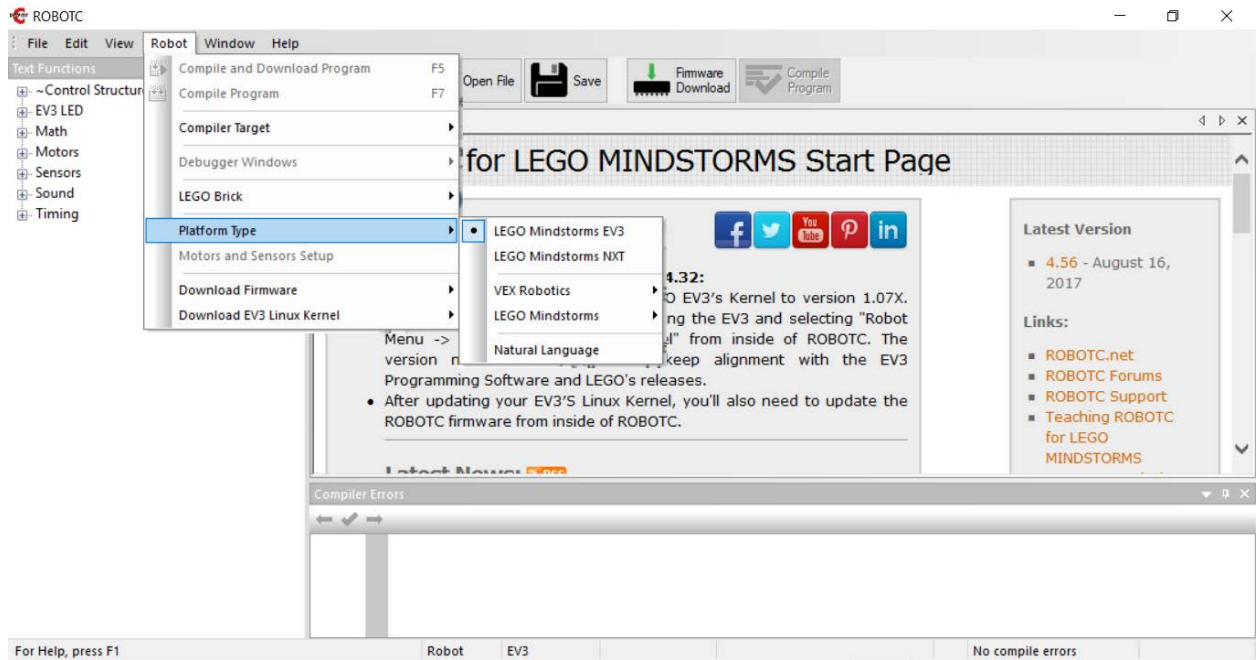


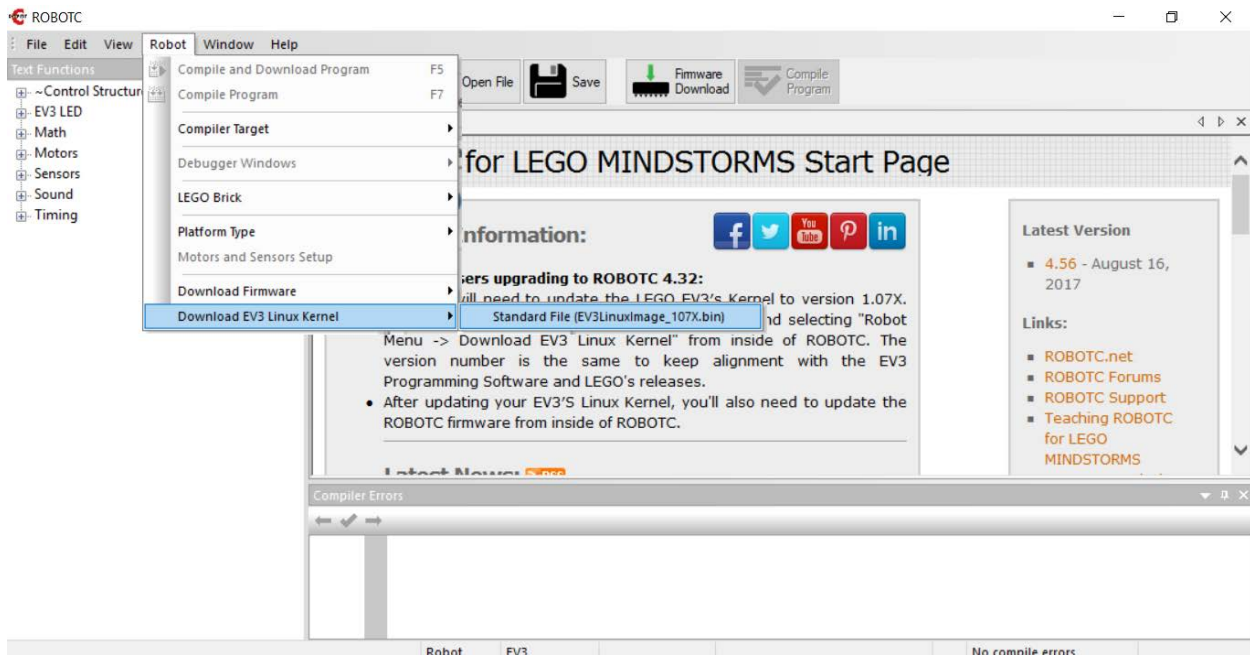
ROBOT LAB

(Please note that documentation can be found at:
<http://help.robotc.net/WebHelpMindstorms/index.htm>)

