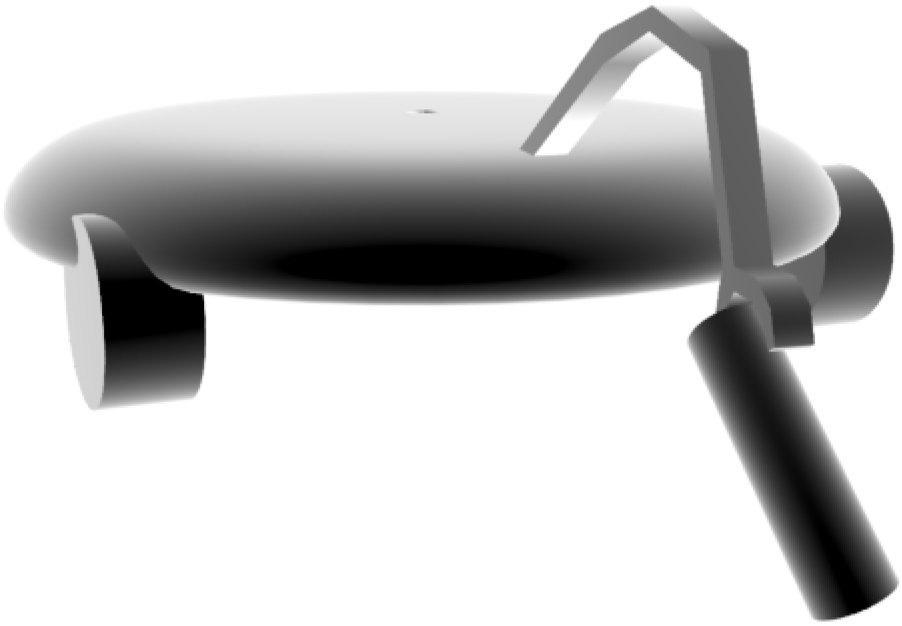
EECS 149: Embedded Systems

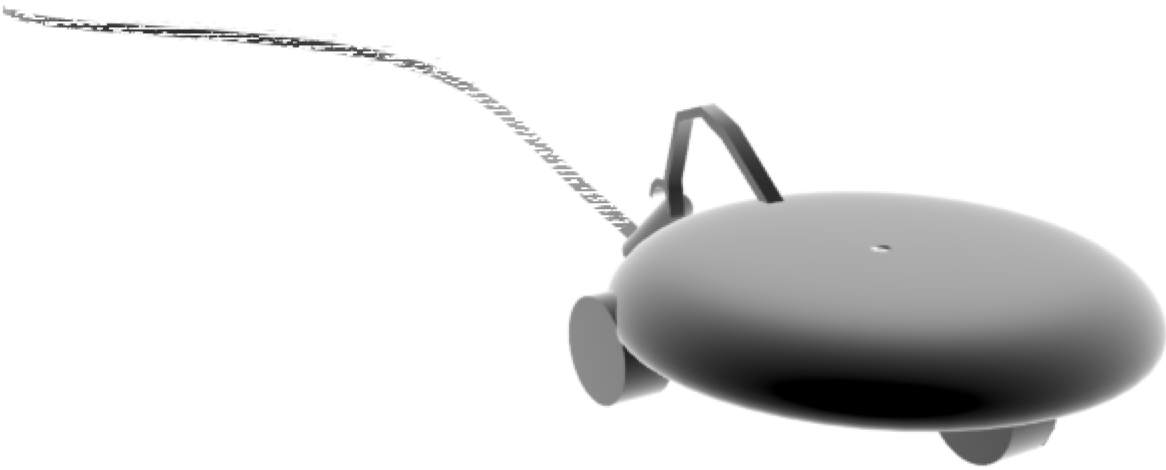
Project: Chalk-Drawing iRobot

Professor: Edward Lee
GSI: Shanna-Shaye Forbes
Students: Andre Zeumault, Skot Croshere, David Burban, Dan Lynch



