materials

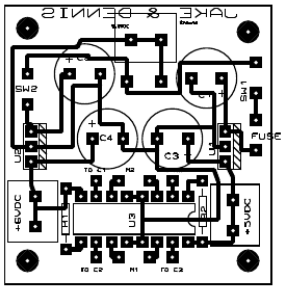
Q1 - 3MM SUPERBRIGHT LED
Q2 - 3MM SUPERBRIGHT LED
Q01 - PHOTOTRANSISTOR LEFT SIDE
Q2 - PHOTOTRANSISTOR RIGHT SIDE
Q3 - INFRARED PHOTOTRANSISTOR
IO1 - LINE DETECTION LEFT SIDE OUTPUT
IO2 - LINE DETECTION RIGHT SIDE OUTPUT
IO3 - CANDLE SENSOR OUTPUT
U1 - LM358N LOW POWER DUAL OPERATIONAL AMPLIFIER
R1/R2 - 10k
R3/R4 - 220
R5 - 47k
RV1/RV2 - 10k VARIABLE RESISTANCE POTENTIOMETER

TITLE:	DATE:
LINE AND CANDLE DETECTION	2012-12-03
BY: DENNIS BELLINGER AND JAKE BIAN REV:2	PAGE: 1 OF 1

4.5 Power Supply Schematic



4.6 Power Supply PCB Design



BILL OF MATERIALS

C1-C4	1000uF Electrolytic Capacitors
R1, R2	10k Ohms Resistors
U1	+5V Power Regulator
U2	+6V Power Regulator
U3	Four channel motor driver
M1,M2	connections to DC motors

TITLE: UNIDRIVE BOT POWER SUPPLY BOARD	DATE: 23-10-12
BY: DENNIS & JAKE REV:1	PAGE: 1

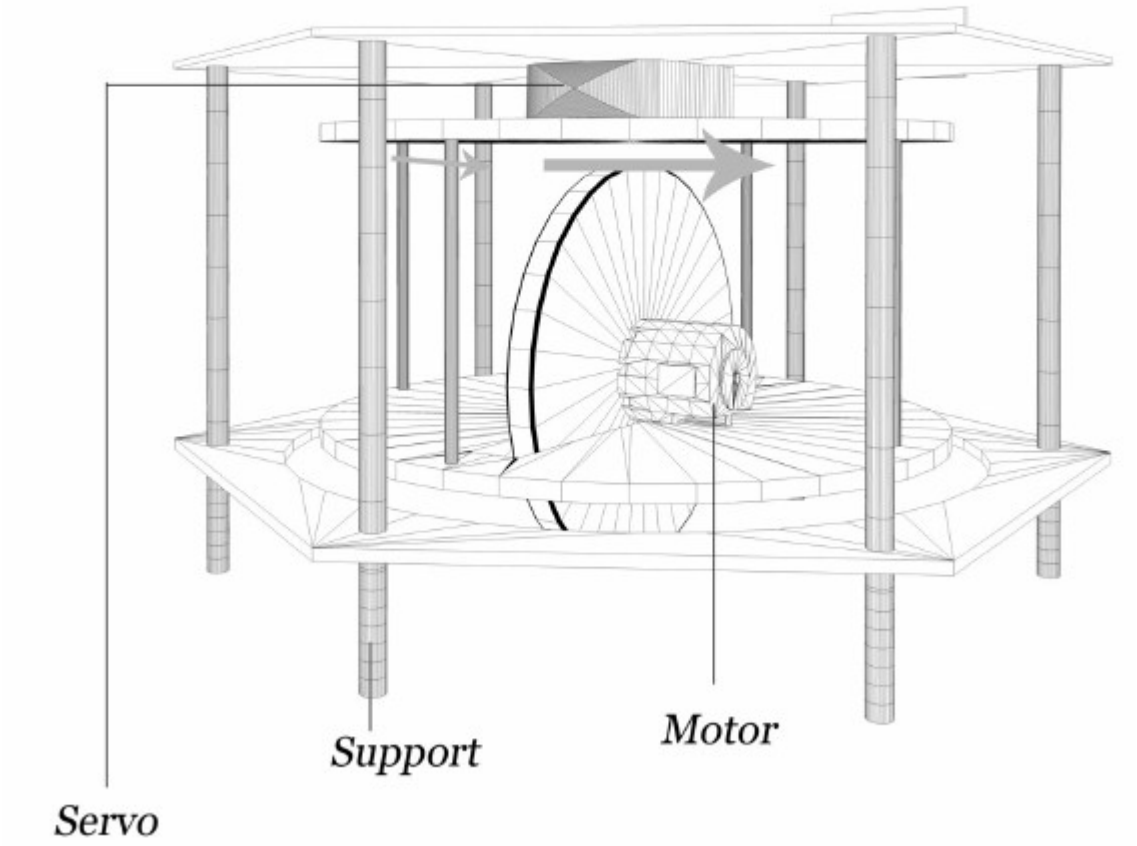
4.7 PCB Design Printouts

This contained physical PCB printouts

(used for etching the PCBs)

