

Lab 06  
CPE 470  
Team 6 Report

Jessica Smith, Luke Fraser

**11 November, 2013**

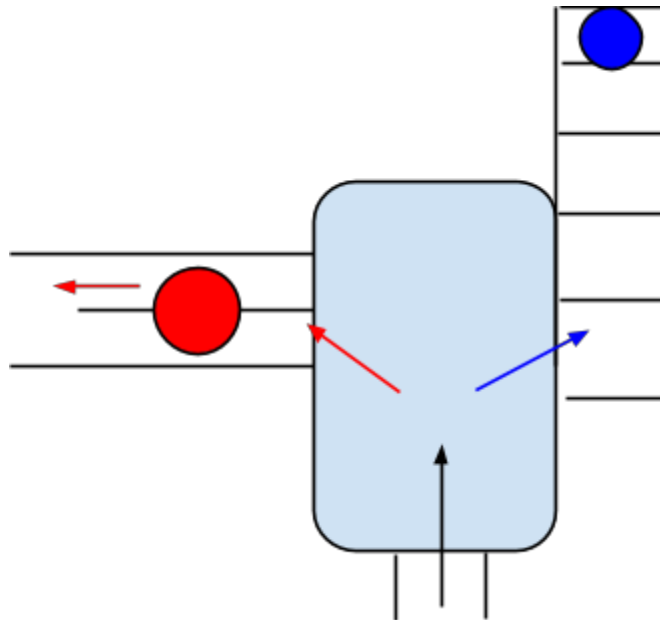
## Introduction

In this lab we were tasked with sorting two different colored balls. The red balls had to be thrown over the side of the board and the blue balls needed to be moved from one side of the board to the other. The robot had to use a compass sensor to detect the direction in which it should travel.

## Hardware and Design

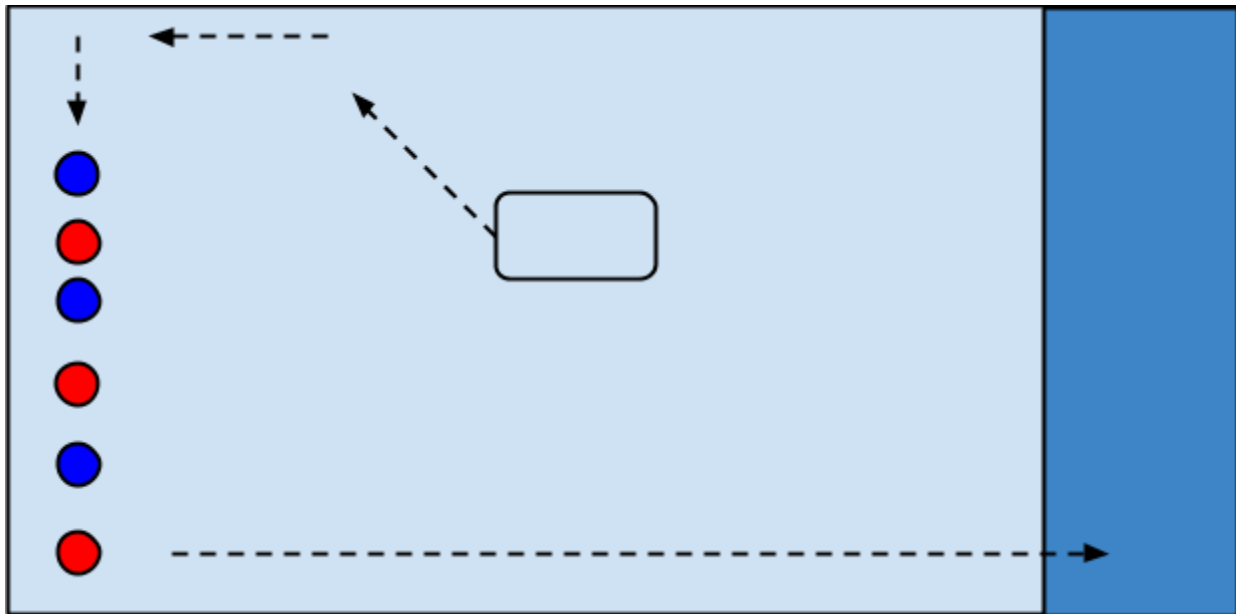
Our robot used a color sensor to detect the balls. It would search for red and blue, and if it found one, it would initialize a robot arm. If it saw blue balls, the arm moved more slowly because in our design, if the ball landed on the top of the robot slowly, it would be deflected to the left, which was a carrying case for the balls. The goal was to collect all the blue balls instead of carrying them back and forth between the sides of the board. If the color sensor detected a red ball, it would throw the ball beyond the blue catching mechanism and off to the other side of the robot. This design is seen in Figure 1.