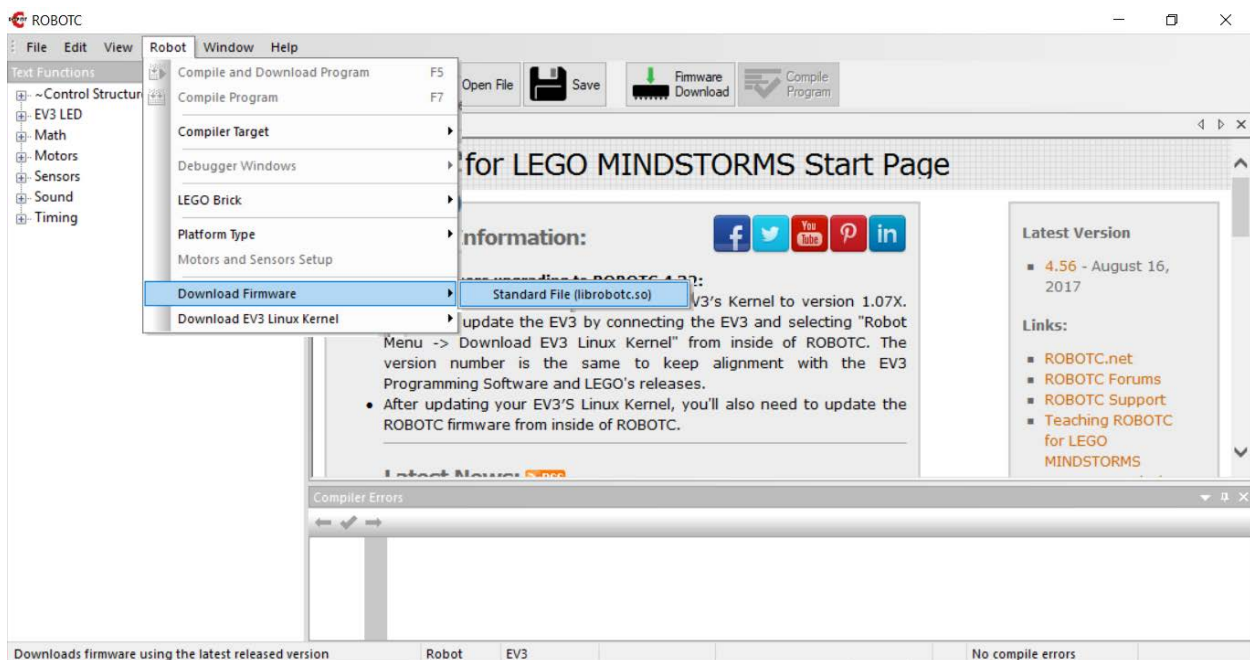
1. From the Start Menu or the Desktop, open **ROBOTC 4.X** (RobotC for Lego Mindstorms 4). Make sure that the Platform Type is set to EV3 as shown below.



