Currently

On Thursday, March 13, the building group worked on the robot and the iRobot. The programmers learned how to use the iRobot simulator, make the robot spin in place clockwise and counterclockwise, and make the robot stop spinning at a certain direction. The programmers also got the robot on the simulator to the shelter on the other side. We have completed in using the iRobot to use the camera to follow the blue cup (satellite).

Our game plan is to move the blue cups and the umbrellas into space. That way, all of the deductive points are gone unless the opponent puts the umbrellas or cups on our board. We will then end up with 25 points at the most.

Robot Testing for Reliability

Robot reliability is a huge factor in any strategy. It just isn't practical to have a robot that will only work half the time, giving you and your team bad results and a robot you cant count on. When we finished our robot, we designed a test program to test our robots functionality. After several runs we found out that our programming was off. We had a lot of trouble programming the robot step by step, it would only knock the cups we wanted over the edge of the board once every blue moon. We decided to switch to a more reliable infrared sensor to run a 'sub-program' type function to knock the cups over. When we finished installing the sensor and changed the program to use the infrared sensor, our robot knocks the cup over the side every time. That is an excellent result for robot reliability.

Botball Lessons Learned

The programmers learned how to make the robot move at different speeds, grab with a claw, turn, use servos, and how to program on the iRobot. The builders learned how to make a successful robot and the functions on the iRobot to make it use the program in the XBC. Everyone learned how to handle the robot, cooperate, and work together.

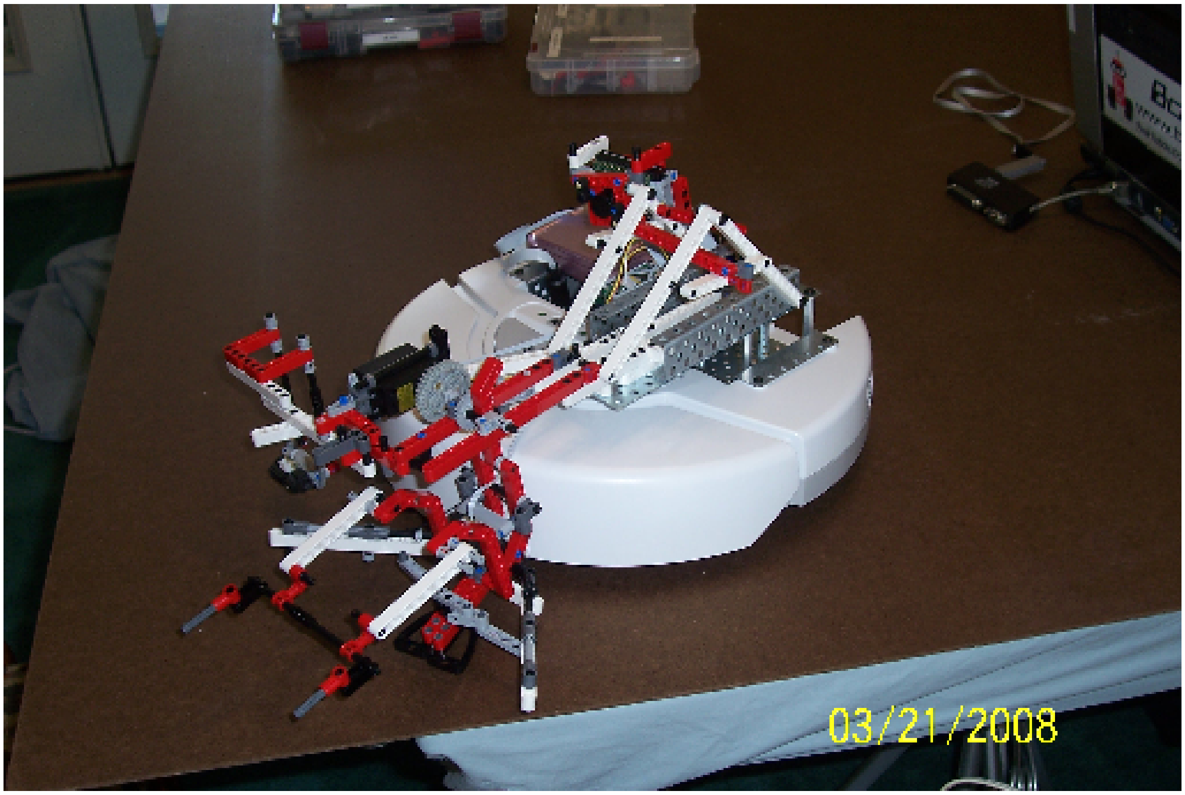
Pictures

We took pictures of the robots on as they were being built, but our computer that we saved them on has technical difficulties opening them and we already deleted them from the camera. Due to this, we will try to describe it the best we can.

The construction of our robot began one fateful Saturday afternoon. We started using the basic support structure we built at the workshop to mount the XBC. We started out by attaching a set of 'rails' to mount or function arm to knock the cups over. We screwed these into place and built an 'A' style bridge support frame.

We attached two long arms to our rails and used the A-frames to support hold them off the ground. After this, we attached a servo to the end of the arm and built a boot style arm to kick over our cups. After attaching the boot to the servo, we built a horseshoe on the end of the arm to collect the workers and plants. Then we built a clothesline bar across the top of the arm to allow our boot to get some purchase on the bottom of the cup. After all our functioning arms where built we installed two infrared sensors to help us operate our bot. One sensor is on the far right of the arm to help sense the PVC piping and other obstacles; and another sensor is on the arm to sense where the various objects we want to move. After this, our final addition to the robot was the camera to help track our objectives and further improve the reliability and the accuracy of our robot.

We also have a "Glamour Shot" of one of our robots. The robot's name is iCake.



Program for the Interactive C 8.0.2

The program here makes iCake drive forward at a speed of 260 as long as needed. iCakke can sense when it is next to a pipe segment using a IR Distance Sensor.
iCake will then stop and push the blue cup over the pipe border into "space" and then drive backwards at a speed of 50.

```
void main() {
   printf("Progaram Start\n");
   create_connect();
   create_sensor_update();
   enable_servos();
   create_drive_straight(260);
```

```
    while(analog(0) < 245); // Stays in loop, repeats untill analog(0) = 255.

    set_servo_position(0,120);// kick
    create_drive_straight(-50); // backup so it will get out of the way
    sleep(3.0);              // and fall over
    set_servo_position(0,160);
    disable_servos();
    create_disconnect();
    printf("Program End\n");
}
```