Date: 03/19/02
Student Name: Clerc Jean-Philippe
TAs: Aamir Qaiyumi
Uriel Rodriguez
Instructors: Dr. A. A Arroyo
Dr. Schwartz

University of Florida
Department of Electrical and Computer Engineering
Intelligent Machines Design Laboratory
EEL 5666

Autonomous Bottle Opener Robot

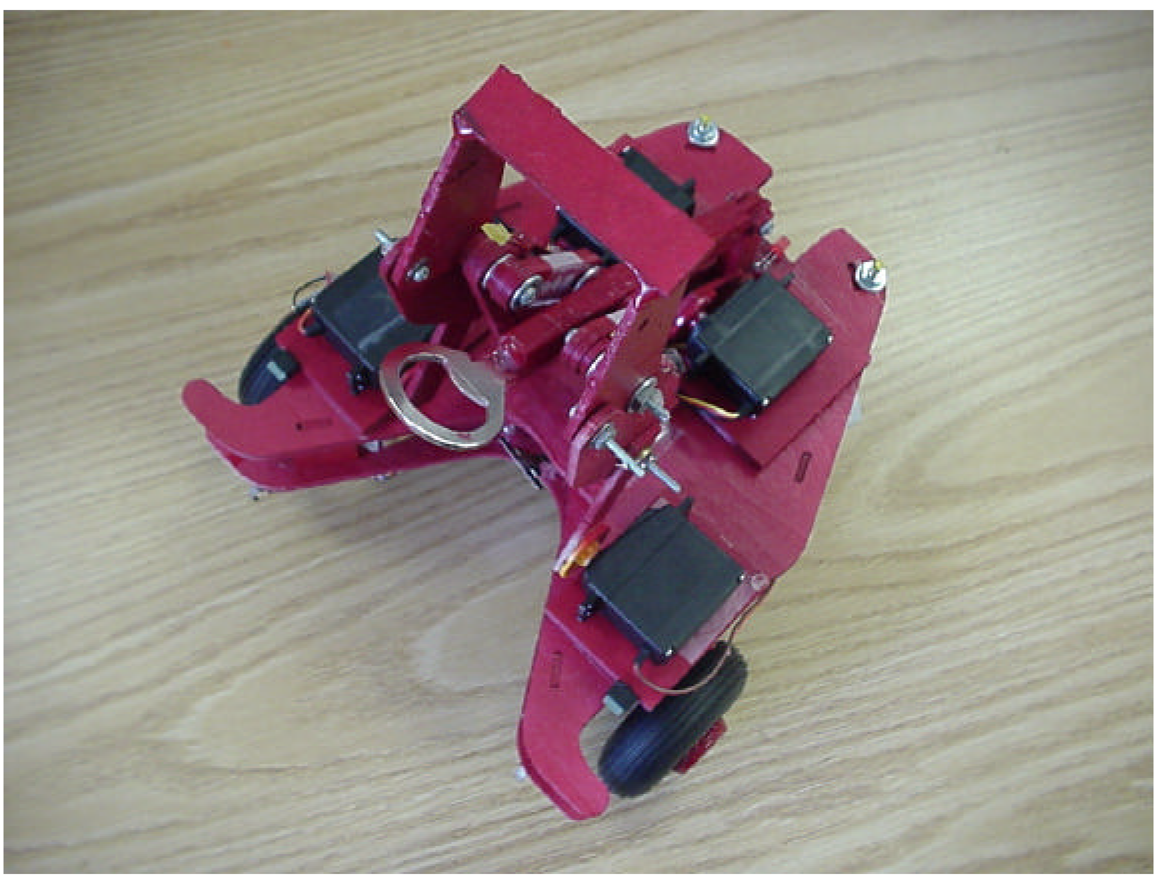


Table of Contents

Abstract

The Autonomous Bottle Opener Robot (ABOR) was designed to navigate on top of a bar and to assist the barman at his task. When the robot detects a beer bottle it uses a five bar mechanism in order to open the beer. Prior to the opening of the bottle the robot makes a 90 degree rotation around the bottle and move it toward the customer. After opening the bottle the robot backup and follow its path looking for another bottle.

Executive Summary

ABOR was an autonomous robot designed to open beer bottles and serve them to the customers. The robot was made using an estimated 150 parts. The main body of the robot was composed of two platforms each made out of wood.

In order to interact with the surrounding and to be able to locate itself, the robot uses three limit switches and three CdS cells. Two limit switches were used on the bottom of the first platform, when triggered they send a signal to the mother board ordering the robot to do a 180 degree rotation. The third limit switch was used to detect a beer; whenever it was triggered the robot went into opening mode.

The three CdS cells were used in order to do line following, depending on the color of the surface the CdS returned different voltages.

A set of five LEDs was used in order to provide feedback to anyone watching the robot in action. For example two LEDs were on when the robot was going forward, two of them were also on when the robot was opening a beer, and finally one LED was on when the robot was moving backward and turning.

The robot used a pair of mechanism in order to be able to open beers. The first one was a four bar mechanism and had for sole purpose to secure the beer while the robot was in opening mode. The second mechanism was used to open the beer and was a five bar mechanism.

The robot was using a MTJPRO11 board from Mekatronics, this board acted as the brain of the robot. The board received all signals and commanded the robot to react to those signals, it also provided power to the different components.

The board was powered by a 7.2V battery pack from RadioShack.

A light, powered separately from the rest of the components, was used to provide light so that the robot did not depend too much on the surrounding light.

Introduction

In order to help a barman in a Friday night bar overcrowded with customer trying to quench their thirst, a robot was designed to assist the barman in opening bottles.
The robot navigated on top of a bar and opened beer bottles that it encountered on its path. The name of the robot was ABOR (Autonomous Bottle Opener Robot). This report describes the design, functioning and testing procedure for the robot as well as the C programming used for driving the robot.

Integrated System

The robot integrated electronic components and mechanical components in order to perform a specific task in an autonomous way. The mechanical part consisted of two servo motors used for moving the robot. Two additional servos were used for the actuation of the opening mechanism and one was used for securing the beer during the opening operation. The opening mechanism was a five bar mechanism. This mechanism was designed to mimic the motion of someone opening a beer. All these components were controlled by a MTJPRO11 board from Mekatronics. In order to interact with the surrounding the robot was equipped with several sensors completing the electrical package of the robot, see Figure 1.

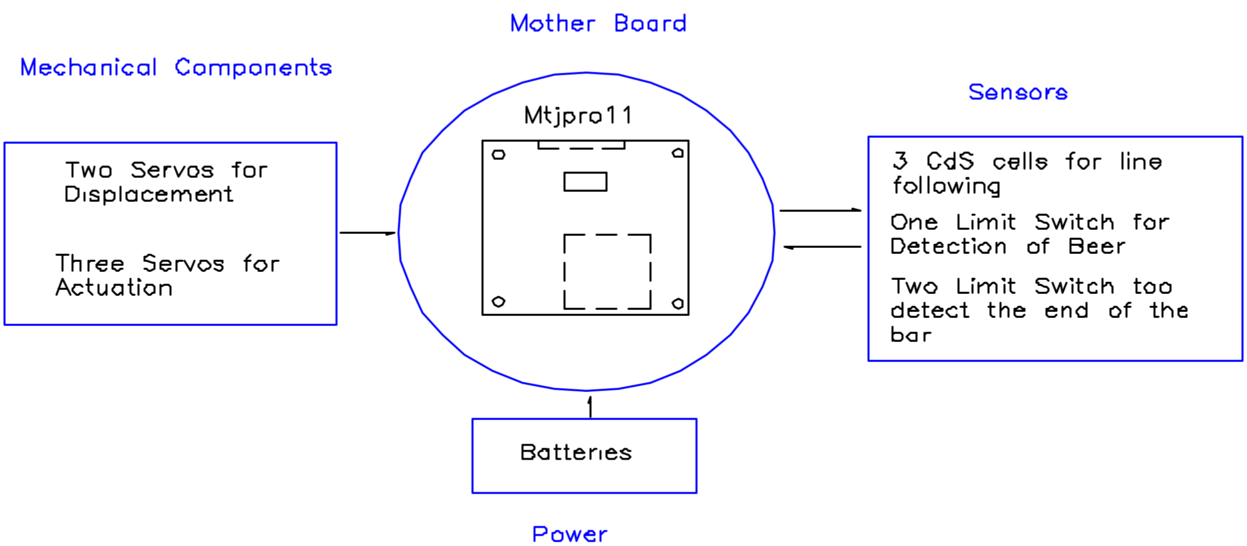


Figure 1, Flow Chart.

Figure 2 and 3 showed the different servo used for the robot as well as the mother board.