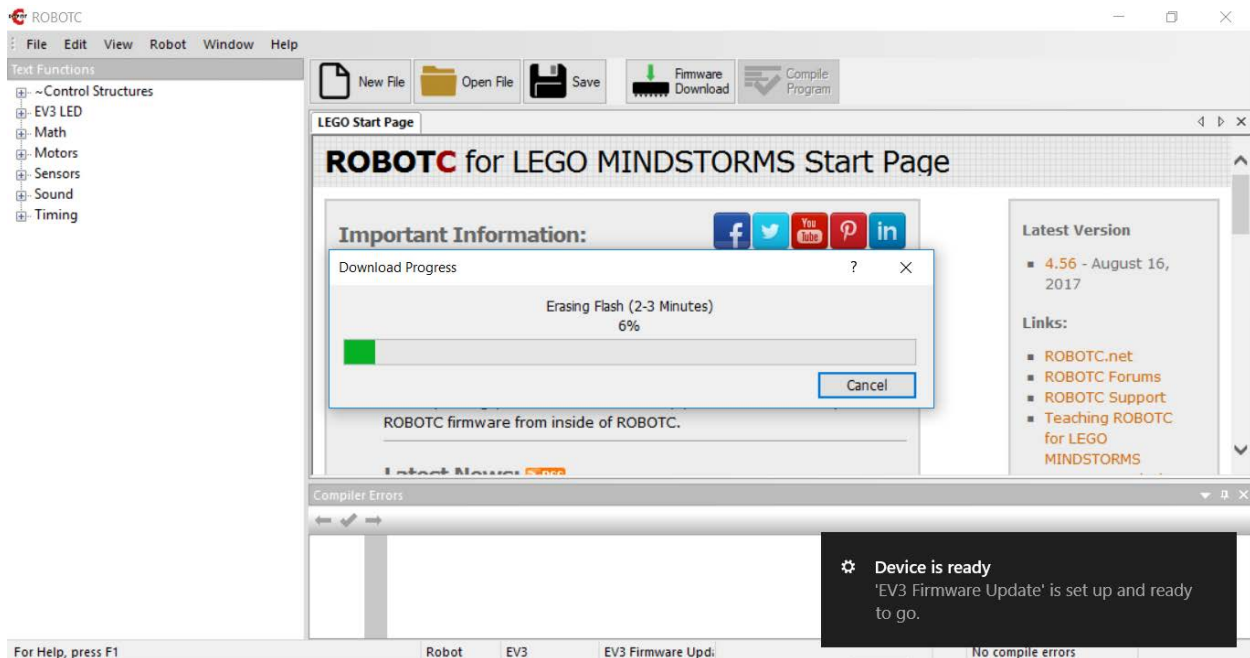
2. Hopefully the LINUX kernel has been updated for you. If you are prompted to do update, see the screen shot below.