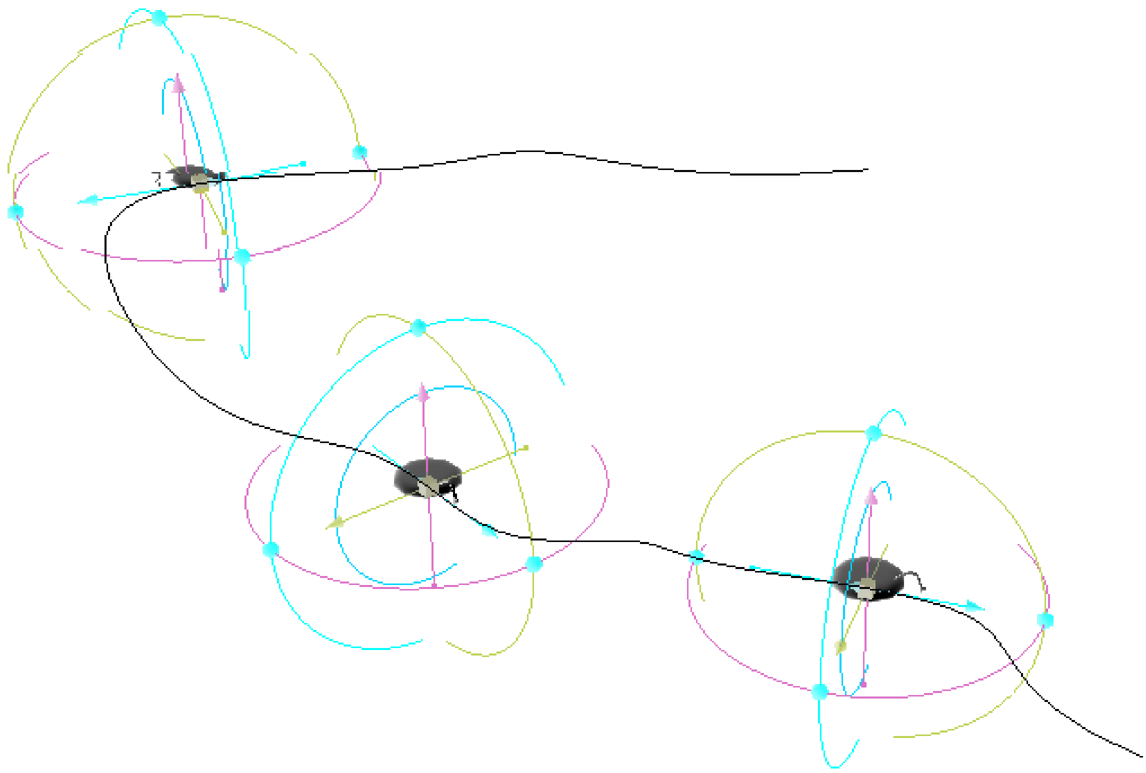
Introduction

Our idea is to take an ordinary iRobot, and mount a robot arm for drawing images with chalk. The paths will be generated via XML parsing of the SVG file format. The robot will drive along in a parking lot or other smooth surface and draw the contents of the SVG file, scaled to the physical size of the parking lot. We think of it as a free-form printer.



Simulation

In order to optimize the process, our group will incorporate what we have learned from our simulation in previous labs before implementation. We will be using 3D software to find out implementation problems before writing any code.