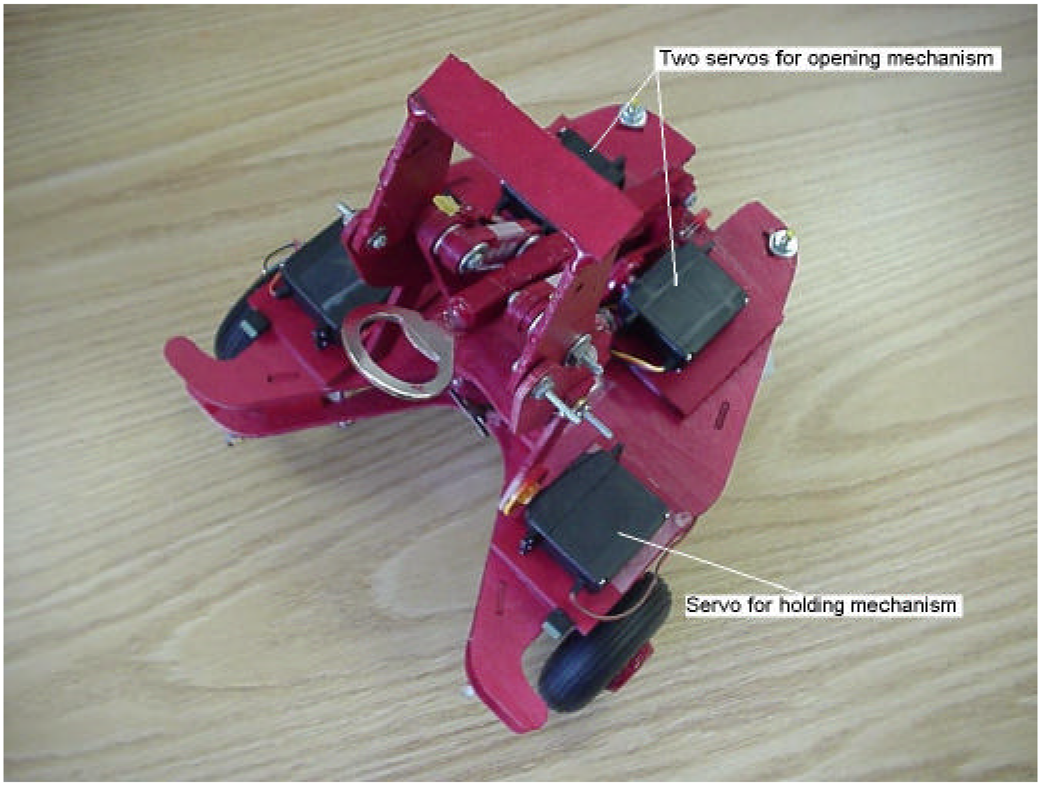


Figure 2, Servo Motors.

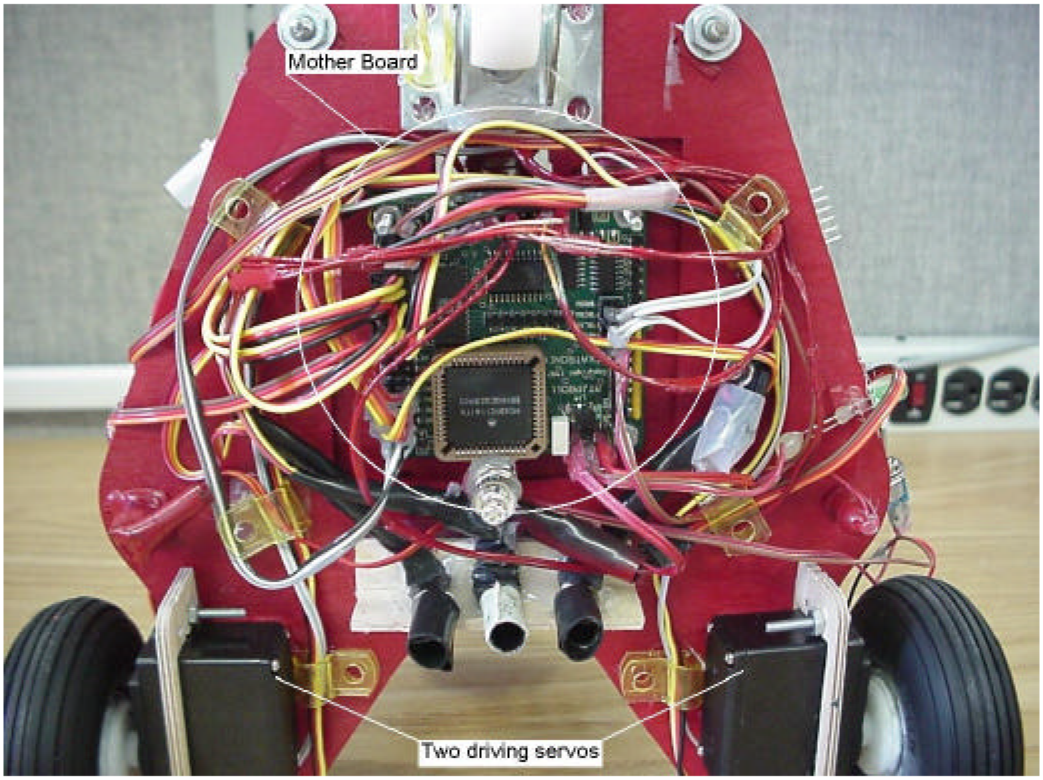


Figure 3, Mother Board.

Mobile Platform

First the concept was drawn in 3D in order to visualize the layout of the robot and check for interferences. The first 3D drawing was done to bring the concept to life with approximated dimension. From this drawing some essential features appeared to be necessary for the good functioning of the robot. Some of those features were: the opening with a shape that matches the contour of a beer, and a securing location were the beer axis intersect with the center axis of the wheel (allowing the robot to rotate around the beer). After this drawing was completed some actual components were bought and the second concept was modified accordingly. All components were actually designed in 3D to their actual specifications, see Figure 4.

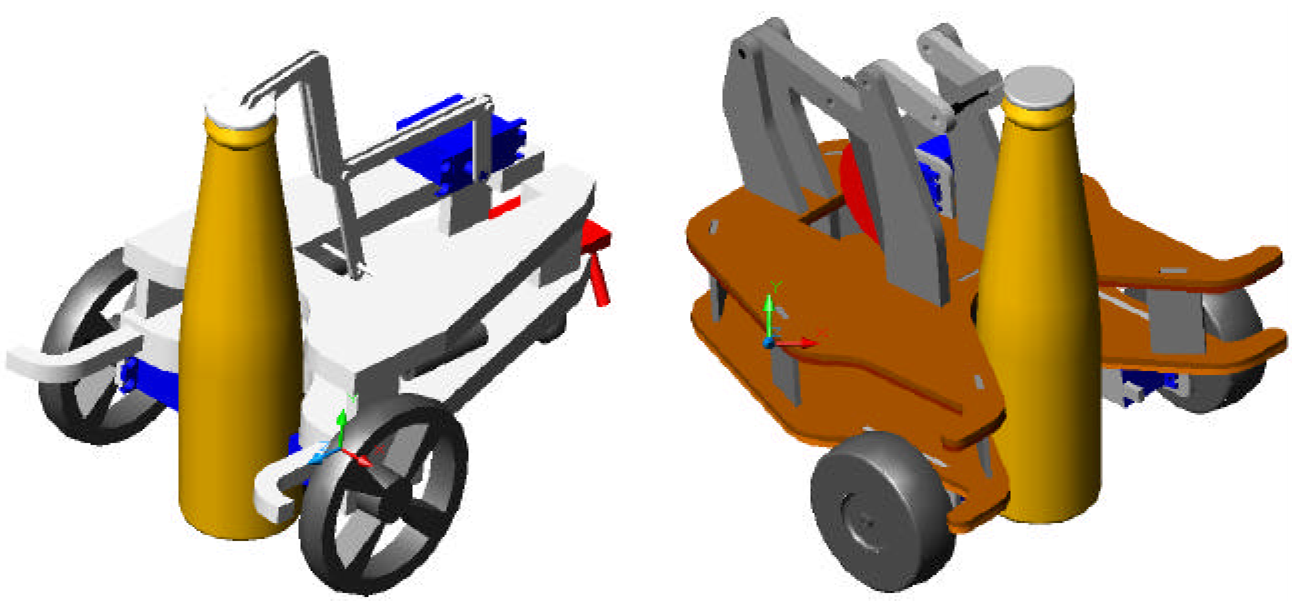


Figure 4, First and Second Concept.

Has the robot evolved the 3D model also evolved and was used to scale the link for the mechanism, do the geometric calculations to determine pivot point and finally to actually cut the parts out of wood using a CNC machine. The final 3d drawing can be seen in Figure 5.

Figure 5, Final 3D Drawing.

The platform serves as a mobile frame for the opening mechanism. See Figure 6 for the main frame.