3. Download the firmware onto the robot, by choosing *Robot* and then *Download Firmware*



4. You might see something like the screen below.



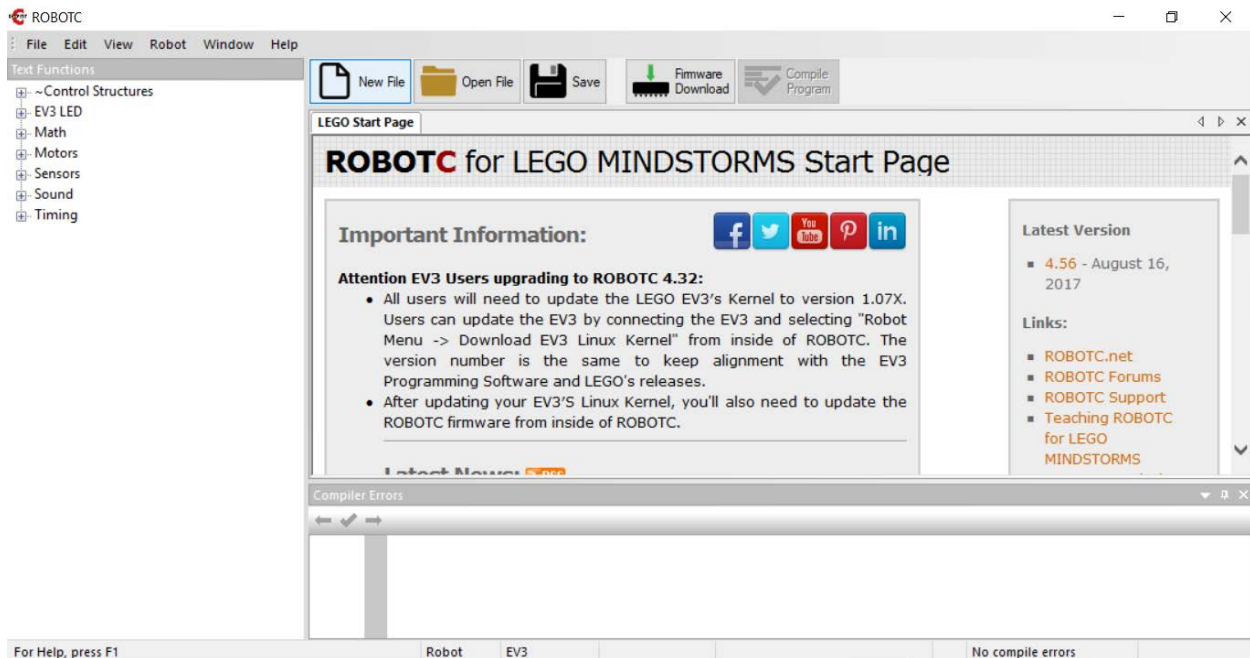
5. A sample program has been downloaded onto your robot. You can run it by using the right gray arrow side key on the EV3 (once to get to the files), then use the down gray key to get to demo, then press the middle button to run.

The next set of instruction can be seen in more detail from the ROBOTC help pages:

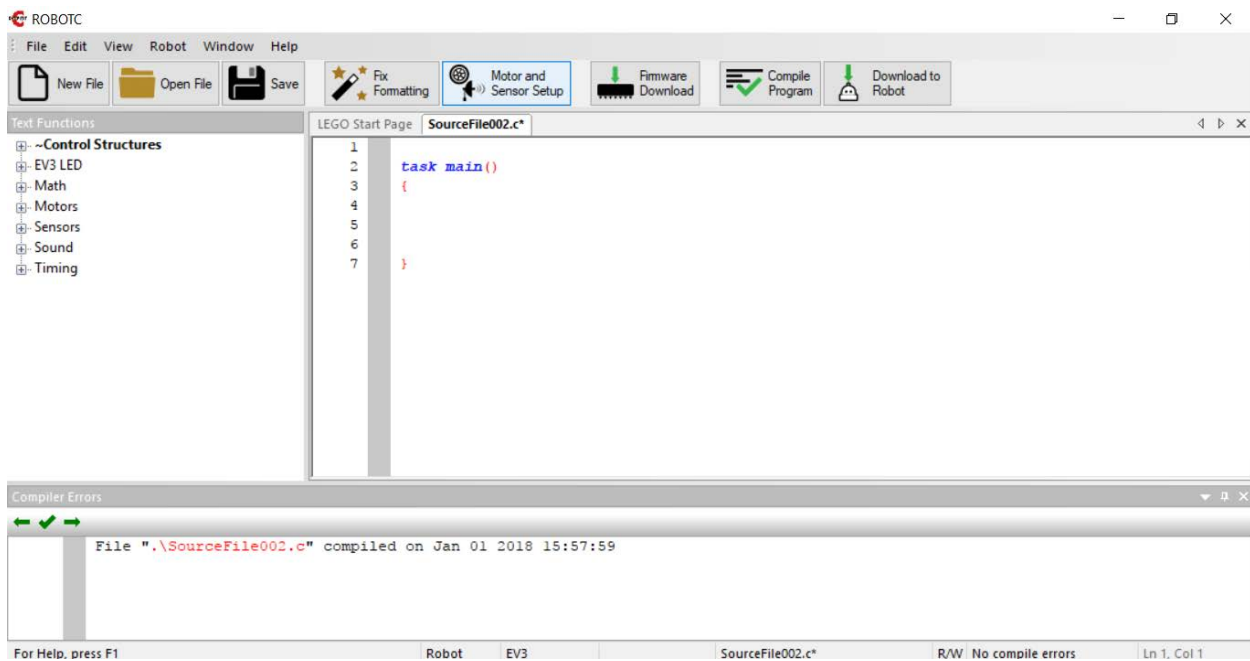
http://help.robotc.net/WebHelpMindstorms/index.htm#Resources/topics/Getting_Started/Motors_and_Sensors_Setup.htm

