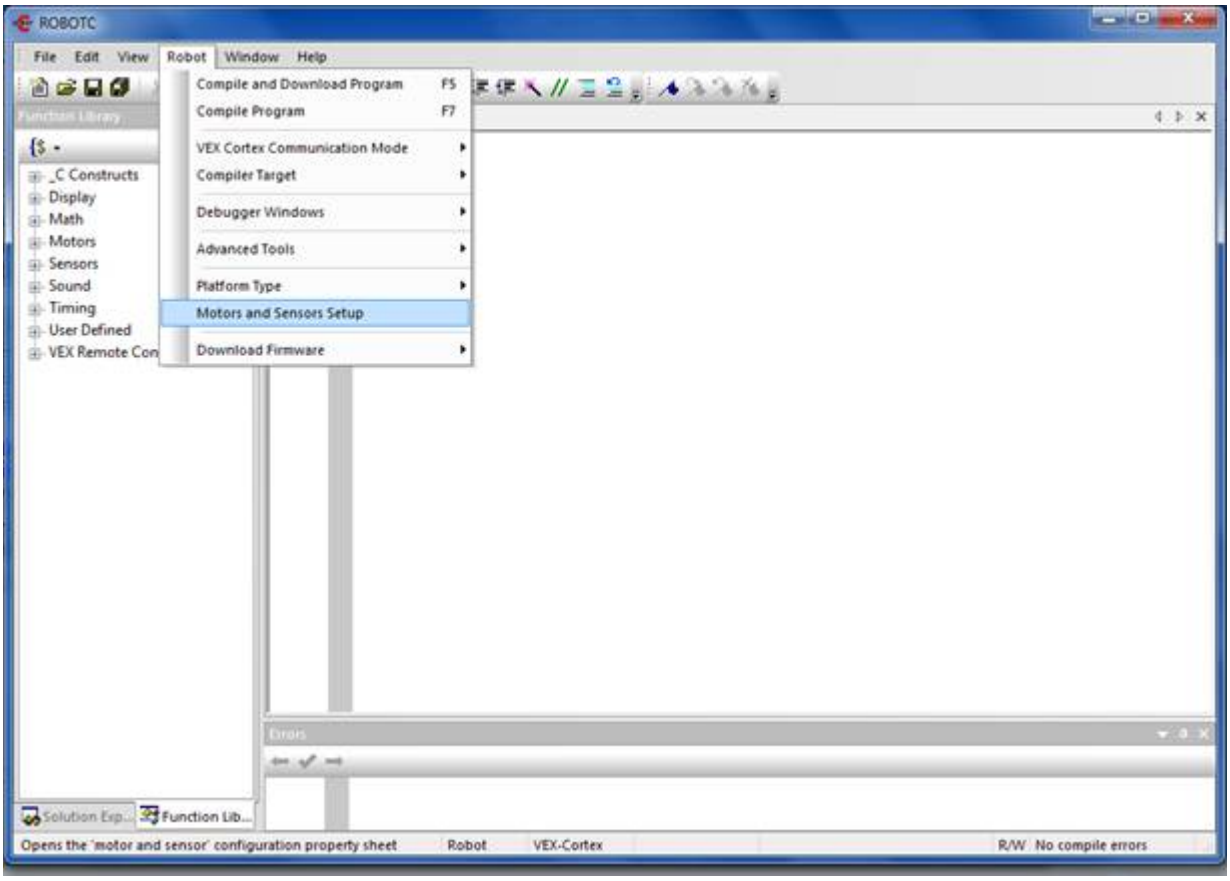
Introduction to Robotics
Writing a Functions for driveStraight and driveTime with RobotC