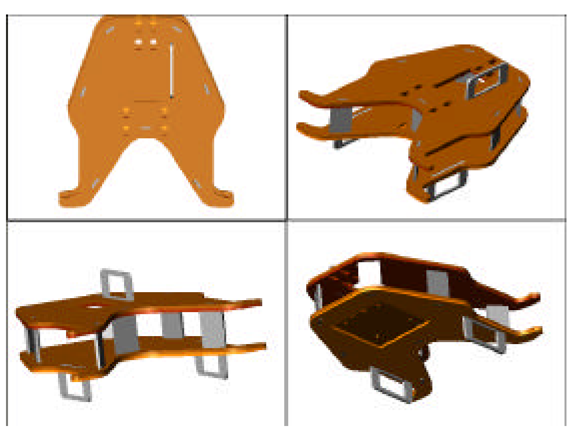


Figure 6, Main Frame.

Actuation

The robot was using two 44oz/in hacked servo motors in order to move along the bar, see Figure 7.

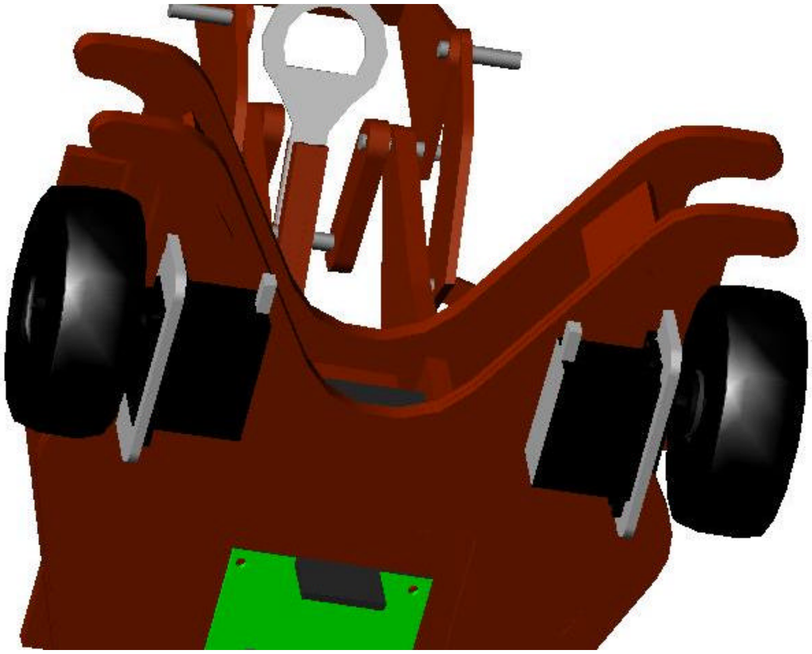


Figure 7, Servo Motors Used For Displacement.

The five bar mechanism was also driven by two servo motor, see Figure 8.

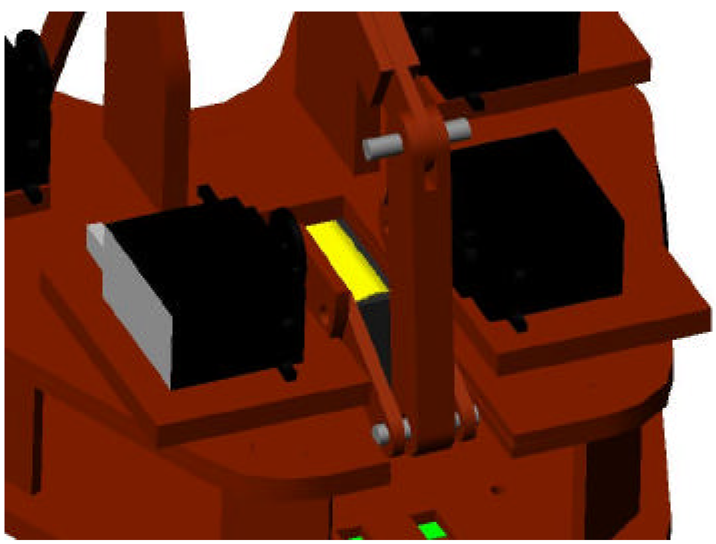


Figure 8, Servo Motors For The Mechanism.

Opening Mechanism

The first step was to design the mechanism and define what kind of motion could be used. In order to do so, two software were used:

- Fourbar (Figure 9).
- Working Model (Figure 10).

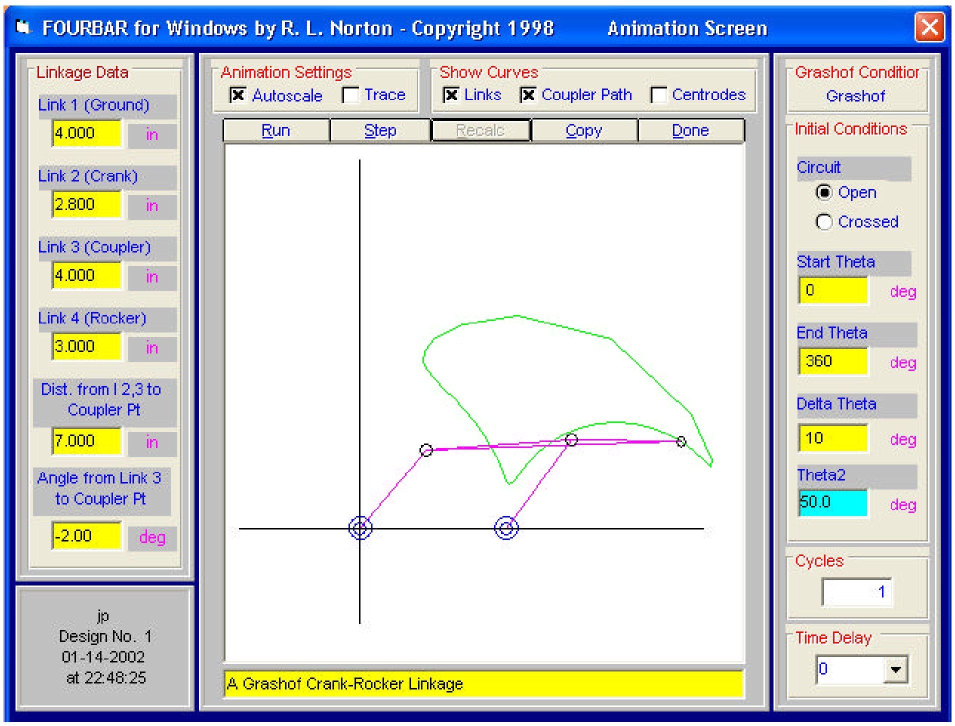


Figure 9, Fourbar.

Fourbar allowed to enter the length of the different links and to get a coupler curve (the green curve in Figure 9). Than by modifying in real time those dimensions it was possible to get a range of curve and pick which one would provide the proper motion. Those linkages were than drawn in Working Model, where the actual physical motion could be simulated.

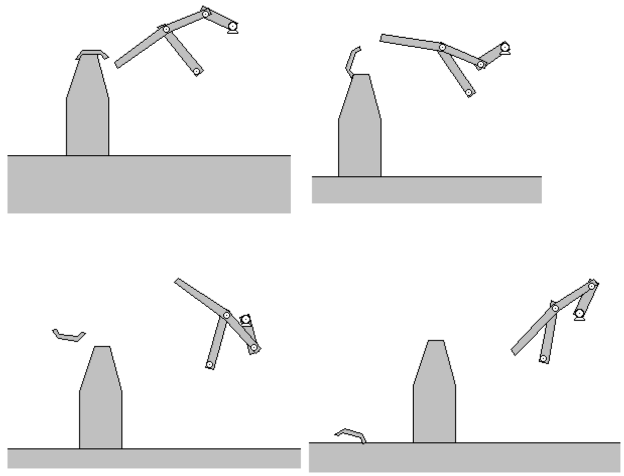


Figure 10, Working Model.

From those simulations it was decided that a four bar mechanism would be the best mechanism to use. Some forces lost due to friction could be compensated by mechanical advantages. With this in mind, a hand drawing was made using the desired position of the mechanism, namely the starting line and ending line of the opening device, see Figure 11. These two positions were obtained from opening a beer manually and mimicking this operation.

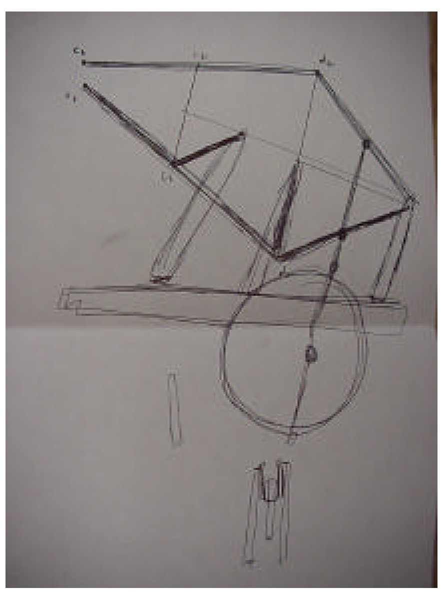


Figure 11, First Mechanism.