For a description of any of the sensors, go to shop.lego.com and search for that particular sensor.

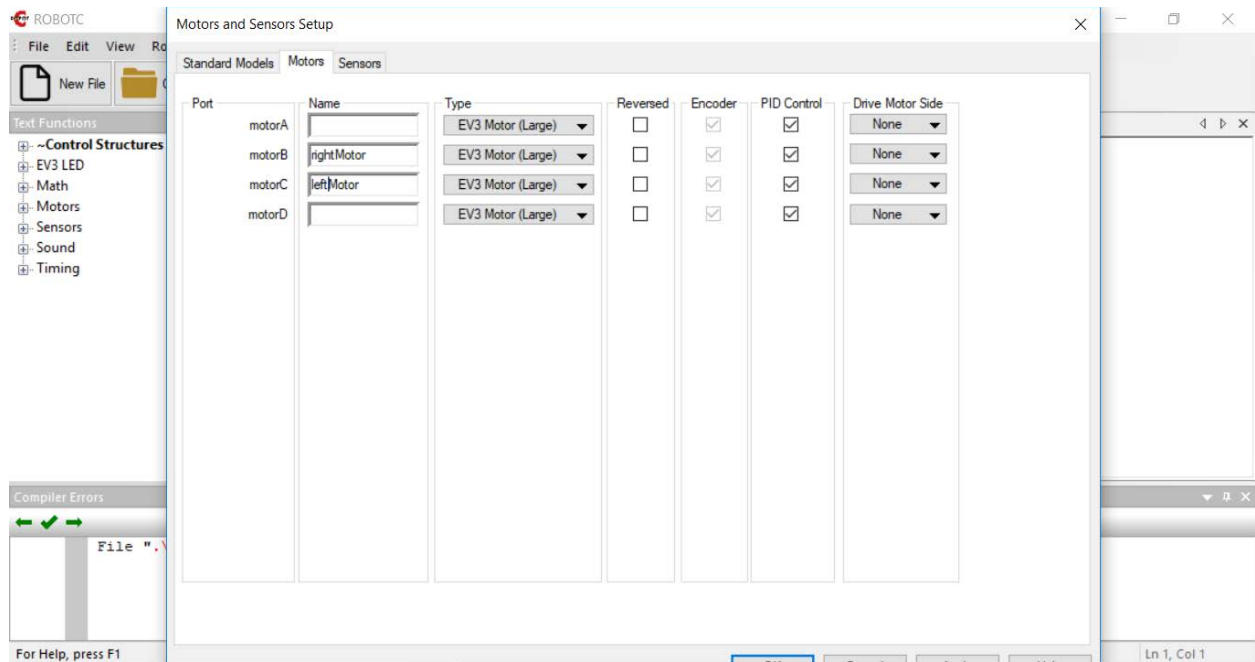
6. To set up the Motor and Sensors, a file must be open. Click on the *New File* button so that the *Motor and Sensor* menus will open. A new empty file with the header line of *task main()* is opened. See the set of screenshots below.