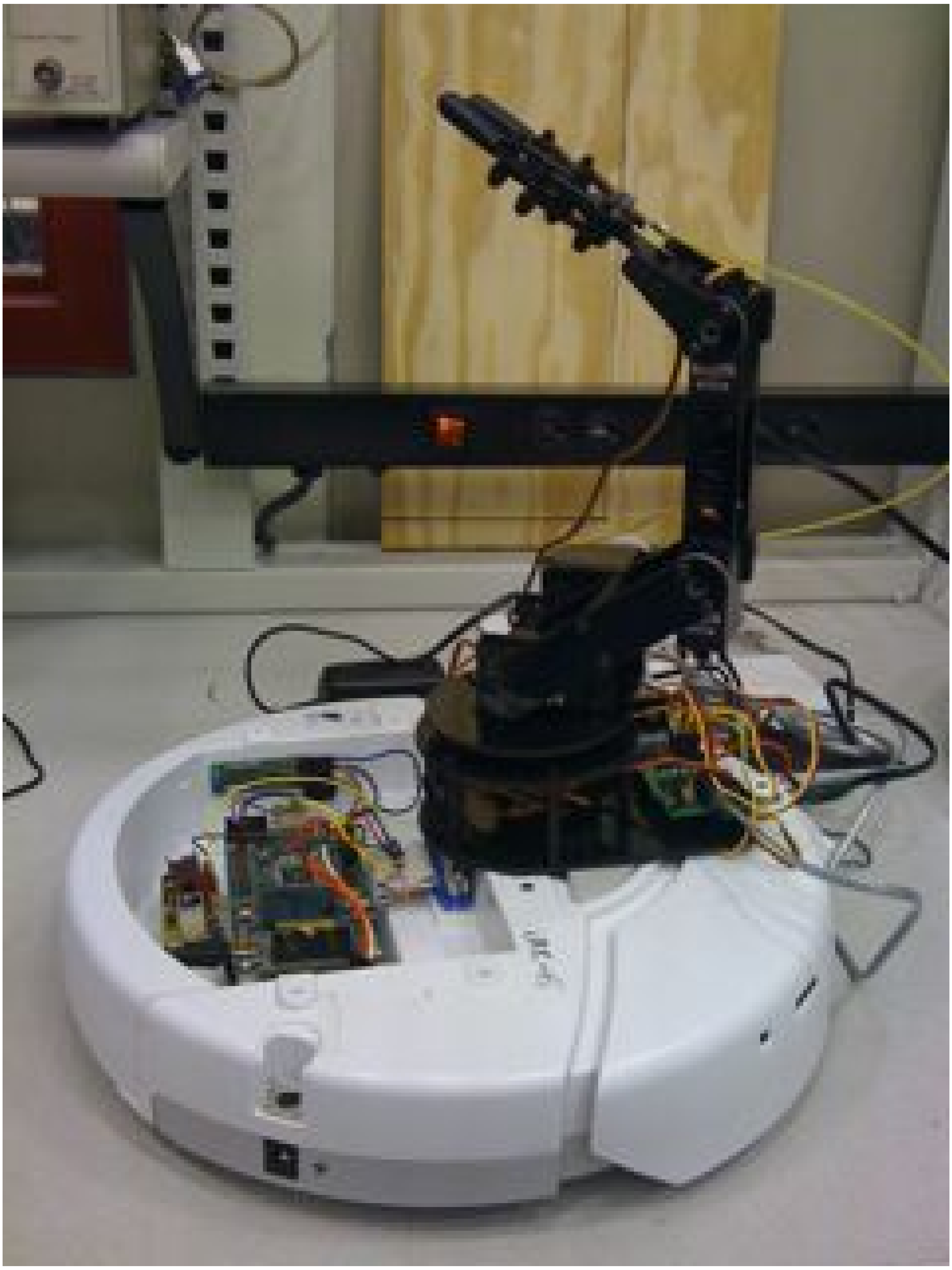


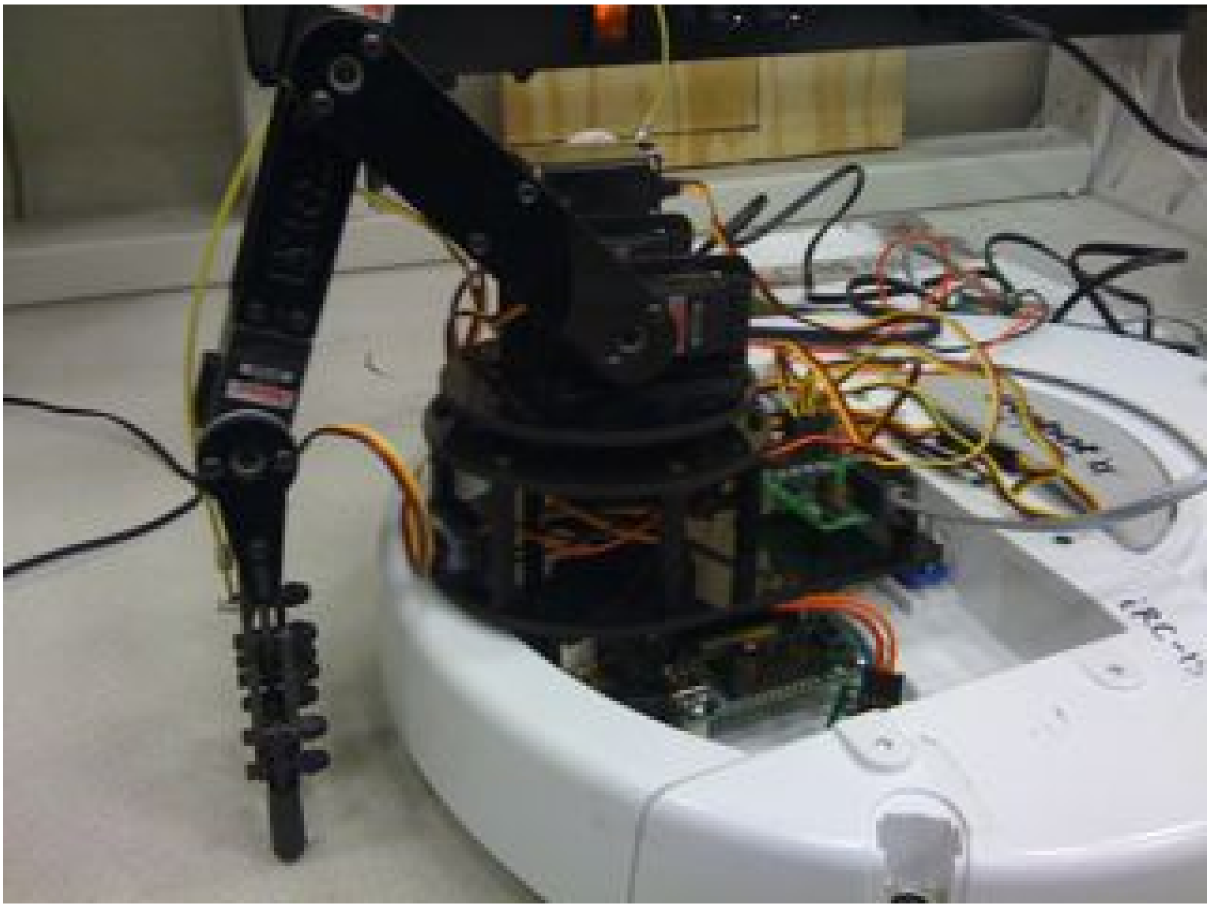
Implementation

The method of holding the chalk is a tricky one. In order to draw discontinuous lines we need a way to lift the chalk up off the drawing surface. While drawing we need to continuously be applying pressure to the chalk so that as the chalk wears way it is still in contact with the ground.

For this we will attempt to use a LynxMotion 5-axis robot arm equipped with a special attachment for the chalk. This robot arm will be mounted to the top of the iRobot create and controlled through the luminary board's serial port.

Below are some mockups of the robot arm placed on top of the iRobot.





We can communicate with the arm via serial, and send commands in the conventional way. Developing a protocol for the drawing mechanism will be a major milestone. We have to develop the means of converting an SVG file to a series of commands to send over a network and control the iRobot and the mounted robot arm. Below is a snippet from an SVG file:

```
<g
   id="g2840"
   transform="translate(-2210,-84.7)"
   style="fill:#ffffff;stroke:#000000">
  <path
     d="m 309,111 c 0,1 -2,5 -6,11 -3,6 -5,9 -5,11 0,3 2,8 5,14 3,6 4,11 4,14 0,8 -4,15 -14,21 -9,5 -18,8 -27,8 -11,0 -26,-5 -45,-16 -19,-10 -33,-
     id="path2842"
     style="fill:#ffffff"
     sodipodi:nodetypes="ccsssscsccccccccccccscscssccccscscscscscccccscccsscscccccscc" />
  <path
     d="m 284,127 c -10,-5 -22,-7 -34,-7 -21,0 -42,7 -63,22 -5,4 -5,8 0,5 22,-11 44,-17 67,-17 12,0 21,2 27,5 9,3 8,-4 3,-8 z"
     id="path2844"