From this drawing realized to scale, a cad drawing was derived in order to cut the necessary parts required to build a prototype, see Figure 12.

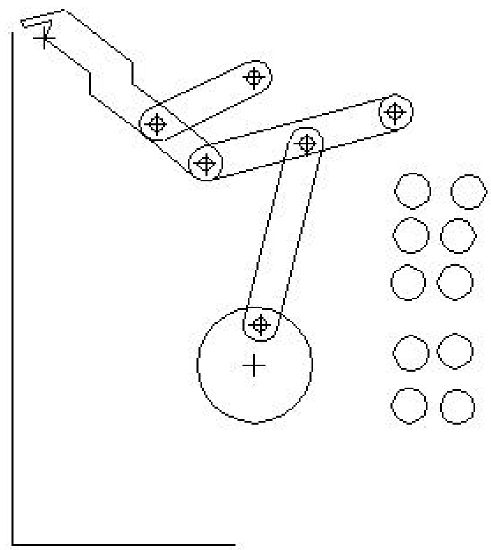


Figure 12, CAD Drawing.

Using these parts a prototype was created. This prototype was a fourbar mechanism. It was composed of two rockers (2 and 3), and a coupler (1), the four linkages were numbered (ground was considered has one linkage), see Figure 13.

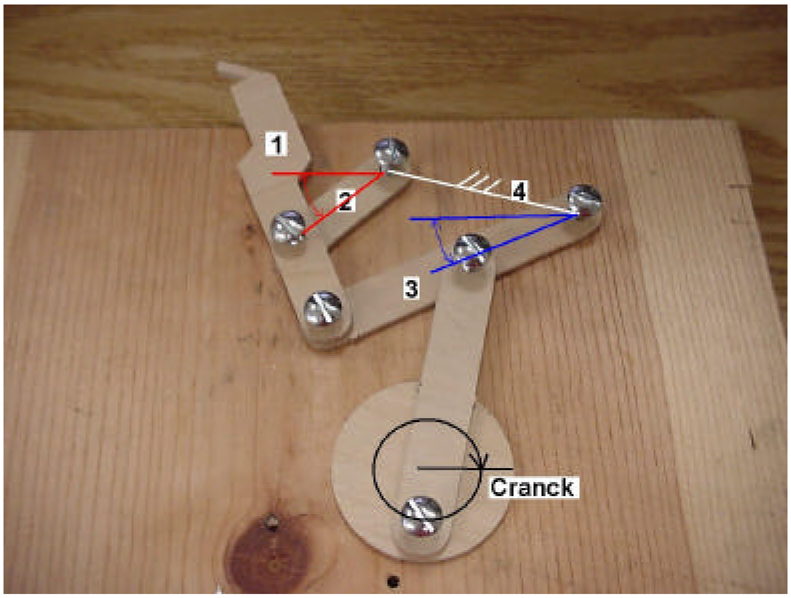


Figure 13, Prototype.

Taking into account the results of the prototype a to-scale hand drawing was derived and the layout of the different linkage was drawn, see Figure 14.

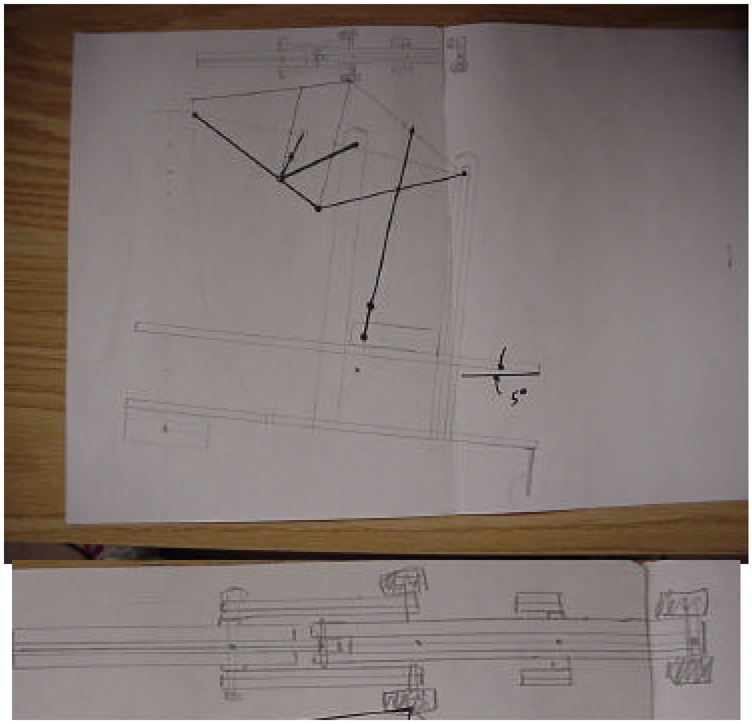


Figure 14, To Scale Drawing and Layout of the Mechanism.

An AutoCAD drawings was than derived in order to prototype and build the mechanism, see Figure 15.

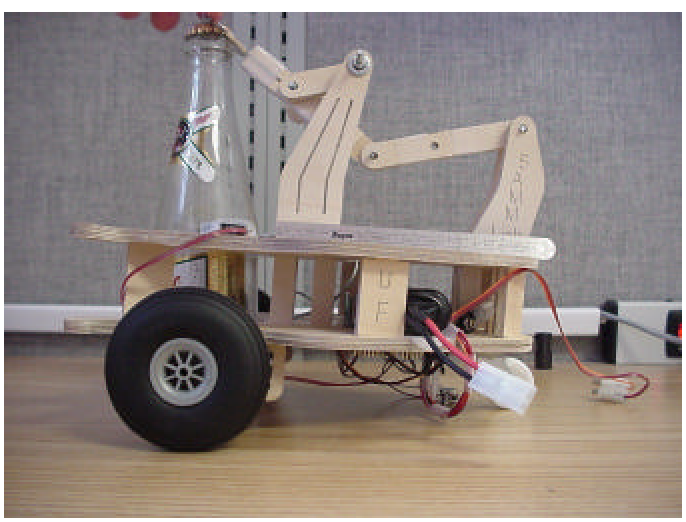


Figure 15, Mechanism on the Platform.

Some testing was done in order to determine if the mechanism would work, but a toggle point was present in the design rendering it unusable. In this case the toggle point would either bring the mechanism to collapse and/or required an added force to open the beer. The next step was to determine the required force in order to open the beer. A force gage was used in order to do so, see Figure 16.

Figure 16, Force Gage.

A beer was open and a force of about six pound at a two inches distance was found giving a torque of 12lbs in. The servo-motors were also tested experimentally to get their forces; the results were in agreement with rated force provided by the manufacturer of the motors, see Figure 17.

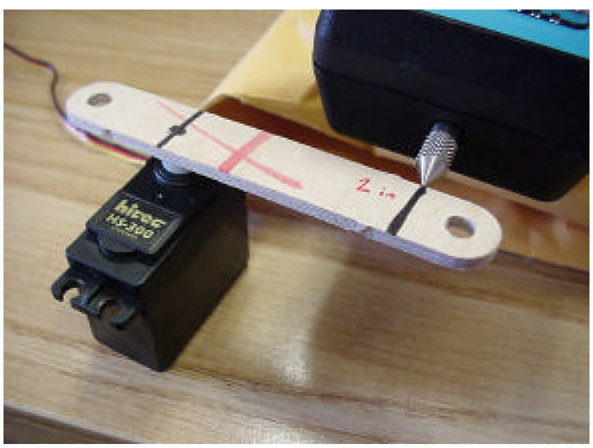


Figure 17, Experimental Setup.

A force of 1.47lbs at 2 in was found giving a torque of 2.94lbs in. At this point it was obvious that one motor would not be enough to open the beer. It was decided to use two motors in parallel and an arm of 4in. This increased the torque to 11.76 lbs in (almost the required torque to open the beer). The following mechanism was obtained, see Figure 18.

Figure 18, Mechanism With Small Servo.

The testing was done using two small servos 42oz/in. This showed that the servo were not powerful enough. It was decided to buy two 75oz/in motor. The torque would be enough to open the beer.

In order to get a working mechanism, AutoCAD was used, instead of the hand derivation, to derive the links, joints and length for the mechanism this added more precision. From those drawing the parts were cut and mechanism tested.

Range of motion of the mechanism was shown in Figure 19. Also shown were the starting position (white) and opening position (blue).