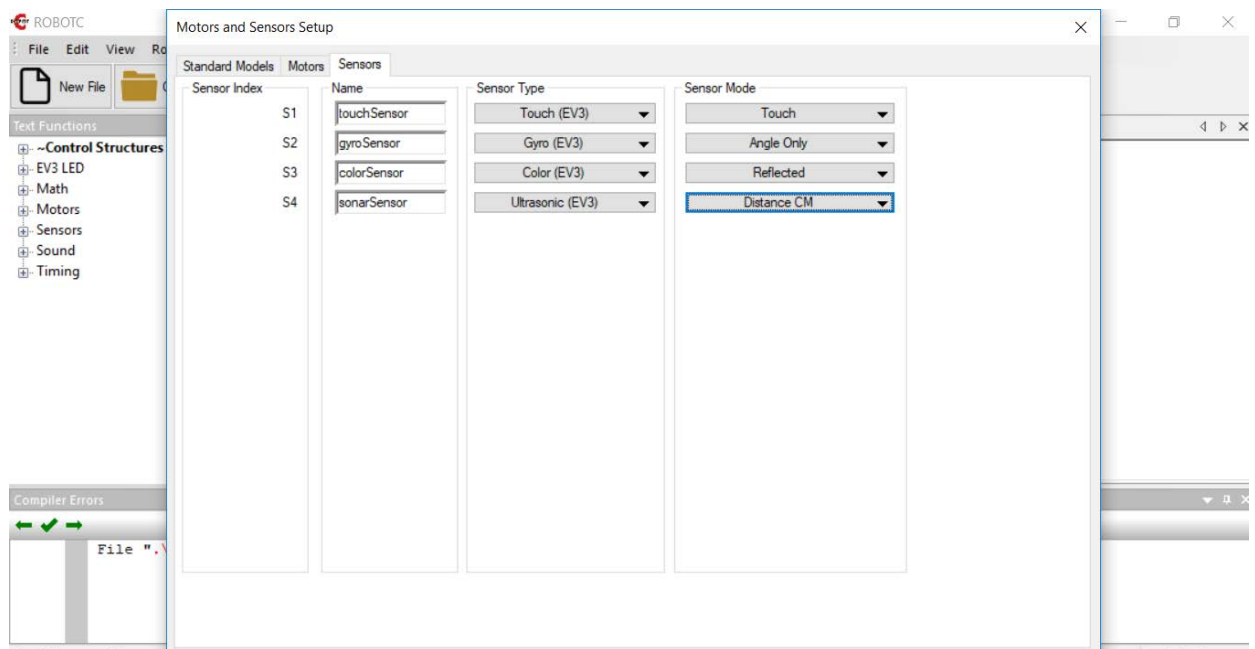
- Now click on *Motor and Sensor Setup* (you can also get to this from the menu (*Robot/Motor and Sensor Setup*))



