CSCI 490: Autonomous Mobile Robotics

Assignment Name: Bumble Bot (a sense of touch)

Assignment Number: 2

Group Members: Ricky Hempel & Nisha Patel

Project Description: A brief description of what was asked of you, and the general constraints of the problem.

In this assignment,Bumble Bot, we have been asked to construct a robot using the touch sensors. The robot should react to touch input. For instance, if it touches an object, it should drive away from the object and continue on its way. Our robot is programmed to define different behaviors based on which or if both touch sensors are touched.

Furthermore, a constraint of the problem was that the robot could get stuck if something got in between the touch sensors.

What Worked (Source and/or Build): Describe things that you tried either in the hardware or software (algorithmically, data structure choices, API choices) that work well (and likely made it into the final product) under the assignment conditions.

With the hardware, we used gray axle, 5-module and attached it on the sensor for the bumper. Then, we attached double angular beams to connect the modules, so it is capable of exploring without getting hung up on objects. We also added two beams that fit in the holes of the touch sensors, so that they apply enough pressure that would easily press in the touch sensors if they were hit by an object, but not enough pressure that they would set the sensors off without hitting an object.

With the algorithm, we chose a simple approach. We kept it simple so it is easy to read and follow through our code. We used loops and if-else statements for creating three conditions for hitting the objects and moving in a specific new direction. For example, one of the if conditions we used is as follow, if our robot touches something with the right touch sensor, reverse then go left.

The API choices that we used were the getTouchValue() function for each of the sensors. This function allowed us to make conditionals based on which or both of the sensors got touched. These API was very easy to understand for this project.

What Didn't Work (Source and/or Build): Describe things that you tried either in the hardware or software (algorithmically, data structure choices, API choices) that didn't work well under the assignment conditions.

We had some difficulties with the bumper. We tried several different designs in order to get the right results. Our previous design was not reacting appropriately when it hit an object, so we had to change our bumper design. Another issue we faced was attaching the bumper to the robot; it was a challenge to find the right pieces that would make the bumper be the right length so that it could reach the other side. Also, finding pieces that would make the right depth, so that it did not touch the sensors, but was not too far from the robot.

What I feel that I learned from this assignment: Describe any algorithm/data structure insights, features of the language, or creative uses of the hardware that you picked up during the assignment.

In this assignment, we learned about how touch sensors work in terms of programming it in such a way that it can react in a certain way in a given condition. Also, we learned about how to use getTouchValue() for the two sensors.

We also learned how to get an input from a physical object. That is how to handle input that is coming from something other than a keyboard. The input we had to handle came from an ever changing environment, we had to learn how to think if the robot was put in a tight corner or an open room.

We divided this assignment into subproblems and tried to get the solution for a small problem before merging them together to complete the overall goal. For example, we wrote a code for hitting objects in the center with both touch sensors, then move backward and go right Overall, this assignment was fun learning and trying new things with our robot.

Build Details Please provide one or more representative images of your configuration with a description on this page (most likely a page or two).