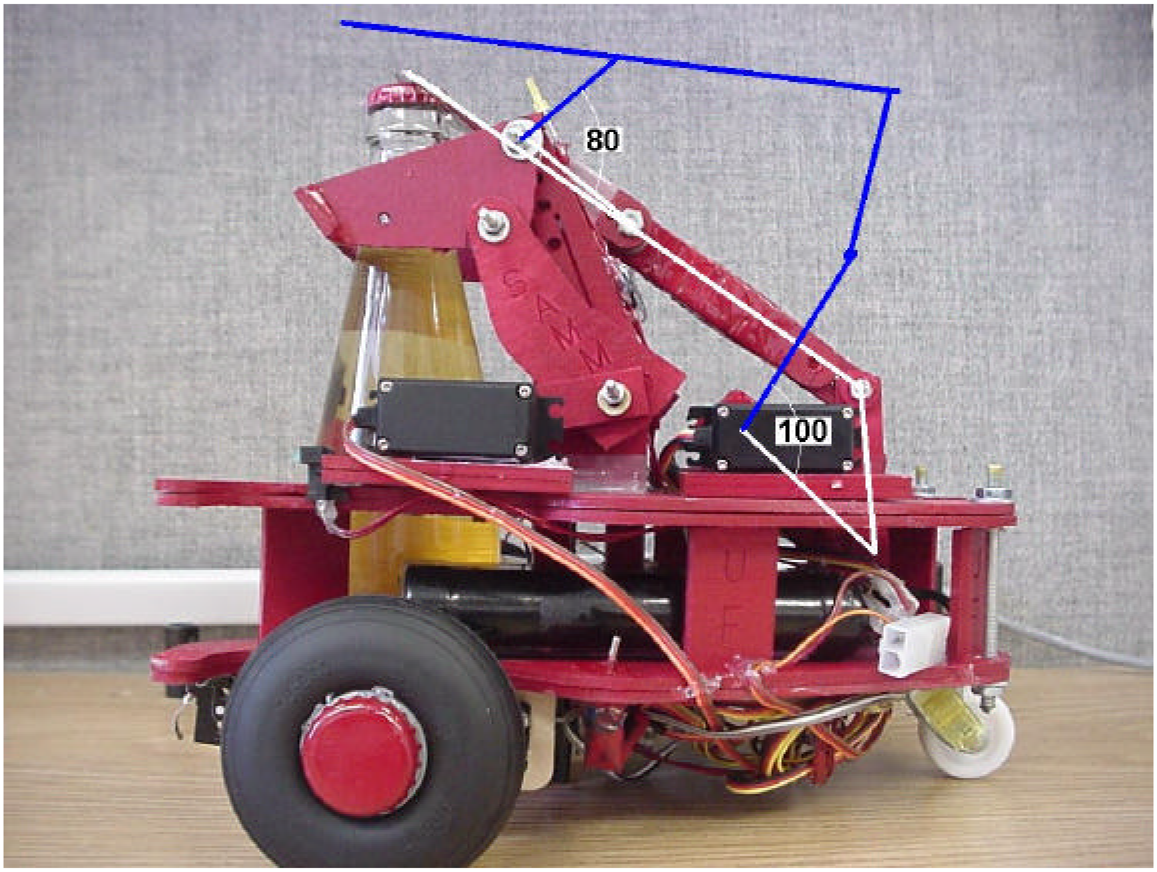


Figure 19, Angle and Position.

The final mechanism 3D Drawing, was shown in Figure 20.

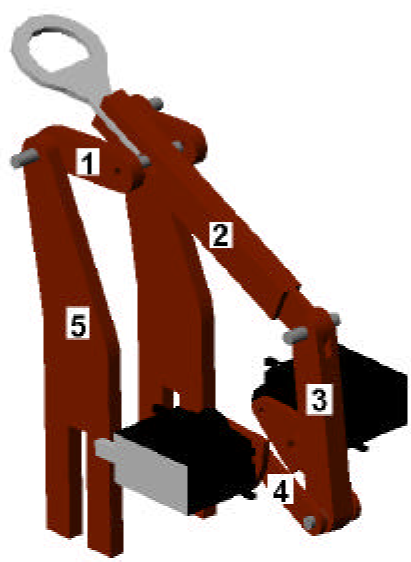


Figure 20, Final Mechanism.

Holding Mechanism

In order to properly secure the beer during the opening operation a holding mechanism was designed. It was decided to use a four bar mechanism to avoid having the servomotors to high on the robot. This fourbar mechanism allowed the servo to be mounted on the second platform, see Figure 21.

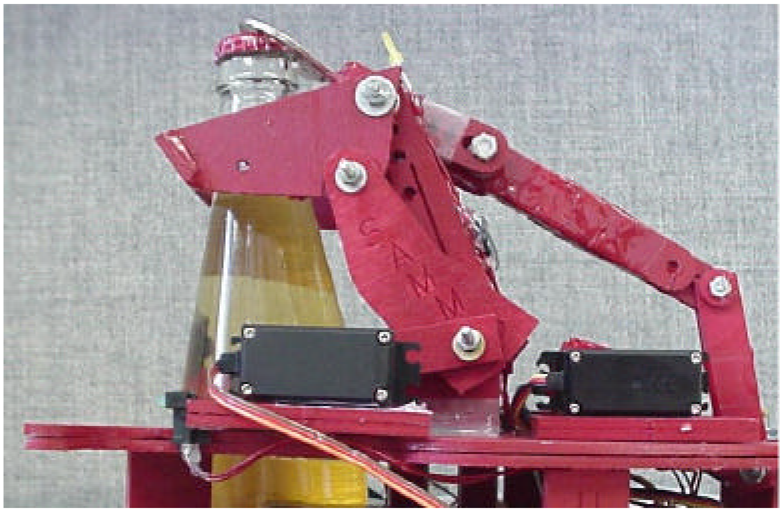


Figure 21, Holding Mechanism.

Figure 22 showed the different linkage used for the mechanism.

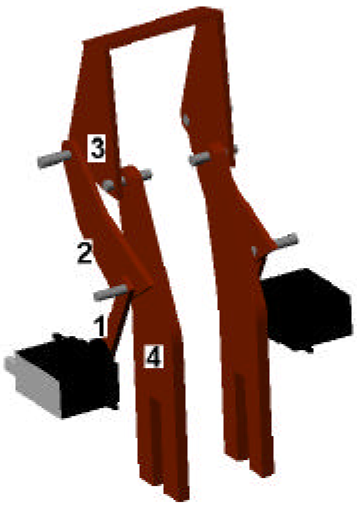


Figure 22, Holding Mechanism.

The operating range of the mechanism was shown in Figure 23, and was 50degree input and 100 degree output.

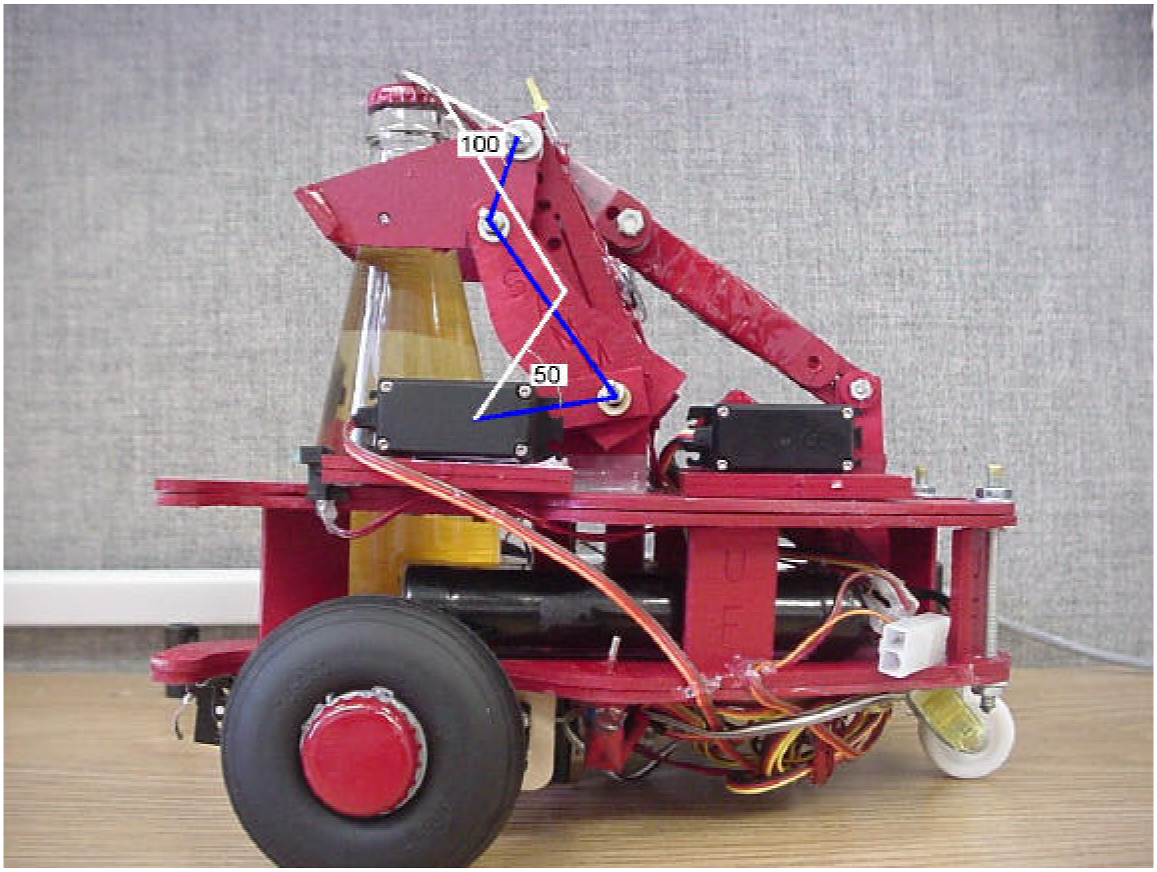


Figure 23, Holding Mechanism Operating Range.