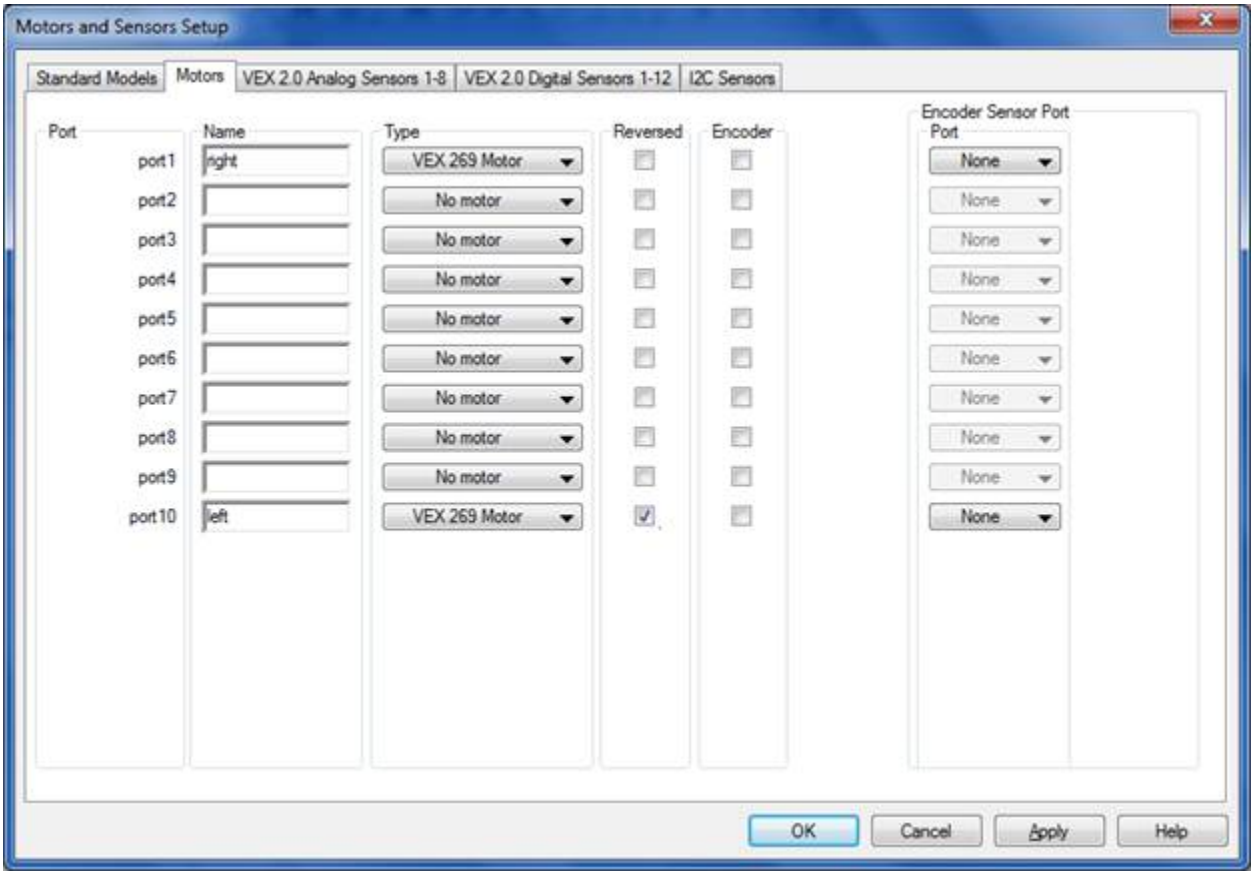
Process:

1. Start RobotC and Select File-New File to create a blank program. Save the Program as "lastname_lab_01.c
2. Select "Robot->Platform Type" from the menu bar and select "VEX 2.0 Cortex"
3. Select "Robot->Motors and Sensors Setup"

