CSCI 490: Autonomous Mobile Robotics

Assignment Name: More Bot Less Bumble

Assignment Number: 3

Group Members: Ricky Hempel & Nisha Patel

In this assignment we are using the ultrasonic sensor with the touch sensors. The assignment is divided into two parts: part one measure the distance of the nearest object in front of the robot and display the distance on the built-in-display and part two add the ultrasonic sensor and the bump sensors to the part one functionality. The robot will avoid any obstacles primary using ultrasonic sensor, but use the touch sensors so that it will not get hung up on objects. When it detects an object in its path it should attempt to avoid collisions and continue to explore its surroundings and the robot is designed to do an avoidance within approximately 8-9 cm from an obstacle. Adding the required changes to assignment 2 functionality, we have design our robot for assignment 3.

This is what worked. We put the ultrasonic sensor as low as we could get it, so that it would detect the bottom of every object. We also lifted our bumper a little bit higher from our previous design so that it wouldn't block the ultrasonic sensor. After changing the physical placement of our design, that fixed the problem. We also added parts to help hold and keep the touch sensors stable, so the weight did not make them fall down.

With the software, we used assignment 2 script and make appropriate changes by adding required functionality to it. The changes were that we include a case for that if the sensor detects something less than 9cm, one of the touch sensors is touched, or both of the sensors are touched to turn a certain direction, or a random direction.

This is what did not work. With the hardware, we were having difficulties putting our ultrasonic sensor on so that it would detect objects without bouncing from the bumper that the touch sensors were on. Also, we tried to attached it behind the touch sensor, but that didn't work because the way our bumper was designed, it was blocking ultrasonic sensor. Lastly, we tried turning the sensor to get it closer to the ground, but we found out quickly because of the way the sensor is made that it would not work.

With the software we tried different ways to get the distance value. Next we tried to get the robot to go until the distance was not equal to 9, but found out quickly that was not the way to go. Lastly, with the software we tried to combine different else statements to make them longer than they should be, so we decided to keep them as simple as possible.

In this assignment, we learned about how to use ultrasonic sensor and how it works in terms of programming it in such as way that it can detect objects in its path while avoiding both collisions and continue exploring and stops within approximately 8-9 cm from an obstacle. We

did some outside research to figure out how to use the function SensorValue() and how to generate random numbers.

We also learned how to use two different type of sensors to achieve a goal. While this was going on we learned the pros and cons about where to attached to the ultrasonic sensors. For example, if it was attached to high it would not sense everything. Next, we learn how weight factors to our bumper design. When we attached it to the top of the brick it would fall over we had to add some support from the bottom to keep it stable. Another, aspect of the bumper we tried was to make it go out more, but that was not activating the sensors and was too complex, so again decided to make it as simple as we could.