Sensors

The sensors were important because they allowed the robot to interact with the surrounding and respond to that surrounding according to the program. ABOR followed a line on top of the bar. In order to do so it was using three CdS cells and the difference in voltage (related to the difference in color) to follow a line, see Figure 24.

It also required limit switches to detect the beer and detect when the bottle was ready to be open, see Figure 25. An extra limit switch was required to define which way the robot was going (in order to always bring the beer toward the customer) and to know when to turn around. This was done by placing an edge at the end of the bar that triggered that limit switch and order the robot to do a 180 degree rotation.

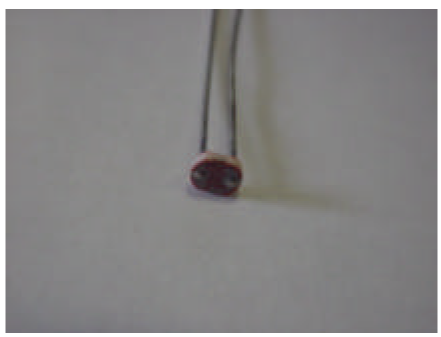


Figure 24, LcD Sensor.

Figure 25, Limit Switches

Behaviors

The robot followed a line on the bar and whenever the barman needed a beer bottle to be open he placed the bottle on the line. The robot detected the beer and brought it toward the customer. The robot was designed so that the center axis of the beer matches the center line between the two driving servos of the robot. This allowed the robot to rotate on itself and around the beer for an angle of 90 degree away from the customer. At that point both servos were turned on and the bottle was pushed toward the customer. The bottle was opened. Finally the robot moved back and followed the line again until it found another bottle, see Figure 26. If an object was placed on the track the robot would move it in the same manner in order to clear its path.

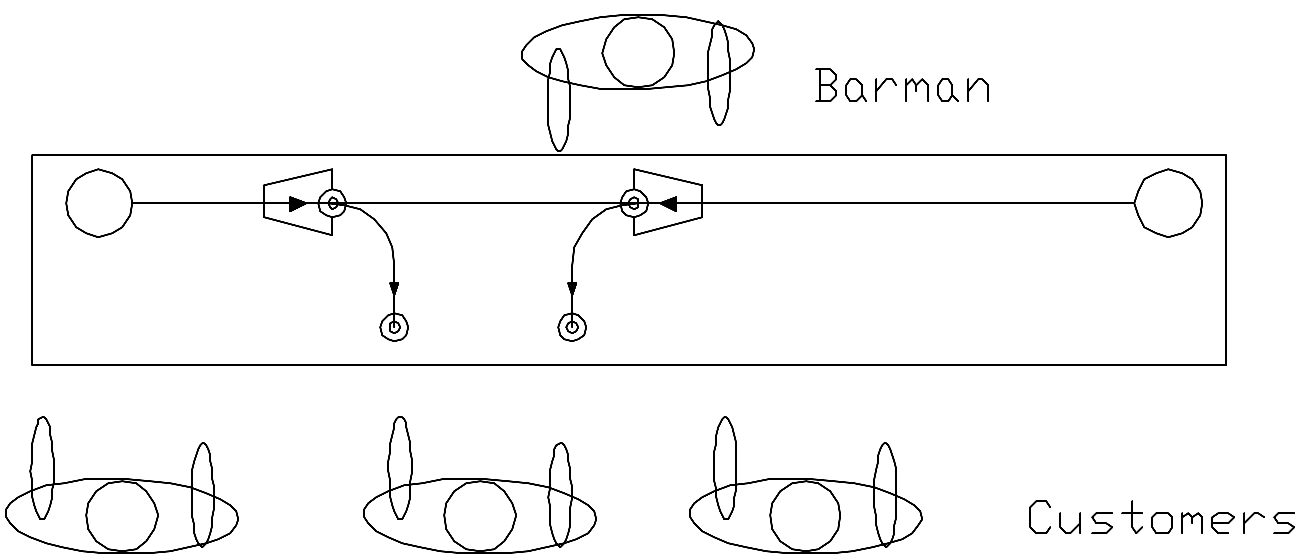


Figure 26, Robot Behavior.

The robot used also a set of five LEDs for feedback, see Figure 27.

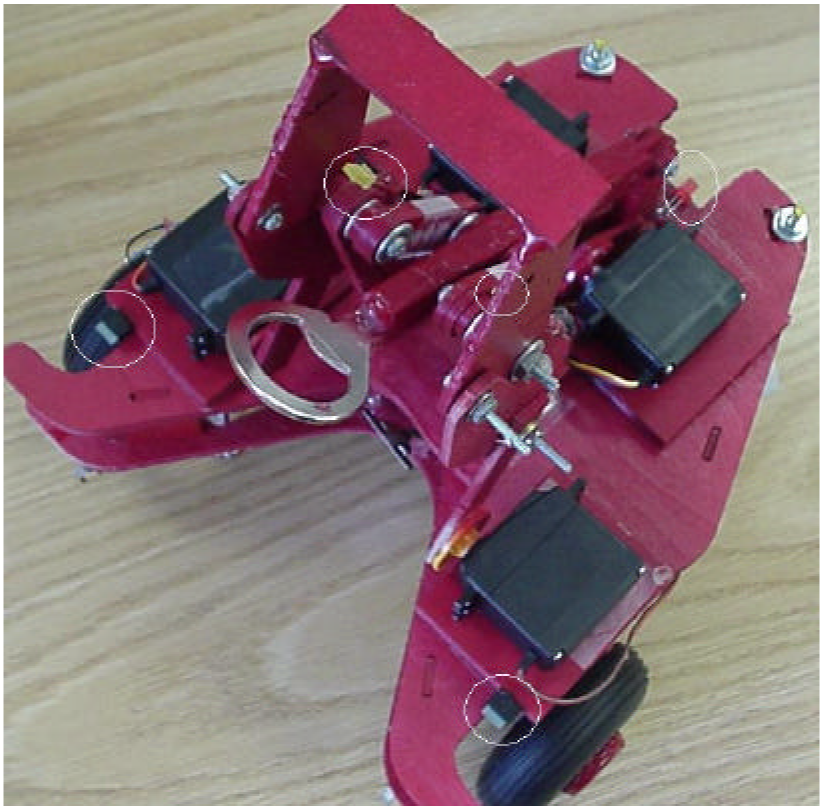


Figure 27, LEDs used For Feedback

The two green LEDs in the front were on when the robot moved forward. The two orange LEDs were on when the robot was in an opening mode and the red one in the back was on when the robots moved backward and turned around.

Experimental Layout and Results

In order to classify the best beer opener device several criteria were considered, especially the torque required to open the beer, the acceptance angle to secure the beer and angle to accept the beer (Figure 28).

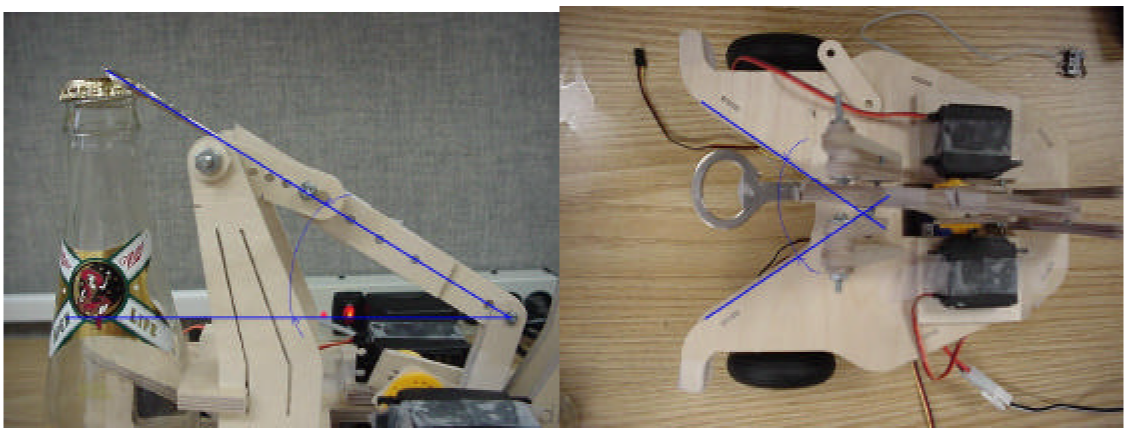


Figure 28, Beer Opener Acceptance Angle and Maximum Angle to Accept the Beer.