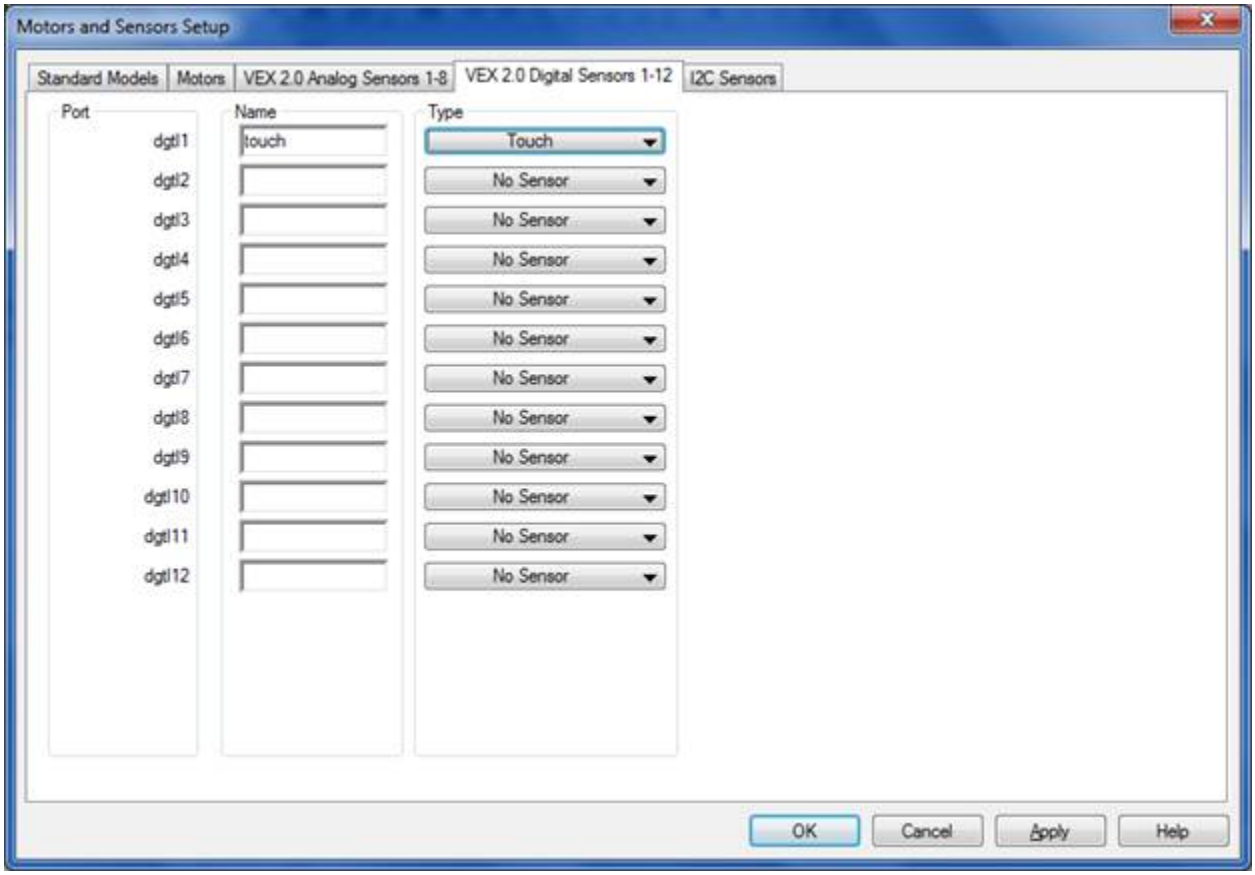


4. Select the Motors Tab and enter “right” for port 1 and set the “Type” to VEX 269 Motor. Enter “left” for port 10 and set the “Type” to VEX 269 or VEX 393 Motor (Depending on your Robot). Check the reverse option for port 10. Then Click “Apply.”

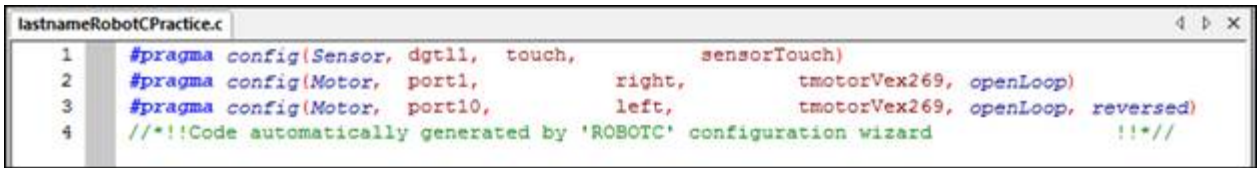
This labels the motors for programming and sets the reverse for the left motor to so the robot wheels will turn in the same direction.



5. Select the “VEX 2.0 Digital Sensors 1-12” and enter “touch” for digtl1 and then select “Touch” for type. Click “Apply” and then click “OK.”



6. Now that we have defined the motors and Sensors – Note that RobotC has entered 4 lines of text in your program. These lines tell the program to define the motor and sensor objects.



7. Add comments starting on line 6 stating your name, class, term and then functions.

lastname_lab_01.c*

```
1  #pragma config(Sensor, dgtl1,  touch,          sensorTouch)
2  #pragma config(Motor,  port1,           right,         tmotorVex269, openLoop)
3  #pragma config(Motor,  port10,          left,          tmotorVex269, openLoop, reversed)
4  //*!!Code automatically generated by 'ROBOTC' configuration wizard               !!*//
5
6  // Firstname Lastname
7  // Intro to Robotics
8  // Term 3 2014
9
10 // Functions
11
12 task main()
13 {
14
15
16
17 }
```

8. In the Functions section, start the definition for the driveStraight() function:

```
11
12   // driveStraight Function
13   void driveStraight(int p) {
14
15   } // end drive straight
16
17
```