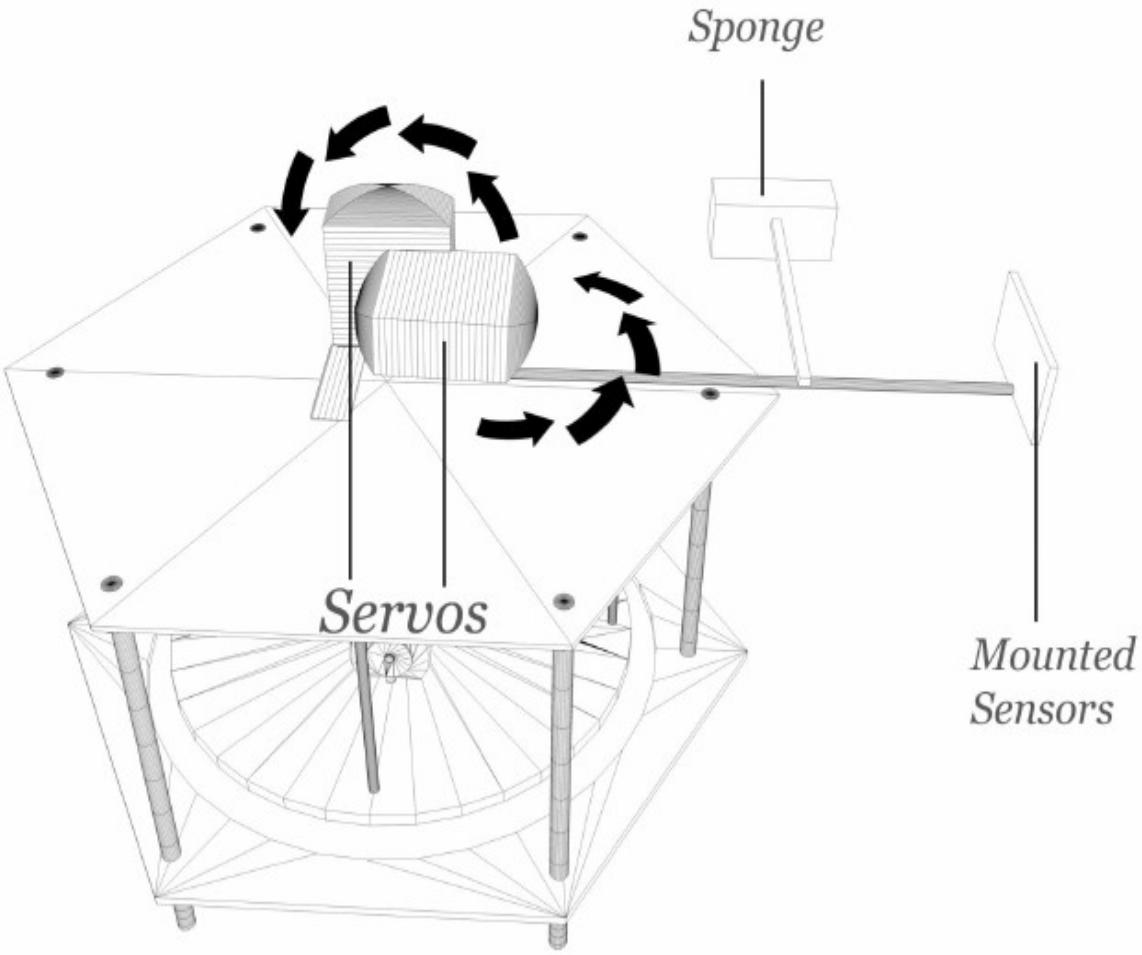
4.8 Original Design: Movement Mechanism

FIGURE 1: MOVEMENT MECHANISM



4.9 Original Design: Arm Rotation

FIGURE 2: Rotating Arm



4.10 Original Design: Side Views

FIGURE 3.1: SIde View

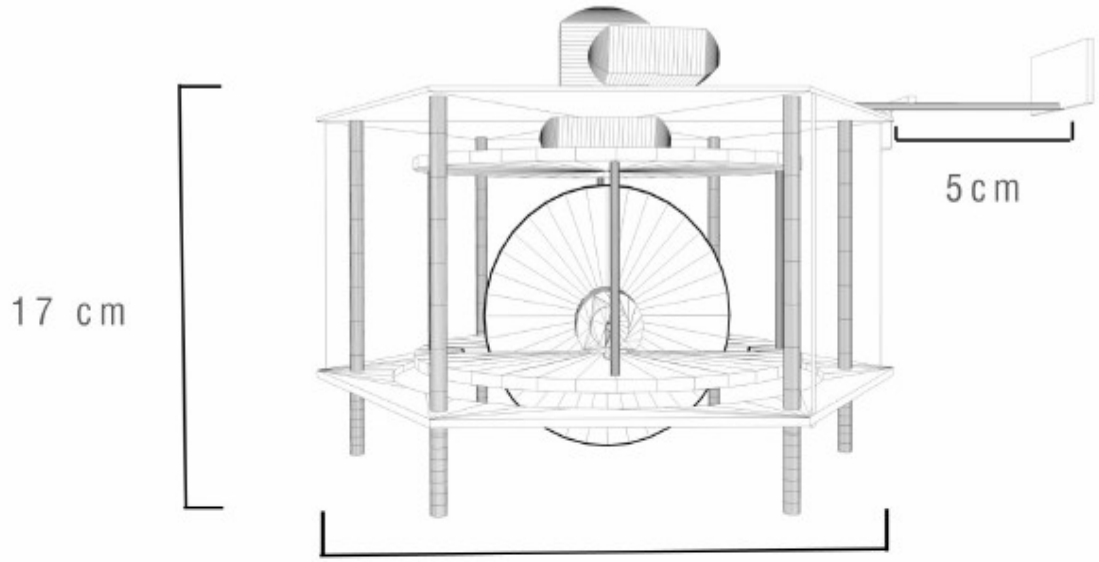
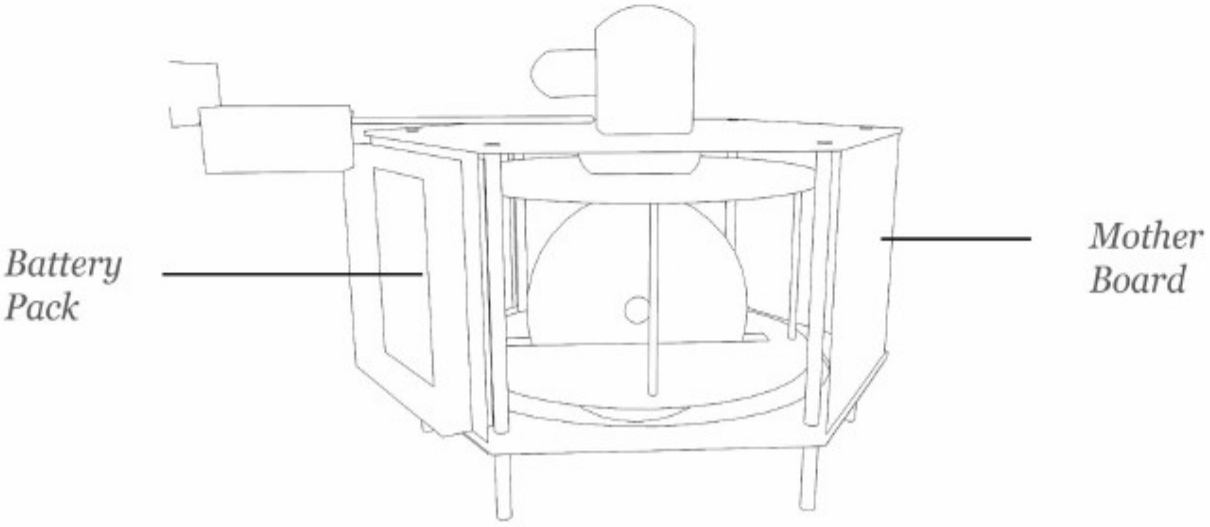