     style="fill:#ffffff"
     sodipodi:nodetypes="csccscc" />
  <path
     d="m 170,34.8 c 3,-4.6 2,-11.3 -6,-5.9 -13,13.4 -20,31.9 -20,55.3 0,12.1 2,25.8 8,39.8 1,1 3,8 4,-1 -2,-11 -3,-23 -3,-35.6 0,-23.2 6,-40.7 17
     id="path2846"
     style="fill:#ffffff"
     sodipodi:nodetypes="ccsccsc" />
  <path
     d="m 123,150 c -21,-10 -40.3,-14 -56.6,-14 -5.3,0 -10.8,0 -16.1,0 -7.6,1 -14.8,2 -23.8,8 -2.7,2 -1.2,8 4.8,7 8,-3 19.7,-7 37.1,-7 15.8,0 33.6
     id="path2848"
     style="fill:#ffffff"
     sodipodi:nodetypes="cccccscc" />
  <path
     d="m 149,188 c 2,-2 2,-8 -5,-4 -23,16 -34,43 -34,78 0,4 0,7 2,10 2,3 5,4 6,-2 0,-42 12,-68 31,-82 z"
     id="path2850"
     style="fill:#ffffff"
     sodipodi:nodetypes="ccsccc" />
</g>
```

We most likely will support a subset of the commands that are in the XML specification for the SVG file format in order to keep the proper scope of the project.

As far as the methods for drawing, we have discussed a few methods for steering the robot based on the SVG data. The first is using a "turtle method", which is based on how printers work. This is analogous to parsing a PostScript file: pretty much a discrete set of move up, down, left, right, and so on, these are commands much like what we programmed in the last few labs with the iRobot.

A second method we have discussed involves sending a sampled version of the data to the robot, and then interpolating the data using *de Casteljau's algorithm.* This is an algorithm for constructing bezier curves, which is useful for reconstructing arbitrary lines or curves based on control points.