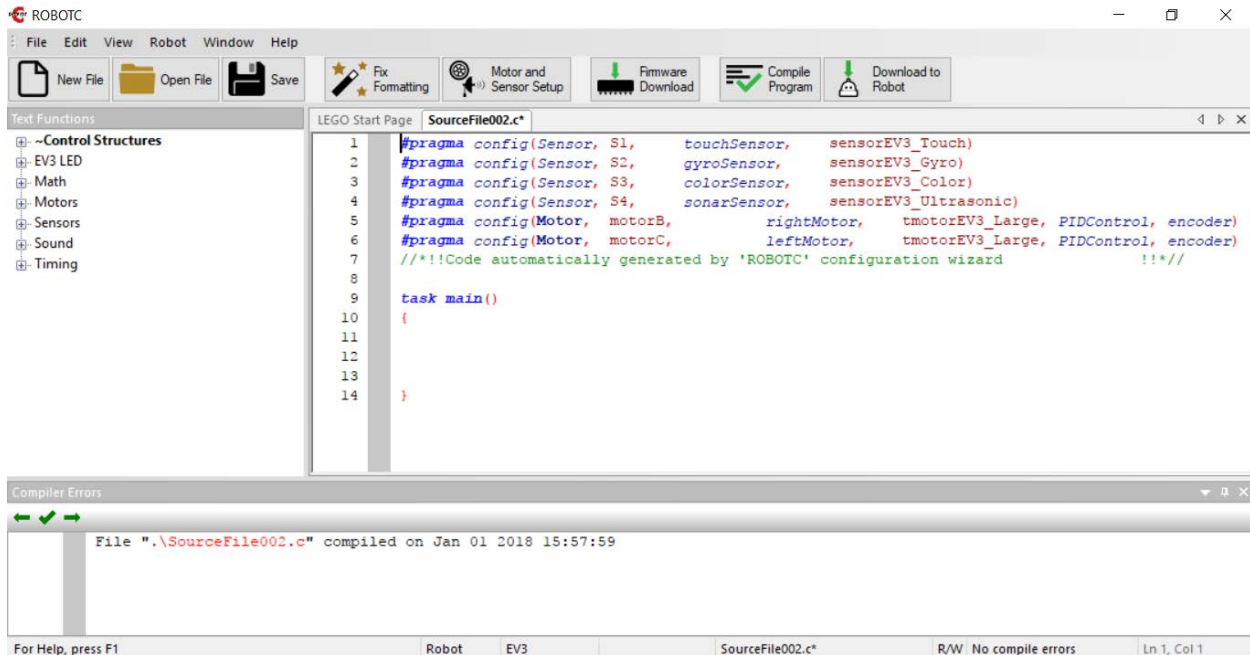
- Click on *Motors*, and give names to the motors. Check your EV3 to make sure that the motors are connected properly. I only have B and C connected so I typed names into B and C and my screen looks as follows:



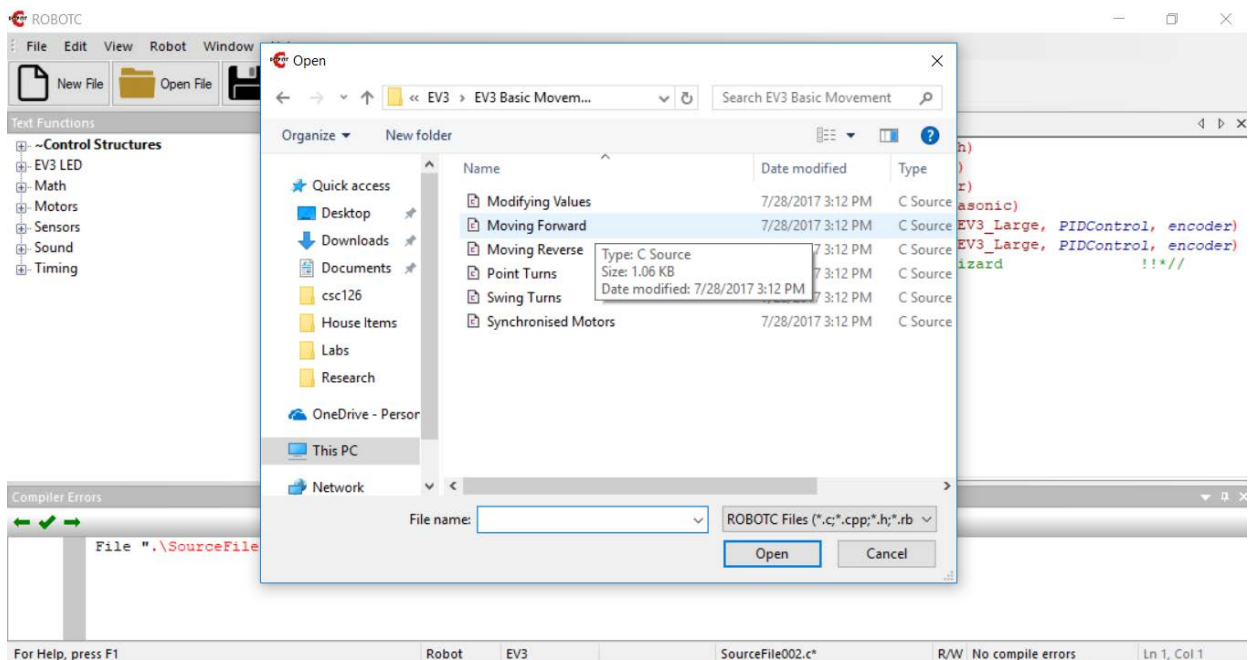
- Click on the *Sensors* tab and give names to the sensors. Check these connections on your EV3 as well. Choose the correct Sensor Type for each of the sensors. The last column will be filled in automatically.