FIGURE 3.2 Side View Rotated



4.11 Original Design: Top and Bottom Views

FIGURE 4.1: Top View

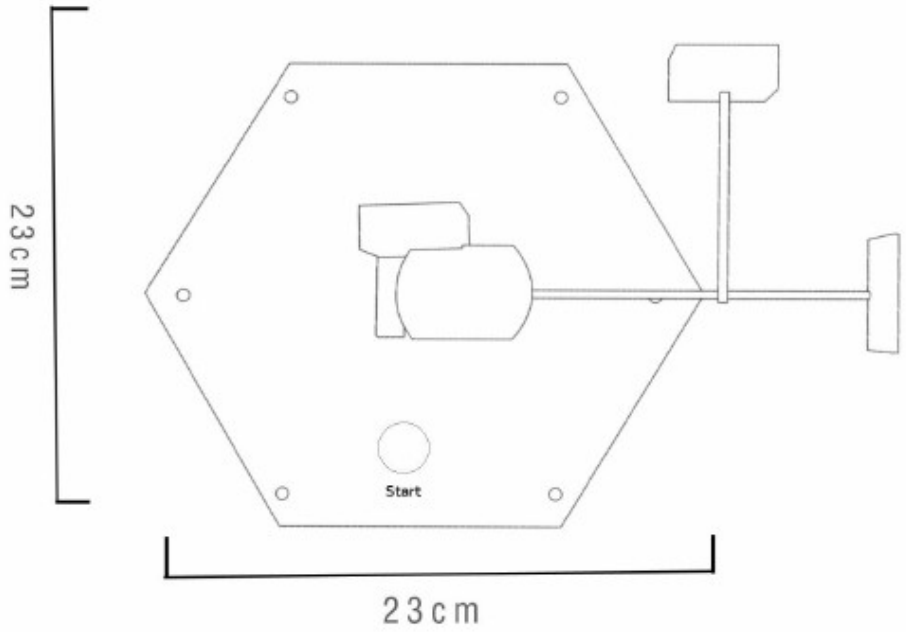
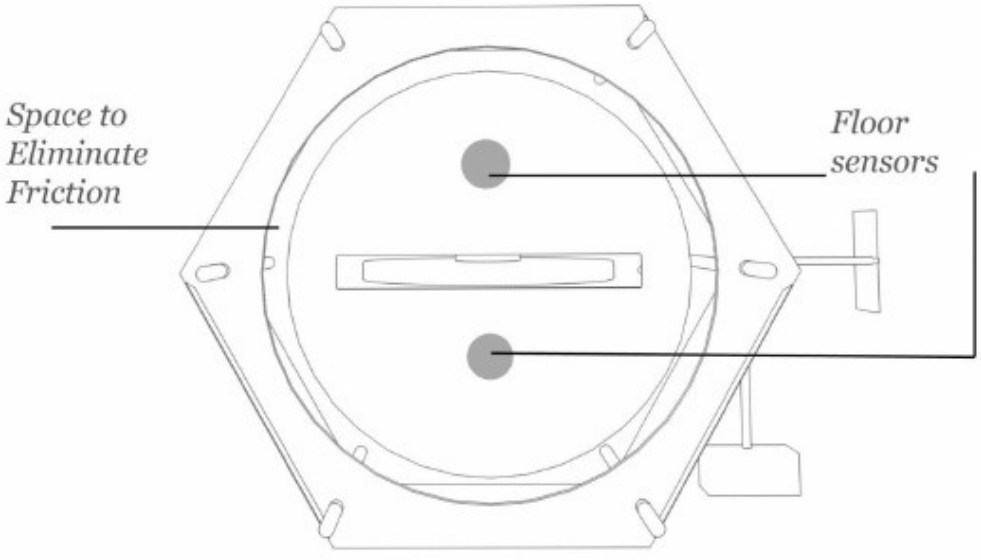


FIGURE 4.2: Bottom View



Section 5: Problems:

Apart from a variety of electrical issues, the majority of the problems encountered during this project were caused by the frictional forces between the legs and the surface of motion. Our assumption that the robot will move in a straight line was proven to be erroneous.

5.1. Actualization of linear motion.

In early November 2012, it became apparent that the robot had a tendency to move in a circular path. The frictional force exerted on the robot by the surface of its motion correlates directly with the normal force exerted on the ground. A large portion of this force is caused by the torque exerted by other objects on the base. We were then presented with the problem of determining the impact of a force on a point in a shared plane outside of its line of action, wherein line of action is a unit vector whose direction is defined by the point where the force is applied and the center of mass: the hollow center of the base. A bold attempt was made to generate individual lines of actions from the vector connecting the originating force to the desired point, assuming the lever arms have equal lengths and consequently the axis of rotation becomes the midpoint of the vector. A set of equations were derived in Euclidean R3 and measurements were taken for the implementation of counter balances. However, it was found that an accurate result obtained from this calculation requires extreme precision in the measurement, after several attempts of physical implementation, this theoretical method was abandoned and an experimental method was introduced. The LCD display was mounted on a static arm at a large radius from the center of the robot in order to balance the weight of the battery pack. The length of this lever arm proved to be too great and it became

necessary to mount the battery on another lever arm in the opposite direction. This issue was resolved in January 2013.

