

CSCI 458: Autonomous Mobile Robotics

Assignment Name: Garbage Collector

Assignment Number: 5

Group Members: John Buckley, Charles Clayton

Project Description:

Similar to the previous assignment, our robot will utilize the color sensor to navigate within a bounded region with the goal to push all objects within the bounded region outside of the boundary and stop when the bounded region is clear of all objects.

What Worked:

We recycled the code from assignment 3 to allow us to detect objects with an ultrasonic sensor and we recycled code from the previous assignment, however, instead of having the robot avoid crossing the black line we had the robot travel a set distance forward when encountering a black line, this insured us that the object was far enough outside the bounded region that the robot would not pick it up with the ultrasonic sensor when it continued to look for more obstacles. The robot ran this code within a timed while loop of 10 seconds, and if the robot didn't execute any functions within that 10 seconds then it would stop, as the bounded region would be free of obstacles.

What Didn't Work:

We initially tried to push the obstacles forward until a black line was reached to keep our robot within the bounded region at all times, however doing this would make our robot detect it again but would fail to be able to push it any further as the black line would be stopping it from doing so, hence being stuck in an infinite loop. We fixed this problem by telling our robot to push the obstacle forward a set distance every time after a black line was encountered ensuring that the ultrasonic sensor would not detect it again.

What we learned from this assignment:

That the speed of the robot can affect how quickly and accurately some sensors detect objects.

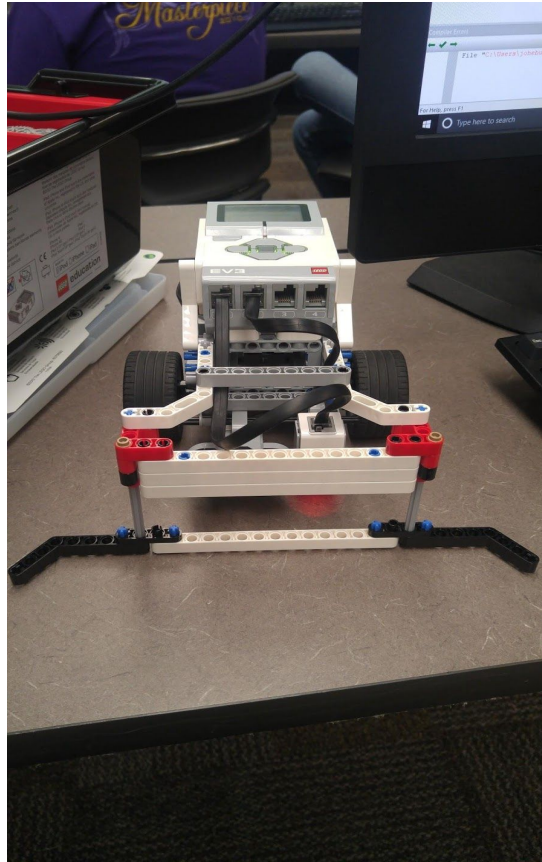


Figure 1: Our first and final design. The scoop allows for a larger barrier to push obstacles.

Source:

```
tMotor LEFT_MOTOR = motorD;
tMotor RIGHT_MOTOR = motorA;

void detecting();
void pushing();

task main(){
    setSoundVolume(100);
    SensorType[S1] = sensorEV3_Color;
    SensorType[S2] = sensorEV3_Ultrasonic;
    clearTimer(T1);

    while(time1[T1]<10000){

        detecting();

    }
}
```

```
}
```

```
void detecting(){  
    setMotorSpeed(LEFT_MOTOR, -15);  
    setMotorSpeed(RIGHT_MOTOR, 15);  
    sleep(400);  
    if(getUSDistance(S2)<60){  
        pushing();  
    }  
}
```

```
void pushing(){  
    setMotorSync(LEFT_MOTOR, RIGHT_MOTOR, 0, 0);  
    sleep(500);  
    setMotorSync(LEFT_MOTOR, RIGHT_MOTOR, 0, 40);  
    waitUntil(getColorReflected(S1) < 15);  
    setMotorSyncEncoder(LEFT_MOTOR, RIGHT_MOTOR, 0, 600, 40);  
    waitUntilMotorStop(RIGHT_MOTOR);  
    setMotorSyncEncoder(LEFT_MOTOR, RIGHT_MOTOR, 0, 1300, -40);  
    waitUntilMotorStop(RIGHT_MOTOR);  
    clearTimer(T1);  
}
```