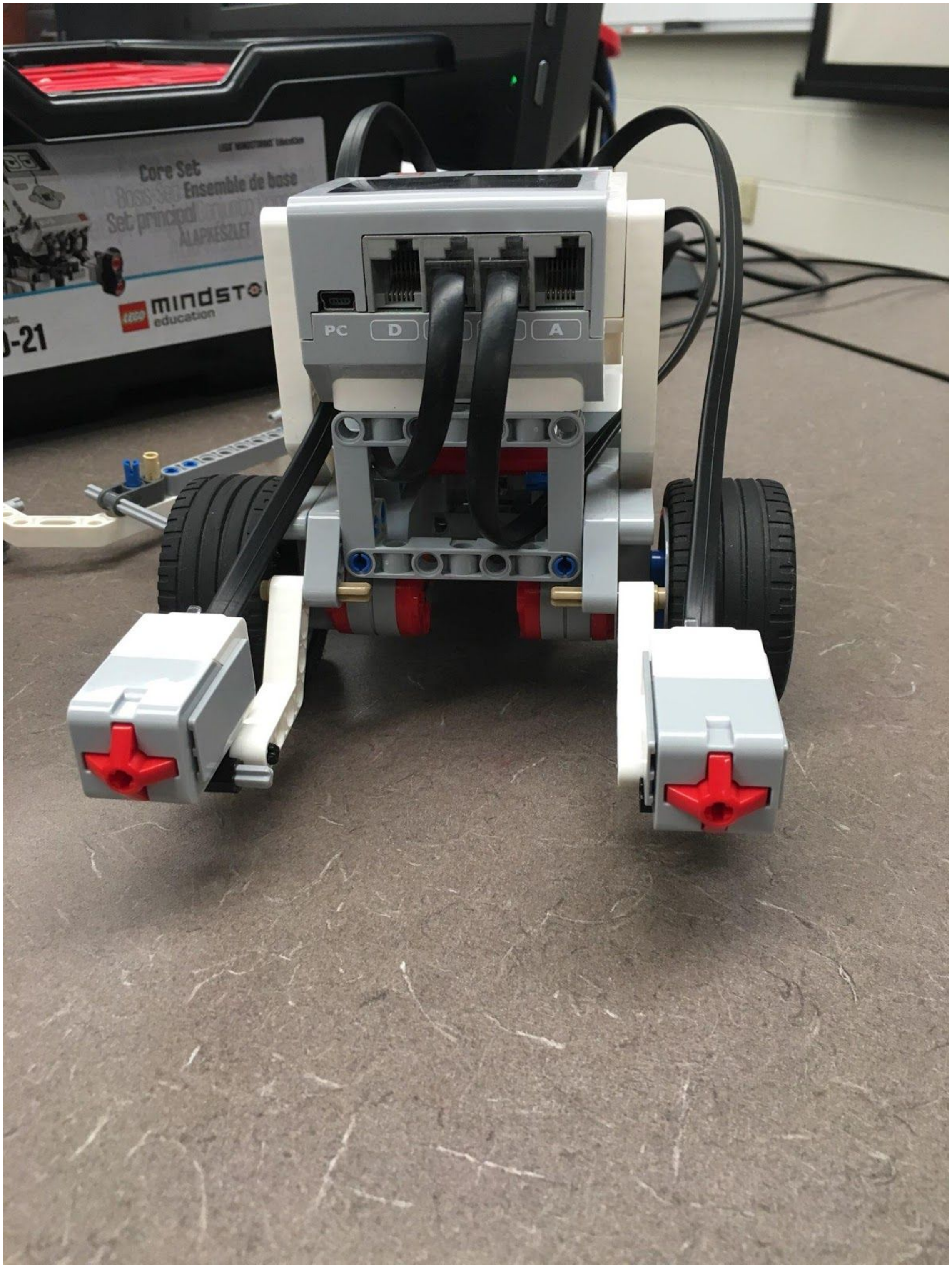

Figure 1: Robot with touch sensors.