5.2. Motion on maze surface

In January 2013 it was discovered that the legs of the robot generated a frictional force with the floor of the maze with a magnitude too large for the motor of the robot to overcome. This issue was eliminated by attaching buttons to the bottom of the robot.

5.3. Friction control through adjustment of vertical (perpendicular to the surface) position of the wheel

The wheel need to supply a large enough amount of friction for it to move the robot, yet the robot demanded a certain amount of friction from the legs so that the body of the robot will not turn during the rotation of the servo controlling the motor. For this to take place the wheel needed to be secured at a certain vertical position, the experimental process of determining the ideal position consumed a large amount of time, it was achieved in early January.

5.4 Candle detection and approach through trigonometry

Due to the physical constraints of the bot, including the aforementioned experimentally determined balance system, the servo to control the motor/wheel assembly and those used to control the arm were not lined up. This resulted in an unforeseen complication with the candle detection and extinguishment. The method by which the bot extinguishes the candle (a wet sponge) requires the bot to be able to accurately approach the candle and determine when it is appropriately close. As a result of the frictional difficulties it was determined that it is not possible to simple drive in any direction, as the balance is only stable on two planes, perpendicular to both the maze floor and each other. This resulted in the need to drive in a stepping

pattern (tracing the second and third sides of a set of triangles of which the hypotenuse is a path leading to the candle). Also contributing to this difficulty is the necessity to adjust the servo positioning between the candle detection and approach subroutines. These two unexpected complications led to the necessity to convert the servo position to degrees and the use trigonometry to determine the path of the bot. This was overcome theoretically in January 2013.

