Lab 03
Jessica Smith and Luke Fraser
Team 06

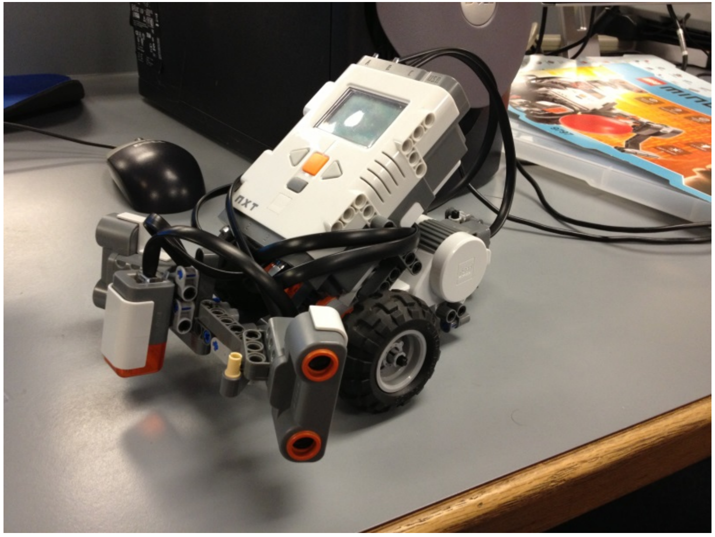
Introduction

The goal of the contest is to move through a tunnel, read a RFID card, emit a beep to signal the card has been read, and return without ever touching the sides of the tunnel. The tunnel has curves in it, and is made of boxes so the surface of the tunnel is also uneven. The team that completed the tunnel the fastest was the winner.

Description

In order to avoid the walls, two ultrasonic sensors were mounted to the front of the robot and angled to the left and right at about a 50° angle. In order to avoid the walls of the tunnel, we approached the problem by attempting to stay in the middle of the tunnel. We took the difference between the two sensors and if one was larger than the other, we compensated it by turning. There was a buffer of 10 units that the NXT could be in and be moving forward as fast as possible to increase the speed at which it traversed the tunnel. It was consistently doing this polling while we were waiting for the RFID card to be located. Once the card was located, we reversed one of the motors completely to turn around and followed the same logic on the way back. The RFID sensor was in the center of the robot and close to the ground.

Figure 1: Our robot configuration



Note: The angle of the sensors is about a 50° angle and the RFID card reader is in between them, and very close to the ground.

Problems

We did not have any unresolved problems, and our robot made it through the tunnel with no problems.

Discussion

We did not account for the problem of missing the RFID card. Had we missed it in the competition, we would likely have rammed the back of the tunnel. However, we designed the sensor to be extremely close to the ground so that there would be less of a chance to miss the ID.

Conclusions

Our robot performed very well, no matter the angle at which it started. It was designed to correct for any hard turns by using the central buffer. It would have had a hard time with a very hard angle, but if we adjusted the sharpness of the correcting turn, it would compensate for a different course.

Appendix

Daniele Benedettelli's RFID library was also used in this project.

```
#include "RFIDlib.nxc"
// tunnel traveler Program

#define PORT IN_3
mutex moveMutex;

bool isturned = false;

void moveLeftSideM(){
    OnFwd(OUT_A,100);
    OnFwd(OUT_B,70);
    Wait(200);
}

void moveRightSideM(){
    OnFwd(OUT_B,100);
    OnFwd(OUT_A,70);
    Wait(200);
}

void turnaround(){
     OnFwd(OUT_B,100);
     OnFwd(OUT_A,-100);
     Wait(400);
}

task moveBothSideM(){
    while(true){
          Acquire(moveMutex);
        OnFwd(OUT_AB,100);
        Release(moveMutex);
    }
}

task check_sensor(){
     while(true){
           int dist = SensorUS(IN_1) - SensorUS(IN_4);
         short gap = 5;
         if(dist < -gap)
         {
            Acquire(moveMutex);
            moveLeftSideM();
            Release(moveMutex);
         }
         if(dist > gap){
            Acquire(moveMutex);
            moveRightSideM();
            Release(moveMutex);
         }
     }

}

task check_RFID(){
     // send dummy command to wake up sensor
     RFIDDummy(PORT);
     byte a[5];



     while(!isturned){
         GetRFIDArray(PORT,a,true);
         //ClearScreen();

         //NumOut(0,LCD_LINE1,10*(a[0]+a[1]+a[2]+a[3]+a[4]));
         if((a[0]+a[1]+a[2]+a[3]+a[4]) > 0){
             Acquire(moveMutex);
             turnaround();
             Release(moveMutex);
             isturned = true;
             PlayTone(1200,1500);
         }
     }
}

task main(){
        SetSensorLowspeed(IN_1);
        SetSensorLowspeed(IN_4);
        SetSensorLowspeed(PORT);
        Precedes(check_RFID, check_sensor, moveBothSideM);
}
```