**Figure 1.** The ball sorting strategy



The robot would follow a wall so that the red ball ejector was over the wall, and the red balls would slide off the edge of the course. The initial direction was found by the compass on the robot, but the wall was followed by two sonar sensors. The path followed by the robot for the competition can be seen in Figure 2.

**Figure 2.** The wall-following strategy.



## Discussion

Our robot failed to perform in the competition because it had run out of battery. Our NXT does not accurately display the battery level, so it was drained and the arm did not throw the balls correctly to sort them. We also had issues constructing the collection part for the blue balls because it we ran out of parts. The robot also had a difficult time initially finding the right-hand wall, and would often run in circles.

To improve the design, we would have to adjust the sensitivity of the sonar sensors so that the wall following was more accurate. The battery level greatly impacts the performance of the robot. If the battery is low, the motors move at much different levels so the ball sorting arm was much less accurate. Had our robot followed the wall as expected, it would have performed very well because it would have found most of the balls, and the sorting method we chose was very accurate. It was rare that a ball was mis-sorted in practice. However, the blue walls of the board did activate the robot lifter, which could have caused more problems had the robot been more functional.

# CODE

```
#define COLOR IN_4
```

```
mutex moveMutex;
```

```
void find_black(){
    int sonar = SensorUS(IN_1);
    Acquire(moveMutex);
    while( sonar > 10){
        while(SensorHTCompass(IN_1) >= 180 && SensorHTCompass(IN_1) < 350){
            OnFwd(OUT_A, 20);
            OnFwd(OUT_B, 80);
            Wait(300);
        }
        while(SensorHTCompass(IN_1) >= 180 && SensorHTCompass(IN_1) < 350){
            OnFwd(OUT_A, 80);
            OnFwd(OUT_B, 20);
        }
    }
    Release(moveMutex);
}
void find_right_wall()
{
    int sonar = SensorUS(IN_1);
    find_black();
    while(sonar < 10){
        OnFwd(OUT_B,-50);
        OnFwd(OUT_A,30);
        Wait(200);
    }
    // move_along_wall();
}
```

```
task move(){
    find_black();
    int sonar = SensorUS(IN_1);
    while(1){
        if( sonar < 10){
            OnFwd(OUT_B,-50);
            OnFwd(OUT_A,30);
            Wait(200);
        }
    }
}
```

```

}
//void check_color()

// The ball lift function will change the speed of lift depending on color of ball
// this is how the ball will go right or left on the robot
task detect_ball_lift()
{
    while(true){
        unsigned int color = SensorHTColorNum(COLOR);
        ClearScreen();
        if(color == 9){
            NumOut(0, LCD_LINE2, color);
            Acquire(moveMutex);
            OnFwd(OUT_AB, 0);
            OnFwd(OUT_C, 100);
            Wait(205);
            Off(OUT_C);
            Wait(300);
            OnFwd(OUT_C, -50);
            Wait(500);
            Off(OUT_C);
            Release(moveMutex);
        }
        else if(color == 2){
            Acquire(moveMutex);
            OnFwd(OUT_AB, 0);
            OnFwd(OUT_C, 25);
            Wait(1100);
            Off(OUT_C);
            Wait(100);
            OnFwd(OUT_C, -25);
            Wait(1100);
            Off(OUT_C);
            Release(moveMutex);
        }
        NumOut(0, LCD_LINE1, color);
    }
}

task main(){
    SetSensorLowspeed(IN_3);
    SetSensorLowspeed(IN_4);
    SetHTColor2Mode(IN_4, HT_CMD_COLOR2_NEAR);
    SetSensorLowspeed(IN_1);
    Precedes(detect_ball_lift, move);
}

```