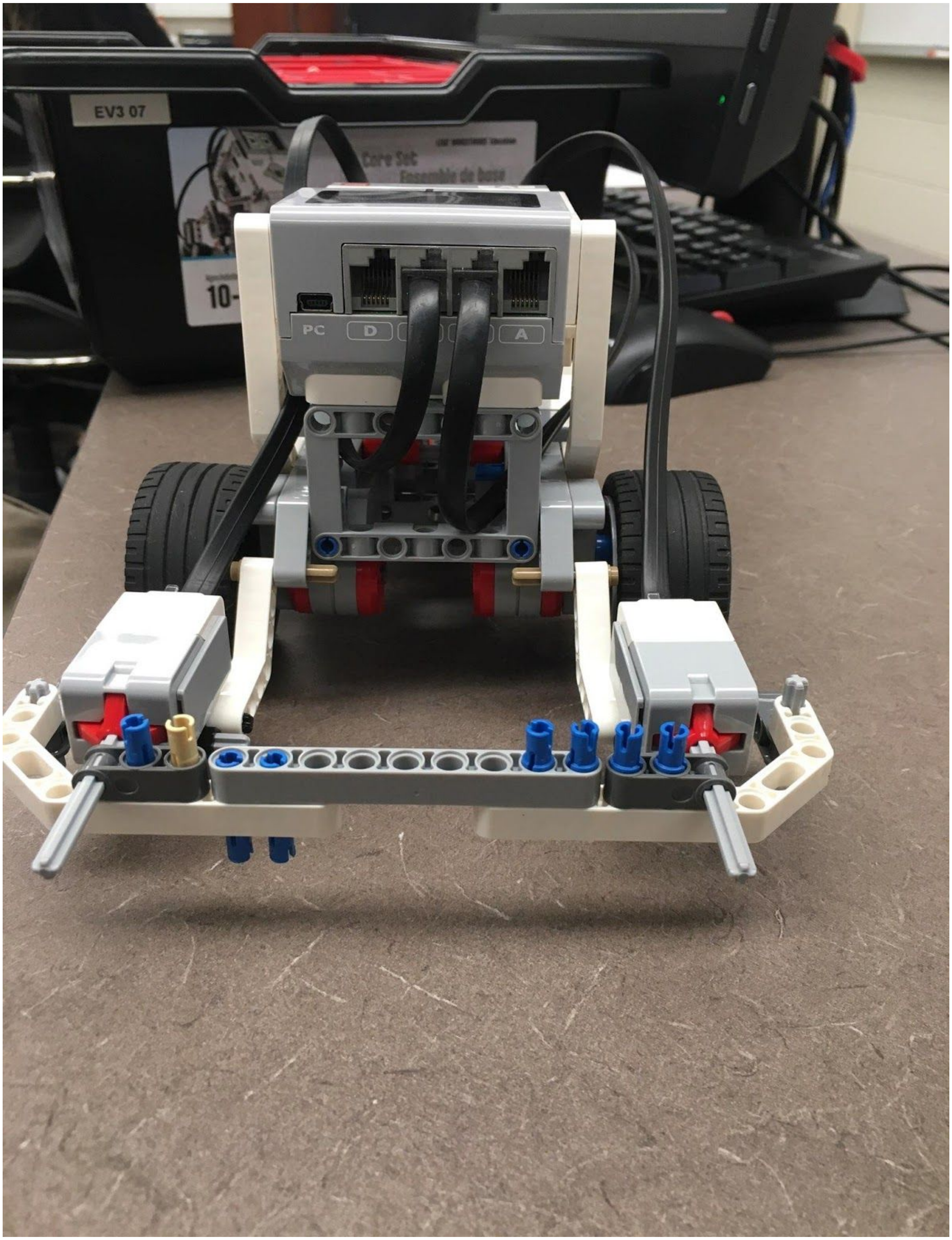

Figure 2: Robot with touch sensors and bummer.

Source Please include documented source(s) here (one file per page).

```
tMotor Left_Motor=motorB;
tMotor Right_Motor=motorC;
task main(){
    //repeat forever
    while(true){
    //goes forward at 50 speed
    repeatUntil(getTouchValue(S1) == 1 || getTouchValue(S4) == 1){
        setMotorSpeed(motorB, 50);
        setMotorSpeed(motorC, 50);
    }   //reverses at 50 speed.
            setMotorSpeed(motorB, -50);
            setMotorSpeed(motorC, -50);
            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 left
            motor[Left_Motor]=-10;
            motor[Right_Motor]=10;
            sleep(2000);
            }
            //if touches both go right
            if(getTouchValue(S1) == 0 && getTouchValue(S4) == 0){
            //turn right
            motor[Left_Motor]=10;
            motor[Right_Motor]=-10;
            sleep(2000);  }
            wait1Msec(2000);
            }
}
```