6 Software/Firmware

```
CPU = 16F876A
MHZ = 20
CONFIG 16242

' Software/Firmware for the unidrive single-wheeled firefighting robot
' Copyright 2013 Dennis Bellinger and Jake Bian

' The following code describes navigation through the maze, examination
of
' rooms, and the process of locating, approaching, and extinguishing the
' candle.
' Prepared for Mr. C. Csanits on January 23, 2013


'Hold the direction
dirone var bit
dirtwo var bit

'' Arm direction
adirone var bit
adirtwo var bit
armcorr var word

time var word

waiting var bit
inRoom var bit
candleFound var bit
candlepos var word
vcandlepos var bit
walldist var word
rdirone var word
rdirtwo var word
approaching var bit

armpos var word
previousir var word
warming var bit
newir var word

harm con C0
varm con C1
serv con C2
motorf con C3
motorr con C4
sensorone con C5
sensortwo con B7
irsens con A0
wallsens con A1
```

```
sensorir var word
sensorwall var word

'The variable that stores the motorcorrection for the motor servo
motorcorr var word

'Hold the current location, relative to home (0,0)
posx var word
posy var word


Main

'' Start out by initializing (variables and such)
gosub init

gosub followWall
gosub turnright
gosub followWall
gosub turnright
gosub checkRoom
gosub turnleft
gosub followWall
gosub turnright
gosub followWall
gosub turnright
gosub checkRoom
gosub turnleft
gosub followWall
gosub turnright
gosub checkRoom
gosub turnright
gosub followWall
gosub turnright
time = 200
gosub forward
gosub wait
gosub brake
gosub turnleft
gosub checkRoom
gosub turnright
gosub followWall
gosub turnright
gosub followWall
gosub done


' finish the trial
''
''    Need a wait loop here...
''     (occupy the processor infinitely)
```

```
''

'' Inform the operator that the bot is finished
done:
    LCDWRITE B4\B5\B6, portb.nib0, [INITLCD1, INITLCD2, TWOLINE,CLEAR,
HOME, SCR]
    LCDWRITE B4\B5\B6, portb.nib0, ["   Routine"]
    LCDWRITE B4\B5\B6, portb.nib0, [scrram+$40]
    LCDWRITE B4\B5\B6, portb.nib0, ["   Completed"]
    finish:
       pause 20000
    goto finish
return

'' This subroutine gets called when one of the sensors picks up a line
linecheck:
    if (sensorone = 1) then
       inRoom = inRoom XOR 1
    endif
return

'' This may not be necesarry (may be superseeded)
wait:
    int var word
    int = 0
    time=time/20
    for int=1 to time
       gosub periodic
       pause 20
    next
return

'' A small subroutine to update the position of the bot
fixpos:
    if dirone = 0 then
       if dirtwo = 0 then
             posy = posy + 1
       else
             posx = posx +1
       endif
    else
       if dirtwo = 0 then
             posy = posy - 1
       else
             posx = posx - 1
       endif
    endif
return

checkRoom:
    gosub turnaround
    return
```

```
    if (candleFound = 1) then
       gosub turnaround
       return
    endif
    gosub forward
    repeat
       gosub periodic
    until (inRoom = 1)
    time = 1000
    gosub wait
    gosub brake
    gosub recordLoc
    gosub candlescan
    if (candleFound = 1) then
       gosub extinguishcandle
    endif
    gosub exitroom
return

' Record the current relative location
recordLoc:
    ' Save teh distance to the wall
    gosub armleft
    gosub checksensors
    walldist = sensorwall

    ' Save the current direction
    rdirone = dirone
    rdirtwo = dirtwo
return

' Scan for the candle
candlescan:
    pulsout varm,1000
    pause 20
    gosub swingarm
    if candlepos = 0 then
       pulsout varm,2000
       vcandlepos = 1
    else
       vcandlepos = 0
    endif
    if candlepos <> 0 then
       candleFound = 1
    endif
return

' Extinguish the candle, if found
extinguishcandle:
    approaching = 1
    gosub approachcandle
    gosub hitcandle
```

```
    approaching = 0
return

'Approch the candle
approachcandle:
       ' Follow a stepping pattern towards the candle

    deg var word
    ' This leaves an angle in degrees of the direction of the candle
    ' it uses a (somewhat) aribitrary start point
    deg = (((candlepos-1000)*180)/(2000-1000))
    ' Correct for servo misalignment
    deg = deg + 25 'Add 25 degrees, needs calibration (difference between
             ' wheel and vertical arm servo 0 deg positions)
    if (deg > 359) then 'Ensure that the calibration does not result in
difficult angles)
       deg = deg - 360
    endif

    'Assign wheel directions
    if dirone = 1 then
