Robotics Club - Colony
Introductory Lab #1 – Target Practice

Release Date: 9/19/09

Demo Date: 9/25/09

Prerequisite

This lab assumes you have already learned how to set up the programming environment as described in Introductory Lab #0, which is available online. It also assumes that you have learned the basics of programming the robot to move around.

Objectives

1. Develop an understanding of and learn how to use the BOM, rangefinders, and encoders
2. Work in a group to implement Target Practice – a seeker robot will navigate towards another robot and then take its position once the target robot is removed

Objective 1: Getting the Robot to Learn About Its Environment

Lab 1 will familiarize you with some of the more complicated input sensors on the Colony robots. Detailed explanations of the functions associated with each sensor can be found in the API located on the website. To exhibit your mastery of these functions, you will make a seeker robot locate and move towards a stationary robot (the "beacon" robot), attempting to come as close as possible to the location of the beacon robot.

Bearing and Orientation Module (BOM)

The bearing and orientation module is a sensor developed by the Colony project which allows the robots to determine their relative position to another robot or infrared source. The sensor consists of 16 infrared emitter-detector pairs (the pink and black components on the top of the robot). In this lab, the target robot will be broadcasting infrared radiation (IR) which will be detected by the black IR detectors of the seeking robot. By determining which of the detector readings is the most intense, the robot knows which angle (out of 16 possible angles) the light is coming from relative to the front of the robot and can use this to navigate towards the light.

Use the `bom_on` and `bom_off` functions to turn the infrared LEDs on and off respectively. To get information out of the BOM, use the `bom_refresh` and the `bom_get_max` functions. The following code example stores the index of the most intense sensor reading in the variable `bom_index`:

```
bom_refresh(BOM_ALL);
bom_index = bom_get_max();
```

The `bom_refresh` function tells the robot to re-poll all of the detectors. This makes sure you have recent information when you call `bom_get_max`, which gives you the index of the sensor that has the most intense IR reading (you can generally assume that there is another robot with its BOM activated in this direction).

For this lab, be aware that the BOM 1.0 board is yellow, while the BOM 1.5 board is green. The distinction will be important because the BOM 1.0 board will be used only for the seeker robot and the BOM 1.5 board will be used only for the beacon. Also, be careful when accessing the BOM sensors, as the BOM sensors are numbered starting with 1 while the `bom_get_max` function returns values starting at 0.

IR Rangefinders

The IR rangefinders are the five black rectangular components underneath the main circuit board. There are three in the front that are used to detect obstacles and two on the sides, which can be used to follow walls. These sensors work by emitting infrared light and detecting the angle at which the light reflects. Due to the way they operate, these sensors have many limitations. If the sensor is closer than a few inches to an object, the light will not reflect into the sensor at all, and an invalid reading will be received. An invalid reading will also be detected if the obstacle is too far away from the detector. Within its operating range, however, these rangefinders are an effective and inexpensive way to determine the rough distance to an object.

The following code samples rangefinder 2 (located on the front of the robot) and stores the relative distance value in the variable `distance`:

```
distance = range_read_distance (IR2);
```

Encoders

As you have probably realized from the last lab, the motors on the robot are not very reliable. Telling the robot to drive straight may actually result in a curved path depending on motor and wheel differences, battery levels, and the last digit of the day's DOW Jones stock index. In order to address these issues, the motors are equipped with magnetic encoders. Magnets are attached to the motor axle such that they rotate in front of a chip which can detect their angle. By constantly monitoring the angle of the wheels, the robot can determine how far each wheel has rotated and thus get an estimate of its new position. In this lab, you will be relying on the encoders to move the robot forward in a straight path for a specific distance.

In the following code example, the first function returns the total distance travelled by the left motor, and the second function returns an estimate of the instantaneous velocity of right motor.

```
encoder_get_x (LEFT);
encoder_get_v (RIGHT);
```

Objective 2: Target Practice

Your beacon robot, using a BOM 1.5 board, will be placed at a location on the floor and will turn on its BOM to emit infrared light. Your seeker robot, using a BOM 1.0 board, will be placed a random distance away at a random angle and will have to navigate towards the beacon robot using the BOM. When it gets within a certain distance of the beacon (detected using the rangefinder), your seeker robot will stop while a person removes your beacon robot. After a button press, your robot will drive straight to the location the beacon robot previously occupied and stop. Prizes will be given to the team whose robot stops closest to the location of the beacon robot.

Target Practice Requirements

- Your seeker robot shall be placed between 3 and 5 feet away from the beacon robot
- Your seeker robot shall stop 1 foot away from the beacon robot (within 4 inches)
- Your seeker robot shall wait for a button press at the 1 foot radius
- The beacon robot shall be removed and your seeker robot shall drive to the location of the beacon robot and stop

The first challenge is to locate the beacon robot. To do this, your seeker robot should read its BOM and change the motor speeds to turn the robot towards the beacon. The next challenge is to get the seeker robot to stop once it detects that it is within a certain distance from the beacon robot. The seeker robot should monitor its rangefinder values and drive towards the beacon robot until the rangefinder readings fall below a certain threshold. The final challenge is to get the seeker to drive straight a set distance. This problem is fairly non-trivial, but a basic behavior can be developed to blindly drive the robot forward a set distance by setting both motor speeds to the same value and stopping the robot through trial and error near the beacon robots location. A more sophisticated robot should implement a control loop between the motors to ensure that the robot actually drives straight. This requires that the seeker robot constantly monitor the difference in the distance traveled by each wheel and compensate by changing the motor speed to straighten the path that the robot travels.

Troubleshooting

- If your **robot encoders do not seem to be working correctly**, load the file `encoder_test.c`, compile the code, and download the result to the robot. Connect the robot to a computer and monitor the output from the robot over the serial port using TeraTerm (gtkTerm). The terminal will show the state of the encoders. Check that the encoders behave as expected (spin the wheels manually and check that the encoder readings reflect the fact that the wheels have been turned).
- If your **robot BOM does not appear to be working correctly**, load the file `bom_test.c`, compile the code, and download the result to the robot. Connect the robot to a computer and monitor the output from the robot over the serial port using TeraTerm (gtkTerm). Each of the BOM sensor readings will be displayed, and a histogram will display which sensor returns the strongest reading. Check to make sure that the BOM behaves as expected (if you turn the robot relative to an IR source, the sensor returning the strongest intensity reading should rotate around the 16 BOM positions).

Extra Challenge

To develop a more robust method for navigating towards the beacon robot, you can create a control loop between the motors so that the robot corrects its path. You can also continually re-orient your seeker robot to the beacon robot by reading the BOM while driving towards the beacon robot.

Demo Date: 9/25/09