Working with Motors

Selecting Your Hardware Platform

RobotC supports a variety of different hardware platforms. For these exercises we will configure RobotC to include support for the Tetrix/FTC robots as well as pure LEGO robots. The “FIRST Tech Challenge (NXT)” platform is a superset of the “NXT” platform. Whether your robot is using LEGO motors or the Tetrix 12V motors, you can control them with nearly identical code.

On the Robot / Platform Type menu make sure that you have “FIRST Tech Challenge (NXT)” selected.

Click on the New icon on the RobotC toolbar to start a new program.

Configuring your Motors

You need to tell RobotC how your motors are attached to your NXT. On the Robot menu, open the Motors and Sensors Setup wizard and select the FTC Servo/Motor Ctrl tab.

A. Using LEGO NXT Motors

If your robot is using LEGO NXT motors for the drive wheels, select “No controllers configured”:
Then go to the Motors tab and fill it in according to which ports your drive motors are plugged into. Make sure to enter the names “motorLeft” and “motorRight” correctly so that your robot will go in the correct direction.

Click OK.

Two #pragma statements and a comment line should have been inserted into your program looking something like this:

```c
#pragma config(Motor, motorA, motorLeft, tmotorNormal, PIDControl)
#pragma config(Motor, motorB, motorRight, tmotorNormal, PIDControl)
//**!Code automatically generated by 'ROBOTC' configuration wizard !!*/
```

A #pragma is an instruction to the RobotC compiler to change something about how your program gets compiled. The config() pragma tells RobotC to automatically include the necessary prefix code to configure your program to use the motor and sensor ports you specified in the wizard. You should not modify these lines of code by hand. If the configuration of your robot changes, go back into the Motor and Sensor Setup wizard and change the values there. When you exit the wizard it will update the #pragmas to reflect the changes you made.

Comment lines in RobotC begin with “/\”. Everything that follows a double-slash to the end of the line is ingored by the compiler. The comment inserted by the wizard is a reminder that the #pragmas preceeding it were automatically created for you.

**B. Using Tetrix 12V Motors**

If your robot is using the Tetrix 12V motors for its drive wheels select the configuration that matches how your robot is wired. For example, if you have one motor controller plugged into sensor port 1 and one servo controller plugged into your motor controller, you should choose “Standard Configuration.”
Then select the Motors tab, select “Motor equipped (12V)” in the lowest two Type dropdowns.
Make sure to name them “motorRight” and “motorLeft” according to how your robot is wired.
Also assuming that your motors are pointing away from each other, make sure to reverse one of
them. Don’t worry about picking the right one to reverse right now. We’ll correct it later if
necessary.

Click OK.

Three #pragma statements and a comment line should have been inserted into your program
looking something like this:

```c
#pragma config(Hubs, S1, HTMotor, HTServo, none, none)
#pragma config(Motor, mtr_S1_C1_1, motorRight, tmotorNormal, PIDControl, reversed)
#pragma config(Motor, mtr_S1_C1_2, motorLeft, tmotorNormal, PIDControl)
//*!!Code automatically generated by 'ROBOTC' configuration wizard !!*//
```

A #pragma is an instruction to the RobotC compiler to change something about how your
program gets compiled. The config() pragma tells RobotC to automatically include the
necessary prefix code to configure your program to use the motor and sensor ports you
specifed in the wizard. You should not modify these lines of code by hand. If the configuration of your robot changes, go back into the Motor and Sensor Setup wizard and change the values there. When you exit the wizard it will update the #pragmas to reflect the changes you made.

Comment lines in RobotC begin with “//”. Everything that follows a double-slash to the end of the line is ignored by the compiler. The comment inserted by the wizard is a reminder that the #pragmas preceeding it were automatically created for you.

**Programming Your Motors**

Enter a blank line or two after the code that was inserted by the Motor and Sensor Setup wizard and then type in the following:

```robotc
task main()
{
    motor[motorLeft] = 50;       // Half power
    motor[motorRight] = 50;

    wait1Msec(1000);               // One second

    motor[motorLeft] = 0;         // Stop
    motor[motorRight] = 0;
}
```