       deg = deg - 180
       if deg < 0 then
             deg = deg + 360
       endif
    endif
    if dirtwo = 1 then
       deg = deg - 90
       if deg < 0 then
             deg = deg + 360
       endif
    endif

    ' Calculate path, using trig
    horiz var word
    horiz = sin deg
    vert var word
    vert = cos deg

    ' First check to see if the candle lies directly ahead, behind or to
one side
    if (horiz = 0) then ' candle is directly infront (or we ran over it
coming in...)
       gosub forward
       repeat
             gosub checksensors
             pause 5
       until (sensorir < 20)
       gosub brake
       return
    endif
    if (vert = 0) then sensor is to one side
```

```
      if (horiz > 0) then ' candle is to the left
            gosub turnleft
            gosub forward
            repeat
                  gosub checksensors
                  pause 5
            until (sensorir < 20)
            gosub brake
            return
      endif
      if (horiz < 0) then ' candle is to the right
            gosub turnright
            gosub forward
            repeat
                  gosub checksensors
                  pause 5
            until (sensorir < 20)
            gosub brake
            return
      endif
   endif

   ' Follow stepping pattern to approach candle
   ahhoriz var word
   avert var word
   ahoriz = (abs horiz)
   avert = (abs vert)
   repeat
      if (horiz < 0 ) then
            gosub turnright
      else
            gosub turnleft
      endif
      time = ahoriz
      gosub forward
      gosub wait
      gosub brake

      if (vert < 0) then
            gosub turnright
      else
            gosub turnleft
      endif
      time = avert
      gosub forward
      gosub wait
      gosub brake
   until
return

'put out the candle with the sponge
hitcandle:
```

```
    tmp var word

    ' Set the horizontal arm to 200 below the candle point
    tmp = 0
    repeat
    pulsout harm,(candlepos - 200)
    pause 20
    tmp = tmp + 20
    until tmp = 200

    ' Make final approach
    time = 200
    gosub forward
    gosub wait
    gosub brake

    'Set the vertical arm to the right position
    tmp = 0
    repeat
    pulsout varm,vcandlepos
    pause 20
    tmp = tmp + 20
    until tmp = 200
    ' hit the candle
    pulsout harm,candlepos

return

'Exit the room
exitroom:
    ' Restore the saved settings
    dirone = rdirone
    dirtwo = rdirtwo

    ' turn to face the wall we measured to
    gosub turnleft

    ' Drive toward wall until we are the same distance as before
    gosub forward
    repeat
       gosub checksensors
       pause 20
    until ((walldist = sensorwall) or (walldist < sensorwall))
    gosub brake
    gosub turnleft
    gosub forward
    repeat
       gosub checksensors
       gosub linecheck
    until sensorone = 1
    pause 500
    gosub brake
```

```
return

'Swing the arm from 0 deg to 180 deg
swingarm:
    candlepos = 0
    armpos = 1000
    repeat
       pulsout harm,armpos
       armpos = armpos + 10
       pause 20
       gosub checksensors
       if ((previousir > (newir + 20)) and (sensorir > (newir + 20))) then
             candlepos = (armpos - 10)
             return
       endif
       previousir = newir
       newir = sensorir
    until (armpos = 2000)
return

'Swing the arm from 180 deg to 0 deg
swingarmrev:
    candlepos = 0
    armpos = 2000
    repeat
       pulsout harm,armpos
       armpos = armpos - 10
       pause 20
       gosub checksensors
       if ((previousir > (newir + 20)) and (sensorir > (newir + 20))) then
             candlepos = (3800 - (armpos + 10))
             return
       endif
       previousir = newir
       newir = sensorir
    until (armpos = 1000)
return

'' Follow a wall until the end
followWall:
    ' Point arm at right wall
    gosub armright

    ' Get distance to wall
    gosub periodic
    dist var word
    dist = sensorwall

    ' Start driving forward
    gosub setdir
    gosub forward
```

```
    ' repeat until end of wall
    repeat

       ' If the bot is moving away from the wall
       if sensorwall < dist - 10 then
            motorcorr = (motorcorr + 10)
            pause 20
       endif

       ' If the bot is moving towards the wall
       if sensorwall > dist + 10 then
            motorcorr=motorcorr-10
            pause 20
       endif

       ' Make sure the servo stays in the right position