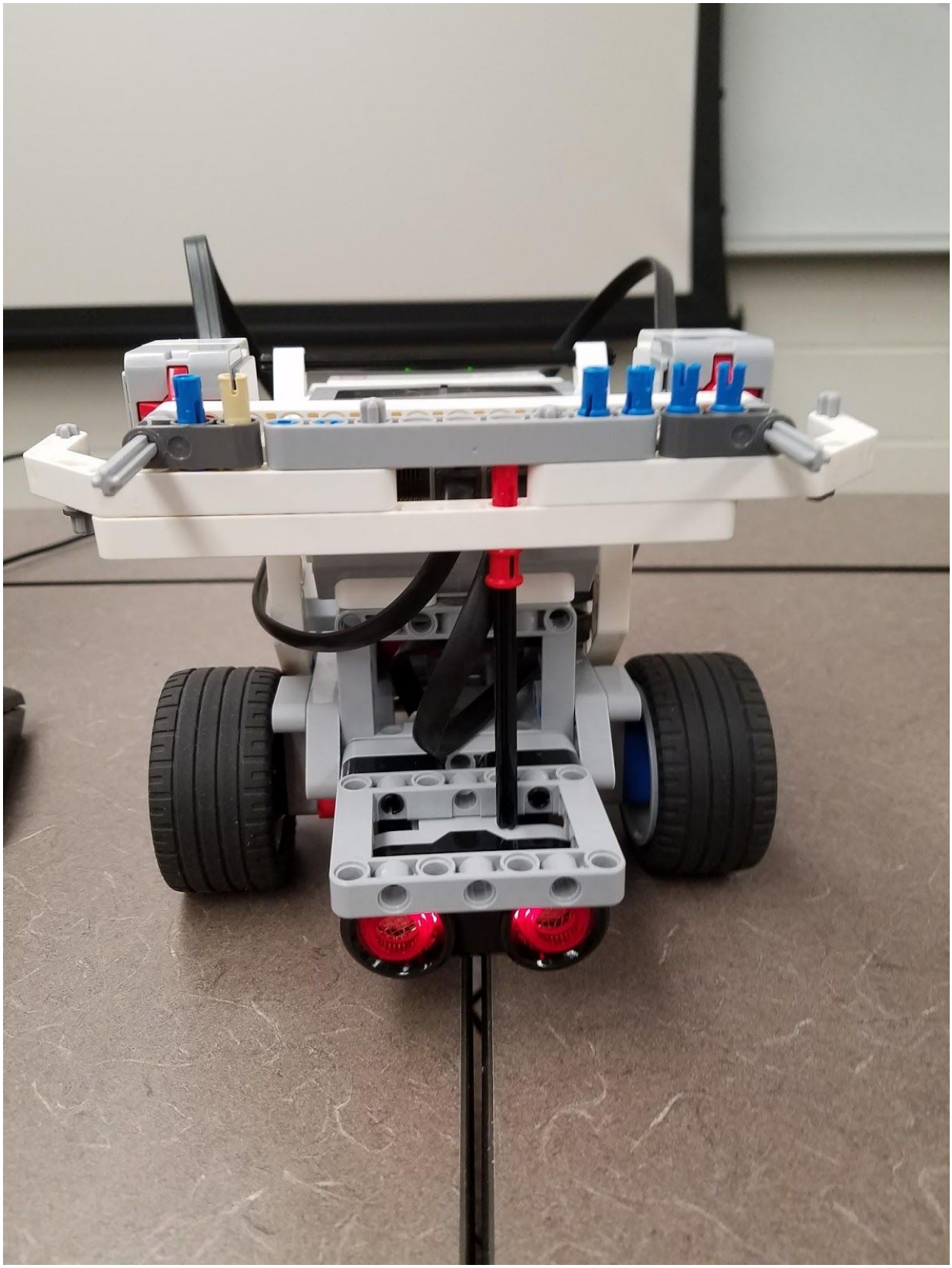

Figure 1: Robot with touch and ultrasonic sensors

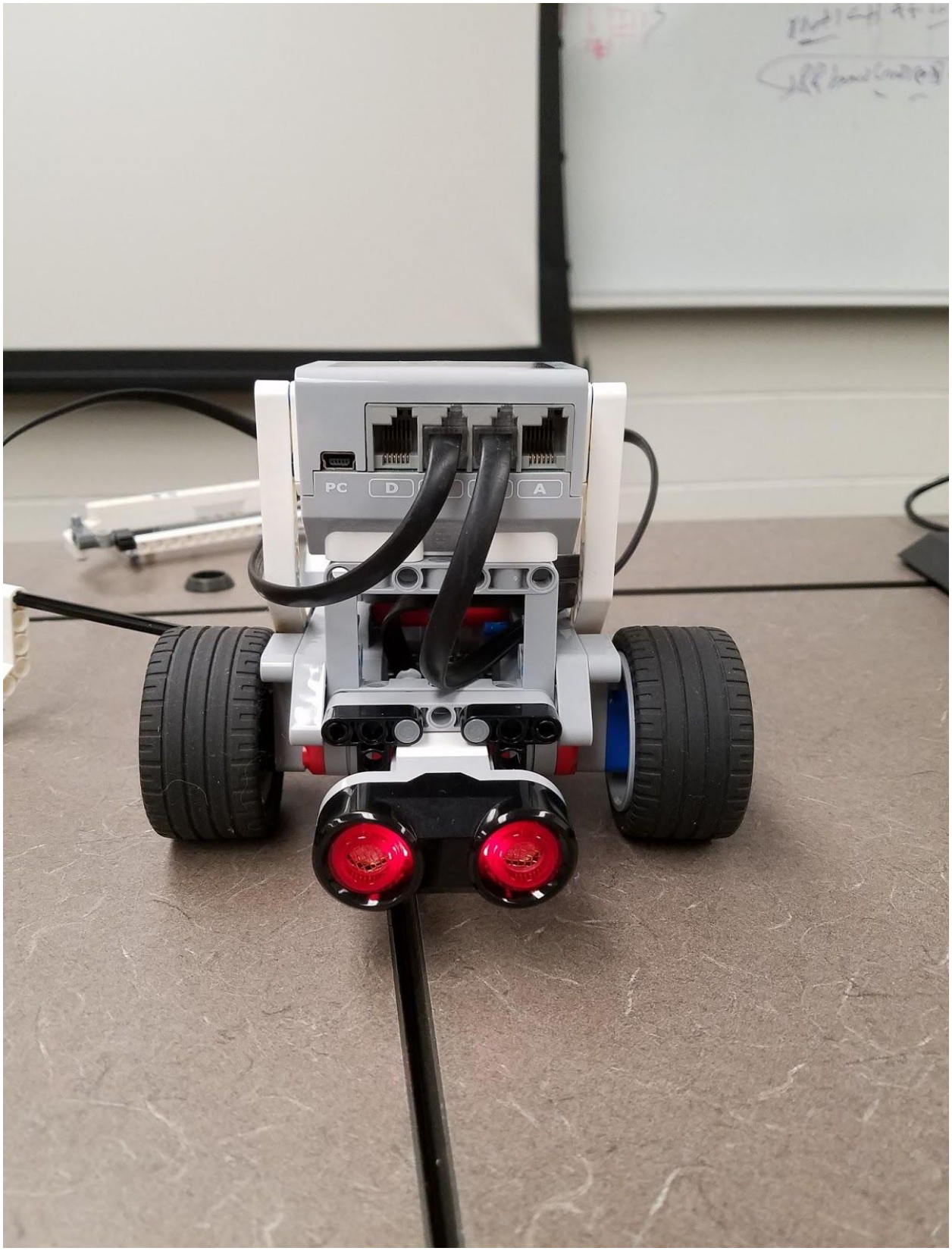

Figure 2: Robot with only ultrasonic sensor

```
/****************
Name Ricky Hempel & Nisha Patel
Purpose source code for ultrasonic.c
Date   10/13/17
**************/
//sesnor info
#pragma config(StandardModel, "EV3_REMBOT")
#pragma config(Sensor, S4,     sonar4,         sensorEV3_Ultrasonic)
//for motors
tMotor Left_Motor=motorB;
tMotor Right_Motor=motorC;
//main
task main()
{
        //repeat forever
        while (true)
        {
        //repeat until distnace is true or a sensor is touched
         while(SensorValue(sonar4)>9 && getTouchValue(S1) == 0 && getTouchValue(S4) == 0)
           {
                //move the robot foward
                setMotorSpeed(Left_Motor, 50);
                setMotorSpeed(Right_Motor, 50);
           }

                 //move the robot backwards
                 setMotorSpeed(Left_Motor, -10);
                 setMotorSpeed(Right_Motor, -10);
                 sleep(1000);

                 //if it touches left, go right
                 if(getTouchValue(S1) == 0 && getTouchValue(S4)==1)
```

```
{
            //turn right
            motor[Left_Motor]=10;
            motor[Right_Motor]=-10;
            sleep(2000);
}

 //if it touches right, go left
else if(getTouchValue(S4) == 0 && getTouchValue(S1)==1)
{
           //turn right
           motor[Left_Motor]=-10;
           motor[Right_Motor]=10;
           sleep(2000);
}

//if it hits the center, turn right
else  if(getTouchValue(S1) == 1 && getTouchValue(S4) == 1)
    {
           motor[Left_Motor]=10;
           motor[Right_Motor]=-10;
           sleep(2000);
     }

//else turn right
else if(random(10)<5)
{
            setMotorSpeed(Left_Motor, 10);
            setMotorSpeed(Right_Motor, -10);
            sleep(1875);
  }
```

```
                //finally turn left if all else are not true
                 else
                   {
                        setMotorSpeed(Left_Motor, -10);
                        setMotorSpeed(Right_Motor, 10);
                        sleep(1875);
                   }
          }
}
```