9. Inside the driveStraight() function, write the code to start the motors at the power level p: (lines 15 and 15)

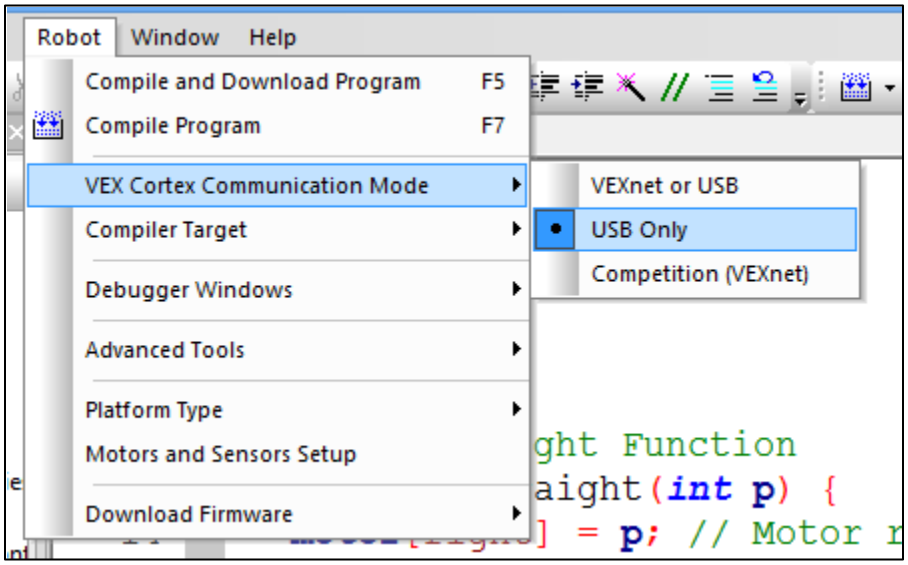
```
10 // Functions
11
12 // driveStraight Function
13 void driveStraight(int p) {
14   motor[right] = p; // Motor right on a p power
15   motor[left] = p;  // Motor left on at p power
16
17 } // end drive straight
18
19
```

10. We now need to test the function. Go to the task main and write these commands to test the function.

```
10  // Functions
11
12  // driveStraight Function
13  void driveStraight(int p) {
14    motor[right] = p; // Motor right on a p power
15    motor[left] = p;  // Motor left on at p power
16
17  } // end drive straight
18
19
20
21  task main()
22  {
23    // Test the function
24    driveStraight(75); // Drive at 75 power
25    wait1Msec(1000);   // Wait 1 second
26    driveStraight(0);  // Turn Motors off (0 power)
27
28  }
```

11. Save and compile the code (F7) to check for errors.

12. Select Robot-VEX Cortex Communication Mode to set the mode to USB Only.



13. Plug in your robot and press F5 to download the code. Test the code on the Robot. Make adjustments to code or setup as needed.

14. We now need to write the driveTime() function. This will allow us to set the power and time for the motors. Go back to the functions section of your code and start the function definition:

```
10  // Functions
11
12  // driveStraight Function
13  void driveStraight(int p) {
14    motor[right] = p; // Motor right on a p power
15    motor[left] = p;  // Motor left on at p power
16
17  } // end drive straight
18
19  // driveTime Function
20  void driveTime(int p, int t) {
21
22  } // end driveTime
23
24
```

15. We can use the driveStraight function inside of driveTime. Note that 'p' is the parameter for power and 't' is the parameter for time in milliseconds.

```
18
19  // driveTime Function
20  void driveTime(int p, int t) {
21    driveStraight(p); // Drive at p power
22    wait1Msec(t);     // Wait for t milliseconds
23    driveStraight(0); // Stop the motors
24
25  } // end driveTime
26
```

16. We now need to test this function. Go to the task main and modify the code to test the driveTime function.

```
18
19  // driveTime Function
20  void driveTime(int p, int t) {
21    driveStraight(p); // Drive at p power
22    wait1Msec(t);     // Wait for t milliseconds
23    driveStraight(0); // Stop the motors
24
25  } // end driveTime
26
27
28  task main()
29  {
30    // Test the function
31    driveTime(75, 1000); // Drive at 75 power for 1 second
32
33  }
```

17. Download and test on your robot. Make adjustments as needed

18. To complete the Lab 01, write functions for:

pointTurn(int p)

turnTime(int p, int t)

touchStop()

rectangle(); // drive a rectangle

avoidObstacle()