

2.12 Introduction to Robotics  
*Laboratory No. 2*

Section: \_\_\_\_\_, Date: \_\_\_\_\_

Students Name: \_\_\_\_\_

**Objective**

The objectives of today's laboratory are:

- Download a user C-code to the robot controller.
- Get acquainted with the C language and how it affects the robot.
- Create actions based on sensor inputs.

**Lab Report**

You are expected to turn in the questions/check sheet. Each group turns in one report at the end of lab session.

**SAFETY**

The hardware and motor that you will be using for 2.12 this semester use considerable current. The Robot controller uses up to 1.5A continuously and the motor draws up to 4A of current. These currents are dangerous and please use common sense when operating the equipment. Most importantly, when you are wiring make sure all the power is turned off.

1. Retrieve all the items listed on the last page. At the end of the lab please return all items to where you got them.
2. Mount all components like the example in class and electrically wire all the components like in Lab 1. This time there are two motors. Hook them up to PWM outputs 1 and 5. There is no need to connect to the robot wirelessly. Find something to lift the mobile robots wheels off the ground or workbench.
3. Before you turn on your robot, have the lab instructor check your wiring.
4. Download the Default program to the RC by following the directions on page 22 of the Programming reference guide. The "FRC\_default.hex" file should be on your desktop. If not, you can download it from the Innovation First website.
5. Send commands from the joystick and verify that you can drive the motors in both directions.
  - a. Setup the Dashboard program and connect it to the OI. This will stop updating information on the computer screen. Hook up the motors to PWM signal 13 and 15. Notice how those channels are mixed with the X and Y motions of the joystick. Also note that pushing the joystick forward





9. Make a copy of the FrcCode Folder and rename it so that you don't change the default program. open the workspace "FrcCode.mcw" in your new folder. Immediately "save your project as..." and choose something different than the default setting. You must "make" the current project to create a new .hex file. Load your new .hex file. It should have the same name as your project. Verify that you can download this to the RC. You should work solely with your new project from this point on.
10. Find the "printf" commands. The information in this function is the data that is displayed in the text box after you use the IFI loader. This is a handy debugging tool. Make changes to this command and have it show different information.
11. Switch PWM1 – 4 to be Y, X, Aux, and Wheel from your joystick in port 1. Load the program again and prove that these work. You will have to alter the "printf" statement or look in the dashboard program to see the differences.
12. (Optional) Find the standard place to put a macro and put "#define MY\_NEUTRAL 127". Then find all the instances of 127 in the user\_routines.c.
13. Obtain a rare earth magnet from the laboratory instructor and move it in front of the magnetic switch. You should be able to hear a faint click of the leads coming together. This will be your mine detector. Electronically connect this to the RC using a breadboard and PWM wire. The RC already has a pull-up resistor built in so it makes it easy to connect. If you are confused about this, ask the laboratory instructor. This is important for most of the sensors we will be working with. What is the approximate proximity of the magnetic switch? Does the behavior change with the magnet orientation? If so, how?

Finally, what is the value of having an internal pull-up resistor? What would happen if we did not have this pull-up resistor?

14. Using the "printf" C command or the lights on the OI, make a visual indicator when you "find a mine" (close the magnet switch). Show the instructor when you accomplish this.
15. Determine the electronic behavior of the whisker switches using a multi-meter. Electronically connect the whisker switches to the robot using the breadboard and a PWM cable. Ask the laboratory instructor if you are confused.
16. Using the "printf" C command or the lights on the OI, make a visual indicator when you "hit a wall" (switch goes on). Show the instructor when you accomplish this.

17. Find the variable resistor you used in lab1 and connect it to analog input 1. Find the correct function to read its value and display its value in the “printf” statement. Show the instructor when you accomplish this.

18. This is the fun part. Be clever and use your imagination and stop/start the motors based on various sensor readings, joystick positions, and joystick buttons. You can fix the mirror problem with the default mixing PWM functions. Maybe even pretend that the robot is moving autonomously and if it hits a wall it must turn, go backwards, or anything you can think of. This is what you will be continuing to program at home and in the extra laboratory hours. Show the instructor when you accomplish this. He is very interested in what you can do.

## **Items Needed for the Lab**

- 1x Laptop with IFI Loader, the C-compiler, and Dashboard.
- 1x Pre-made Mobile Robot
- Access to all the Innovation First Documents
- 1x Robot Controller
- 1x Operator Interface
- 1x Tether cable (RED)
- 1x Joystick (the second is optional)
- 1x Battery switch
- 1x 12 V Battery
- 1x Backup Battery
- 2x PWM Amplifier (Victor 884)
- 1x Battery Ground Connector
- 2x Motor Connectors
- 2x Whisker Switches
- 1x AC Adapter
- 4x PWM extension cables (you may need more of these later in the lab)
- 2x DB9 connectors (BLACK)