

## CSCI 458: Autonomous Mobile Robotics

Assignment Name: More Bot Less Bumble

Assignment Number: 3

Group Members: John Buckley, Charles Clayton

### Project Description:

Attach an ultrasonic sensor onto the robot, adjusting the build from assignment 2 as needed to accommodate, and have the robot avoid obstacles using the ultrasonic sensor while keeping the functionality of the touch sensors from assignment 2. Have the ultrasonic sensor display the distance on the brick for debugging purposes. This should result in the robot colliding less with obstacles and avoid them instead.

### What Worked:

We opted to put the ultrasonic sensor close to the ground in order to avoid more obstacles such as table and chair legs. In order to print the distance being recorded by the ultrasonic sensor onto the screen we declared a string and set that string equal to the distance from the sensor and displayed it using `displayBigTextLine` in order to more easily read the output.

```
SensorType[S2] = sensorEV3_Ultrasonic;
string toshow;

while(true){

    toshow = getUSDistance(S2);
    toshow+=" CM";
    displayBigTextLine(3, toshow);
}
```

All we did to incorporate this new sensor was to add a new if statement inside our infinite loop that tells the robot to stop, back up, and turn away from an obstacle with the same amount of randomness from the previous assignment if an obstacle is detected within 13cm.

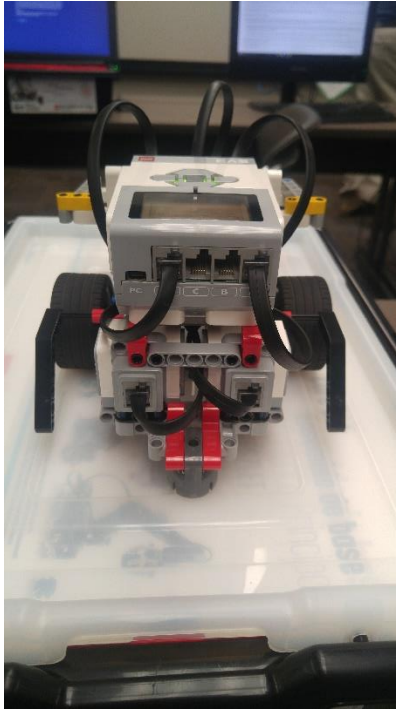
```
if(getUSDistance(S2)<13){
    setMotorSpeed(motorD, 0);
    setMotorSpeed(motorA, 0);
    playSound(soundException);
    setMotorSyncEncoder(LEFT_MOTOR, RIGHT_MOTOR, 0, 350, -50);
    waitUntilMotorStop(RIGHT_MOTOR);
    setMotorSyncEncoder(LEFT_MOTOR, RIGHT_MOTOR, 50, (250+rand()%500), 50);
    sleep(300);
    waitUntilMotorStop(RIGHT_MOTOR);
}
```

**What Didn't Work:**

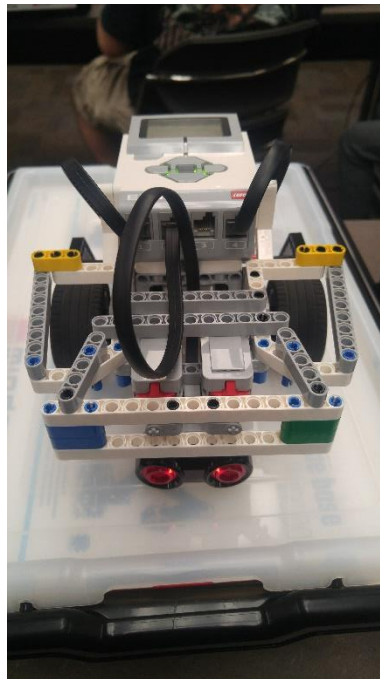
The only trouble we ran into with this assignment was where to mount the ultrasonic sensor. We quickly agreed to mount it as low as possible without it hindering the robots movement in order to detect the maximum amount of objects in our environment.

**What we learned from this assignment:**

This assignment we found to be fairly easier than the rest because it directly relates to part of our senior project which involves a robot with an ultrasonic sensor with obstacle avoidance. However, we learned that it is important to display some readings from sensors onto the display in order to observe the functionality of the sensors and to debug when necessary.



**Figure 1:** We added rear bumpers to our robot to keep the tire from riding over obstacles and flipping.



**Figure 2:** This was our final design, it kept the same design as the previous assignment however the bump bar was moved up to accommodate the ultrasonic sensor.

```

tMotor LEFT_MOTOR = motorD;
tMotor RIGHT_MOTOR = motorA;

task main(){

SensorType[S1] = sensorEV3_Touch;
SensorType[S4] = sensorEV3_Touch;
SensorType[S2] = sensorEV3_Ultrasonic;
string toshow;

while(true){
    setMotorSpeed(motorD, 50);
    setMotorSpeed(motorA, 50);

    //Prints distance on display
    toshow = getUSDistance(S2);
    toshow+=" CM";
    displayBigTextLine(3, toshow);

    //Ultrasonic sensor
    if(getUSDistance(S2)<13){
        setMotorSpeed(motorD, 0);
        setMotorSpeed(motorA, 0);
        playSound(soundException);
        setMotorSyncEncoder(LEFT_MOTOR, RIGHT_MOTOR, 0, 350, -50);
        waitUntilMotorStop(RIGHT_MOTOR);
        setMotorSyncEncoder(LEFT_MOTOR, RIGHT_MOTOR, 50, (250+rand()%500), 50);
        sleep(300);
        waitUntilMotorStop(RIGHT_MOTOR);
    }

    //Following 3 if statements are touch sensors
    if(getTouchValue(S4)==1&&getTouchValue(S1)==1){
        setMotorSpeed(motorD, 0);
        setMotorSpeed(motorA, 0);
        playSound(soundBeepBeep);
        setMotorSyncEncoder(LEFT_MOTOR, RIGHT_MOTOR, 0, 800, -50);
        waitUntilMotorStop(RIGHT_MOTOR);
        setMotorSyncEncoder(RIGHT_MOTOR, LEFT_MOTOR, -50, 700, 50);
        sleep(300);
        waitUntilMotorStop(RIGHT_MOTOR);
    }

    if(getTouchValue(S1) == 1){
        setMotorSpeed(motorD, 0);
        setMotorSpeed(motorA, 0);
        setMotorSyncEncoder(LEFT_MOTOR, RIGHT_MOTOR, 0, 350, -50);
        waitUntilMotorStop(RIGHT_MOTOR);
        setMotorSyncEncoder(LEFT_MOTOR, RIGHT_MOTOR, 50, (500+rand()%500), 50);
        sleep(300);
        waitUntilMotorStop(RIGHT_MOTOR);
    }

    if(getTouchValue(S4)==1){
        setMotorSpeed(motorD,0);
        setMotorSpeed(motorA,0);
        setMotorSyncEncoder(LEFT_MOTOR, RIGHT_MOTOR, 0, 350, -50);
        waitUntilMotorStop(RIGHT_MOTOR);
        setMotorSyncEncoder(LEFT_MOTOR, RIGHT_MOTOR, -50, (500+rand()%500), 50);
        sleep(300);
        waitUntilMotorStop(RIGHT_MOTOR);
    }

}

}
}

```