       gosub periodic
    until sensorwall < 75

    ' Wall is done, stop bot
    pause 500
    gosub brake

return

'' Below this point is only low level control code
'' NO MAPPING OR FIREFIGHTING LOGIC BELOW THIS POINT

'' dir 1 and 2 represent the direction arm follows same pattern
' dirone dirtwo Direction
'    0    0       North
'    0    1       East
'    1    0       South
'    1    1       West

init:

    ' Set all ports to low
    low C0 'Horizontal arm control
    low C1 'Vertical arm control
    low C2 'motor Servo
    low C3 'Motor forward
    low C4 'Motor Reverse
    low C5 'Line sensor 1
    low B7 'Line sensor 2
    low A5 'Empty

    '' Set the line sensors to input
    Input C5
    Input B7

    '' Also set the IR sensor (candle detection)
```

```
    '' and wall sensor as inputs
    Input A0 'Candle sensor (IR)
    Input A1 'Wall sensor

    'Set the initial direction to North
    '(this is an arbitrary direction)
    dirone = 0
    dirtwo = 0

    ' Set the arm also
    adirone = 0
    adirtwo = 0

    ' Make sure that home is indeed the origin (0,0)
    posx = 0
    posy = 0

RETURN

forward:
    if dirone =1 then
       high motorr
       low motorf
       'high C4
    elseif dirone = 0
       high motorf
       low motorr
       'high C5
    endif
return

brake: 'Make sure that the bot actually stops moving
    low motorr 'This makes sure that the motor cannot break itself
    high motorf         '(Would occur through trying to turn both ways)
    pause 5
    low motorf
    high motorr
    pause 5
    low motorr

return

turnleft:
    gosub brake
    if dirtwo=0 then
       dirtwo=1
       dirone = dirone XOR 1
    elseif dirtwo = 1
       dirtwo = 0
    endif
    gosub setdir
return
```

```
turnright:
    gosub brake
    if dirtwo = 1 then
       dirtwo = 0
       dirone = dirone XOR 1
    elseif dirtwo = 0
       dirtwo = 1
    endif
    gosub setdir
return

turnaround:
    gosub brake
    dirone = dirone XOR 1
    gosub setdir
return

setpos:    'This actually sets the servo to the desired position
    if dirtwo = 0 then
       pulsout serv,625 ' + motorcorr
    elseif dirtwo = 1
       pulsout serv,1475 ' + motorcorr
    endif
return



setdir:
    num var word
    num = 0
    for int=1 to 100
       gosub setpos
       pause 20
    next
return

' This subroutine simply collects data
checksensors:
    adsetup con %10000000 ' Pin configuration information
    clk con 2
    adin irsens,clk,adsetup,sensorir
    adin wallsens,clk,adsetup,sensorwall
return

armright:
    if dirtwo = 1 then
       adirtwo = 0
       adirone = adirone XOR 1
    elseif dirtwo = 0
       adirtwo = 1
    endif
```

```
    gosub setarmdir
return

armleft:
    if dirtwo = 0 then
       adirtwo = 0
       adirone = adirone XOR 1
    elseif dirtwo = 1
       adirtwo = 0
    endif
    gosub setarmdir
return

setarmdir:
    num=0
    for int=1 to 100
       gosub setarmpos
       pause 20
    next
return

setarmpos:    'This actually sets the servos to the desired position
    if (approaching = 1) then
       return
    endif
    pos var word
    if adirone = 0 then
       '' Arm vertical control at 0 deg
       pos=1000 '+armcorr
       'pulsout varm,pos
    elseif adirone = 1
       '' Arm vertical control at 180 deg
       pos=2000 '+ armcorr
       'pulsout varm,pos
    endif

    if adirtwo = 0 then
       '' Arm horizontal control at 0 deg
       pos=1000 ' + armcorr
       pulsout harm,pos
    elseif adirtwo = 1
       '' Arm horizontal control at 90
       pos=1475 ' + armcorr
       pulsout harm,pos
    endif
return

printsensors:

    LCDWRITE B4\B5\B6, portb.nib0, [INITLCD1, INITLCD2, TWOLINE,CLEAR,
HOME, SCR]
    LCDWRITE B4\B5\B6, portb.nib0, ["Line 1:"]
```

```
    LCDWRITE B4\B5\B6, portb.nib0, [DEC sensorone]
    LCDWRITE B4\B5\B6, portb.nib0, [scrram+$0a]
    LCDWRITE B4\B5\B6, portb.nib0, ["2:"]
    LCDWRITE B4\B5\B6, portb.nib0, [DEC sensortwo]
    LCDWRITE B4\B5\B6, portb.nib0, [scrram+$40]
    LCDWRITE B4\B5\B6, portb.nib0, ["IR:"]
    LCDWRITE B4\B5\B6, portb.nib0, [DEC sensorir]
    LCDWRITE B4\B5\B6, portb.nib0, [scrram+$47]
    LCDWRITE B4\B5\B6, portb.nib0, ["Wall:"]
    LCDWRITE B4\B5\B6, portb.nib0, [DEC sensorwall]

return

periodic:
    gosub checksensors
    gosub setpos
    gosub setarmpos
    gosub fixpos
    gosub printsensors
    gosub linecheck
    pause 20
return
```