The first device was a cheap $3 opener. It was dismantled and left only with the metal part seen in Figure 29. One advantage was the circular feature allowed the beer to be accepted no matter at what angle the beer came in as well as a low securing angle, but it required a torque of 12 lbs in to open a beer (the highest measured).

Figure 29, First Beer Opener Tested.

The next beer opener was the most expensive one. At a price of $9.99 a high expectation was placed on that opener. Unfortunately it seemed that most of the price of that opener was coming from a heavy metallic frame and fancy grip, see Figure 30.

Figure 30, Second Beer Opener.

After dismantling the second beer opener, all that was left was a simple but original opener (See Figure 31). This device was attached to the rocker of the mechanism and was tested.

Figure 31, Second Beer Opener Device.

This device would secure properly all the time but require a high acceptance angle and a higher angle to completely open the beer, as well as the beer required to come in straight. This increased the angle of the rocker device and did not work properly. It would lift up the cap without completely removing it. The required torque was similar to the first device. In order to accommodate for this device, the changes would be consequent, so without any real advantages the device was abandoned.

The third device was from the same brand as the second one but half the price $5.99, but had two ways to open the beer, see Figure 32.

Figure 32, Third Beer Opener.

The third beer opener was similar to the first one except that it required a larger acceptance angle for the beer; it could also accept beers from any angle. The can piercer device (the second option) could open the beer but it required a higher torque due to the fact that the pierce point was past the center of the cap.

The last device considered ($4.99) was a circular opener device that used the twist off capability of most of the beer bottle allowing to open the beer with your hand. After a success rate for opening the beer of one out of 22 this device was considered requiring a complete redesign of the mechanism. After doing some testing the torque required was estimated to be about 8 lbs in. This idea was quickly abandoned due to the fact that it required also an 8 lbs force to prevent the beer from rotating, see Figure 33.

Figure 33, Third Beer Opener.

In order to obtain a more efficient opener, it was decided to create a new device by combining two previously bought, Figure 34.

Figure 34, Two Devices Used for the New Device.

The gage force showed a torque of 9 lbs in. But the new device was so sharp that instead of opening the beer it would pierce the cap, see Figure 35.

Figure 35, Cap After being Open By The New Device.

In order to prevent this, the tool was modified, with a hammer, see Figure 36.

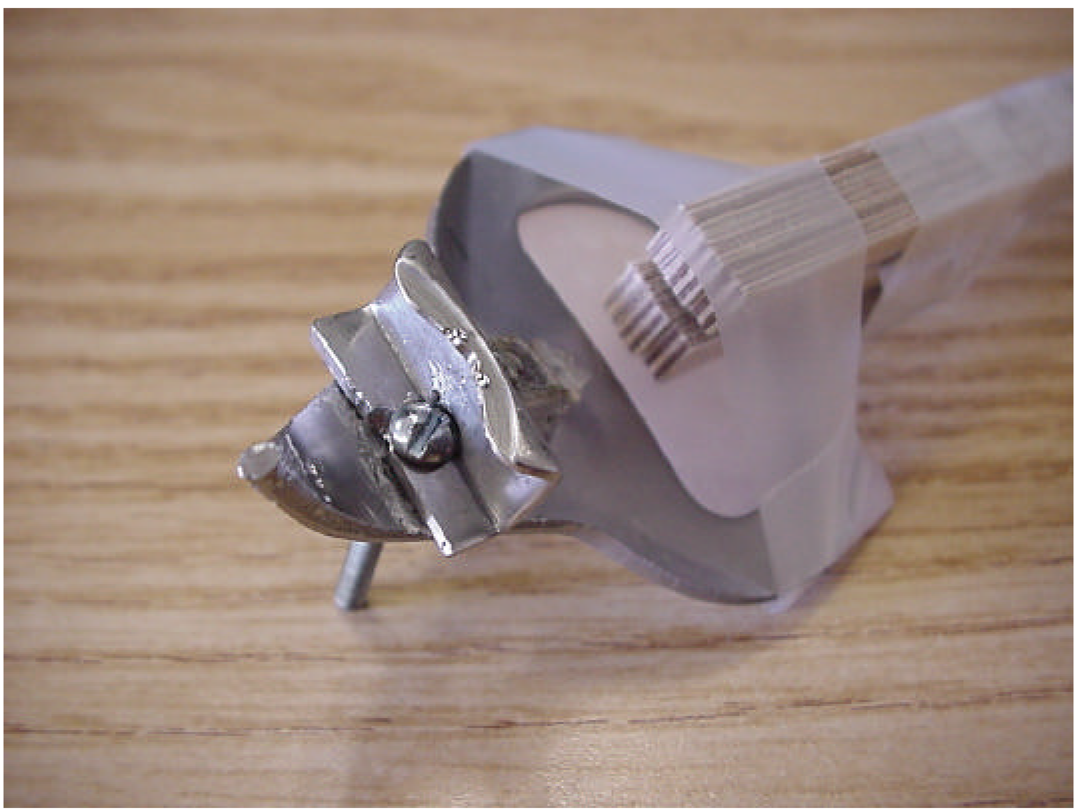


Figure 37, Cap After Being Modify.

After the modification this new device would work but required a really accurate positioning of the beer and would not open all the time.

After all this time and money spent it was decided to go back to the first device and to modify it in order to make it work by increasing the length of the rocker, see Figure 38.

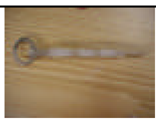
Figure 38, Final Device and Rocker.

After attempting to open 32 beer only 4 where open completely, all with device one, 3 with device one modified. Below are all cap attempted be open, see Figure 39.

Figure 39, All Cap Used During the Mechanism Testing.

All results obtained during the testing of the devices were summarized in Table 1.

Table 1, Summary for all devices.

	Device	Acceptance Angle to Secure the Beer (smaller is better)	Acceptance Angle to accept the beer(larger is better)	Price $	Torque Lbs in	# of beer opened /32	Rating #1 being the best
	Device1	Small	Large	3	12	1	2
	Device2	Large	Small	9.99	11	0	5
	Device3	Large	Large	5.99	11	0	3
	Device4	Large	Small	4.99	8	0	6
	Device5	Large	Small	15.98	9	0	4
	Device1 Modified	Small	Large	3	11	3	1

Conclusion

After 134 beers the robot was opening beers successfully which was the main challenge due to the fact that the robot was dealing with a real product. The fact that the robot was dealing with real part was a challenge because the beer could not be modified in order to make the task easier. That is why so many beer bottles were used for calibration and testing purposes. The robot was not doing a good job for the line following but this was due to the fact that the CdS cells took some time to react and that part of the navigation depended on time. IMDL was a great experience.

Documentation

Special thanks to Aamir Qaiyumi and the IMDL team *for their help in designing and building this robot*.

Intelligent Machine Design Laboratory from Professor AA Arroyo.

Design of Machinery from Dr Norton.

TJ Pro Assembly Manual from Keith Doty.

Appendix A, MTJPROBOARD with rewiring.

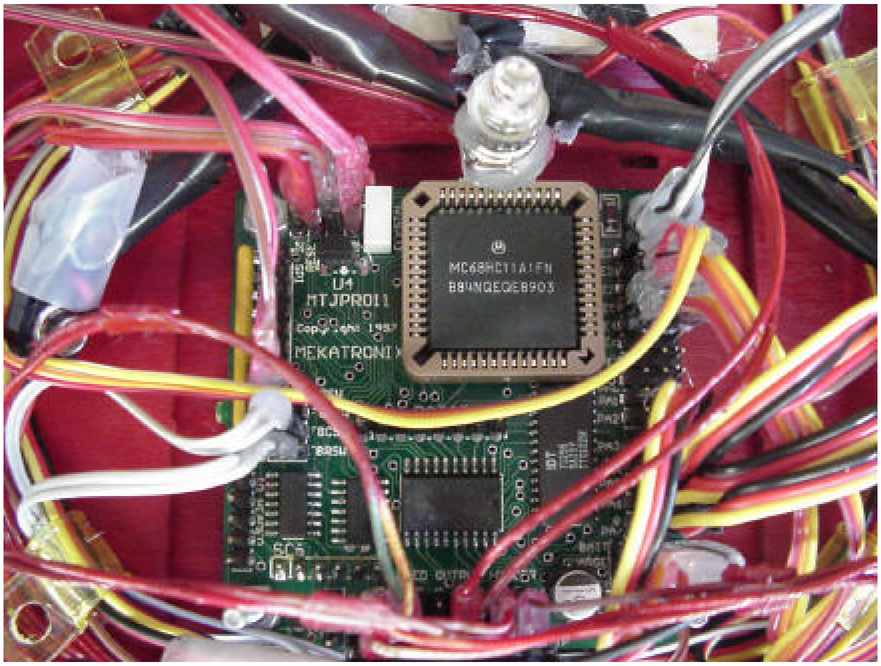


Figure A1, Mother Board with Rewiring.

