

Duke/DPS Robotics Program

10/22

Lesson Plan 2

Goal: *Create a bumper car that can push other cars around.*

Materials Needed: *Brookbot with touch sensor and bumper.*

Part 1: Dead Reckoning, Continued

To finish up what we started last week, have your team write a program that drives the robot in a square around the tabletop. They will have to figure out how to make a 90-degree turn for this to work. Show them how this can be done by turning the wheels a specific number of degrees, or for a specific amount of time at a given power level. Guessing and checking is fine! This should take no more than 15 minutes.

Part 2: The Touch Sensor

As always, the first thing you should do is the Motors and Sensors setup. This time, in addition the motors, click on the sensors tab and select Touch Sensor from the dropdown menu of the appropriate port:

Port	Name	Type
S1		Touch
S2		No Sensor
S3		No Sensor
S4		No Sensor

A. Conditionals

To use sensor input, we need to be familiar with conditionals. Familiarize your team with the use of the if statement. Be sure to mention that the code inside the parentheses is the *condition*, while the code inside the brackets is the *body* of the conditional that is executed if the condition is true. Write a simple program to demonstrate the use of conditionals, such as the following:

```
task main()
{
    if (2 + 2 == 4)
    {
        motor[motorA] = 50;
        wait1Msec(5000);
    }
    if (2 + 2 == 5)
    {
        motor[motorC] = 50;
        wait1Msec(5000);
    }
}
```

Only motor A should come on. Stress the fact that they must use the comparison operator “==” rather than the assignment operator “=” when writing conditions.

B. Loops

Before we start using the sensor, we want our programs to be running continuously. Therefore, most programs we write will be sandwiched inside an infinite while loop:

```
task main()
{
    while (true)
    {

    }
}
```

Make sure to point out that what’s between the parentheses is the *condition* that drives the loop, while what’s between the curly braces is the *body* of the loop that actually gets executed.

C. Construction

For the subsequent parts, your Brookbot will need a touch sensor mounted with a bumper attached. Feel free to use the default Brookbot touch bumper, or to have students make their own. It is suggested that you split the team; Have two work on the bumper and two continue programming. Make sure your builders stay on task and finish in a timely fashion!

D. Sensor Input

The final tool the students need to build the bumper car is the `SensorValue()` function. Explain to them that function returns a number representing the state of the sensor. In the case of the touch sensor, it will return 0 if the sensor is currently untouched, or 1 if the sensor is depressed. With this function, loops, and conditionals, guide them as they write a program that drives forward when the touch sensor is not depressed, and stops when it is depressed.

```
task main()
{
    while (true)
    {
        if (SensorValue[S1] == 0)
        {
            motor[motorA] = 50;
            motor[motorC] = 50;
        }

        if (SensorValue[S1] == 1)
        {
            motor[motorA] = 0;
            motor[motorC] = 0;
        }
    }
}
```

At this point you may want to cover the `else` statement, so you can eliminate the redundant second condition in the program above.

Part 3: Bumper Car Challenge

Using these programming tools and your bumper-equipped robot, build the best bumper car you can! The above program that goes straight till the bumper is triggered is a good starting point. From that point, your group can decide what to do. Some options are: backing up and then ramming forward at full speed, backing up, turning and moving in a different direction, or moving randomly. The bumper cars will be judged on both creativity and effectiveness.

Optional

If you have considerable time left, you can have your team change the gearing on the robot to gain more power or more speed, depending on what your team wants to achieve. Do this **ONLY** if you have enough time to explain gearing up vs. gearing down and torque. Do not simply re-gear the robot for them.