7. 1 Conclusion (Jake Bian)

Despite its functional failure, the idea of the Unidrive introduces a navigation method which offers precision and efficiency, an innovation which has potential industrial applications beyond the particular project. Overall this project consisted of design and production of electronic components, construction of a robotic system using limited materials, rigorous physical calculations operating with parametric vector equations in Euclidean R3 and programming in a low-level platform.

I hold a thorough appreciation for the lessons learned through this project. One of the most significant discoveries in this the project is the realization that the efficiency in computation and navigation required a compromise in the efficiency of motorized motion - the motor must overcome the force of friction of the robots, a fact which reduces the robot's speed and therefore, power efficiency (wherein kinetic energy is the desired energy form). I am in the process of seeking a solution to that problem which does not require another level of hardware complexity. Through this project I was familiarized with the process of hardware construction and the time required to eliminate existing issues, I believe this knowledge will assist me greatly in future engineering projects.

In regards to the design of the project, I feel that It would be ideal if the student's ability to design software was evaluated independently of the state of their hardware. However I have yet to think of a way to accomplish that abstaining from a complete restructuring of the course.

Despite my deep regret for our failure to achieve the desired result on the deadline, I am grateful for the opportunity to implement this idea and hope to continue to develop this innovation in the future.

7.2 Conclusion (Dennis Bellinger)

The main design feature and the whole reason for restructuring the design of the fire fighting robot from the standard two wheeled car, was to achieve consistent, predictable, and accurate turning. The Unidrive single-wheeled fire fighting robot excelled in this area. The changeup in design resulted in many unexpected and often unpredictable behaviours, including the inability to drive in an even remotely straight line. Also as the robot was built around a completely different design philosophy it had a very different shape and layout resulting in complications ranging from assembly to reassembly.

Despite (or because of) the many difficulties we encountered while designing and constructing this robot, I have learned a great deal about engineering in general, and specifically mechanical engineering. One of the things that I learned that I feel should have been more emphasized in this course as well as other areas of life and school is that just because something looks good on paper that is not any indication that is will work or perform well in the physical world.

I am grateful for the opportunity to have learned the value of failure, and believe that far more was learned (not just by us, but by everyone else in the class too), by the problems that were presented and attempted (even if they were not completely overcome in time), than if everything had gone smoothly. In hindsight I wish that I had pursued the idea of using gyroscopes to stabilize the bot, as two carefully positioned gyroscopes would have eliminated the need for feet (and the robots ability to rotate). If I were asked to complete the project again, I would once again attempt a single wheeled design, as I feel that the benefits far outweigh the drawbacks, and I would make an attempt at using gyroscopes to solve most of the drawbacks.