After burning the board with a wrong connection the board was fixed using a wire connecting the power source to the power of the servos. See Figure A2

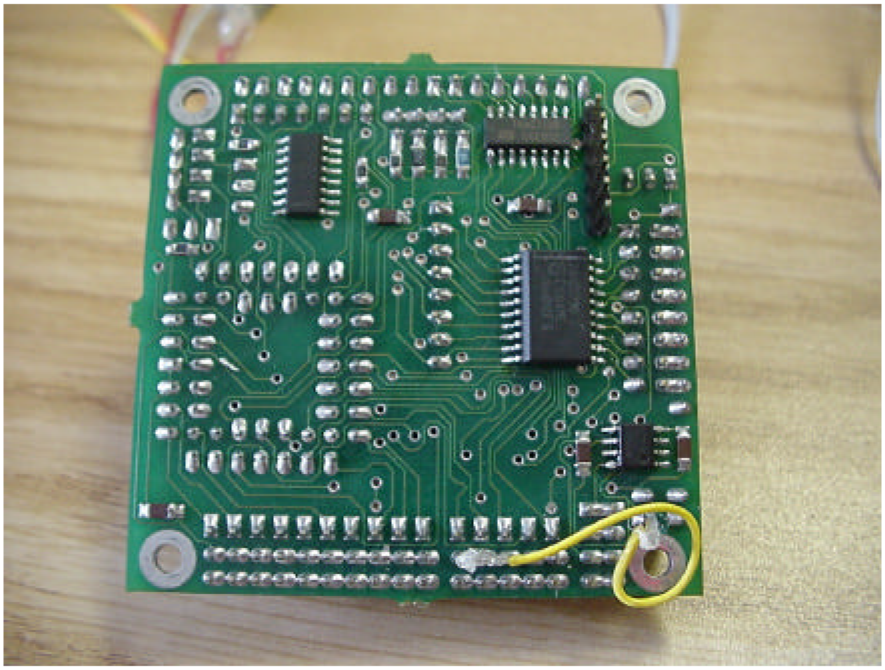


Figure A2, Mother Board Fixed With Yellow Wire.

Appendix B, Code Used For ABOR:

```
/************************************************************************
*
* *
* Title: ABOR Code
* Programmer: jp clerc  {A. Antonio Arroyo}*
* Date: 04 06 2002 *
* Version: 1.0 *
* *
* Better line following
************************************************************************
/
/*************************** Includes ********************************/
#include <analog.h>
#include <motortjp.h>
#include <clocktjp.h>

#include <stdio.h>
#include <tjpbase.h>
/************************ End of Includes ****************************/
/*************************** Constants ********************************/

#define LEFT_MOTOR 0
#define RIGHT_MOTOR 1
#define MAX_SPEED 100
#define ZERO_SPEED 0
#define BUMPER analog(0)
#define LEFT_CDS  analog(4)
#define CENTER_CDS  analog(5)
#define RIGHT_CDS  analog(7)

/************************ End of Constants ****************************/
/*************************** Prototypes *********************************/
void turn(void);
/************************ End of Prototypes *****************************/

void main(void)
/****************************** Main ***********************************/
{

init_analog();
init_motortjp();
init_clocktjp();
init_servotjp();
```

```
while(1)
{
  motorp(LEFT_MOTOR, -24);                          /* go straight */
  motorp(RIGHT_MOTOR, 56);
  wait(150);
  *(unsigned char *)  0x7000=0xC0;                  /* turn on both front lcd */

  if(LEFT_CDS > 190 )                        /* decelerate left accelerate right servos */
         {
         motorp(LEFT_MOTOR, -20);
      motorp(RIGHT_MOTOR, 50);
                  *(unsigned char *)  0x7000=0x40;                  /* turn off left front
lcd */
                  wait(100);
         }
  if(RIGHT_CDS > 230)                 /* decelerate right accelerate left servos */
         {
                  motorp(LEFT_MOTOR, -29);
                  motorp(RIGHT_MOTOR, 65);
         *(unsigned char *)  0x7000=0x80;                  /* turn on left front lcd */
         wait(100);
         }



  if((BUMPER>10)&&(BUMPER<30))                         /* turn 180degree left */
         {
                  *(unsigned char *)  0x7000=0x00;
                  *(unsigned char *)  0x7000=0x10;
                  motorp(LEFT_MOTOR, 0);
                  motorp(RIGHT_MOTOR, 0);
                  wait(300);
                  motorp(LEFT_MOTOR, -48);
                  motorp(RIGHT_MOTOR, 28);
                  wait(1500);
                  motorp(LEFT_MOTOR, 0);
                  motorp(RIGHT_MOTOR, 0);
                  wait(300);
                  motorp(LEFT_MOTOR, -49);
                  motorp(RIGHT_MOTOR, 61);
                  wait(3100);

                  *(unsigned char *)  0x7000=0x00;
         }
```

```
        if((BUMPER>30)&&(BUMPER<50))                    /* turn 180degree
right  */
        {
                *(unsigned char *)  0x7000=0x00;
                *(unsigned char *)  0x7000=0x10;
                motorp(LEFT_MOTOR, 0);
                motorp(RIGHT_MOTOR, 0);
                wait(300);
                motorp(LEFT_MOTOR, -48);
                motorp(RIGHT_MOTOR, 28);
                wait(1500);
                motorp(LEFT_MOTOR, 0);
                motorp(RIGHT_MOTOR, 0);
                wait(300);
                motorp(LEFT_MOTOR, -49);
                motorp(RIGHT_MOTOR, 61);
                wait(3100);

                *(unsigned char *)  0x7000=0x00;

        }


        if((BUMPER>120))                /* turn 90degree for opening and serves */

                {
                        *(unsigned char *)  0x7000=0x00;
                   *(unsigned char *)  0x7000=0x03;
                        motorp(LEFT_MOTOR, 0);
                        motorp(RIGHT_MOTOR, 0);
                        wait(300);

                        servo(0, 1400);
                        wait(400);



                        *(unsigned char *)  0x7000=0x00;
                        *(unsigned char *)  0x7000=0x01;
                        motorp(LEFT_MOTOR, -44);
                        motorp(RIGHT_MOTOR, 59);
                        wait(2700);
                        *(unsigned char *)  0x7000=0x00;
                   *(unsigned char *)  0x7000=0x02;
                        motorp(LEFT_MOTOR, 0);
                        motorp(RIGHT_MOTOR, 0);
```

```
wait(300);
*(unsigned char *)  0x7000=0x00;
*(unsigned char *)  0x7000=0x03;
motorp(LEFT_MOTOR, -22);
motorp(RIGHT_MOTOR, 55);
wait(2000);
*(unsigned char *)  0x7000=0x00;
motorp(LEFT_MOTOR, 0);
motorp(RIGHT_MOTOR, 0);
wait(300);
*(unsigned char *)  0x7000=0x03;

servo(1, 3600);
servo(2, 2650);
wait(2000);

*(unsigned char *)  0x7000=0x00;
servo(1, 1600);
servo(2, 4700);
wait(500);
servo(1, 0);
servo(2, 0);

servo(0, 2500);
wait(1000);
*(unsigned char *)  0x7000=0x03;

servo(0, 1400);
wait(400);
*(unsigned char *)  0x7000=0x00;
motorp(LEFT_MOTOR, -22);
motorp(RIGHT_MOTOR, 55);
wait(300);
*(unsigned char *)  0x7000=0x03;
motorp(LEFT_MOTOR, 0);
motorp(RIGHT_MOTOR, 0);

servo(1, 3550);
servo(2, 2700);
wait(2000);

servo(1, 1600);
servo(2, 4700);
wait(500);
*(unsigned char *)  0x7000=0x00;
servo(1, 0);
```

```
			servo(2, 0);

			servo(0, 2500);
			wait(400);
			servo(0,0);
			*(unsigned char *)  0x7000=0x03;
			motorp(LEFT_MOTOR, -48);
			motorp(RIGHT_MOTOR, 28);
			wait(1000);
			motorp(LEFT_MOTOR, 0);
			motorp(RIGHT_MOTOR, 0);
			wait(300);
		motorp(LEFT_MOTOR, -22);
			motorp(RIGHT_MOTOR, 28);
			wait(1700);
			motorp(LEFT_MOTOR, 0);
			motorp(RIGHT_MOTOR, 0);
			wait(300);
			*(unsigned char *)  0x7000=0x00;
		}
}
}
/*************************** End of Main ******************************/
```