- Click *Apply*, and code will be generated for you as in the screen shot below.



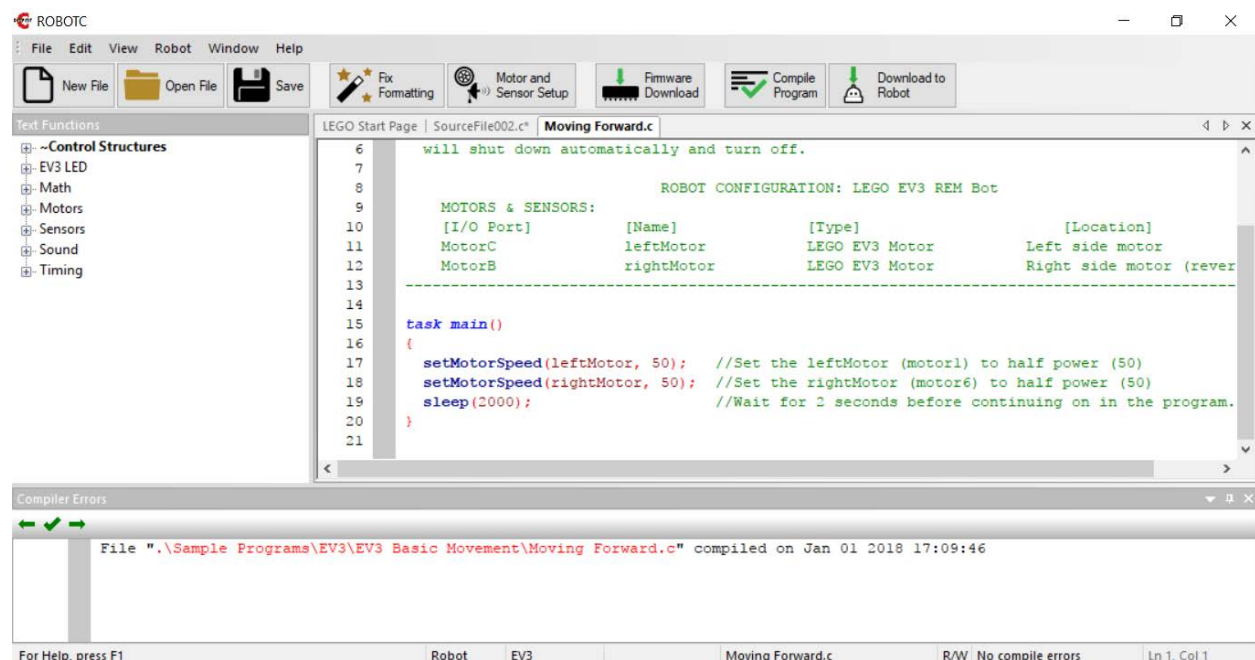
11. Go to *File/Open Sample Programs* to open a sample program (Example: *EV3 Basic Movement Moving Forward*)



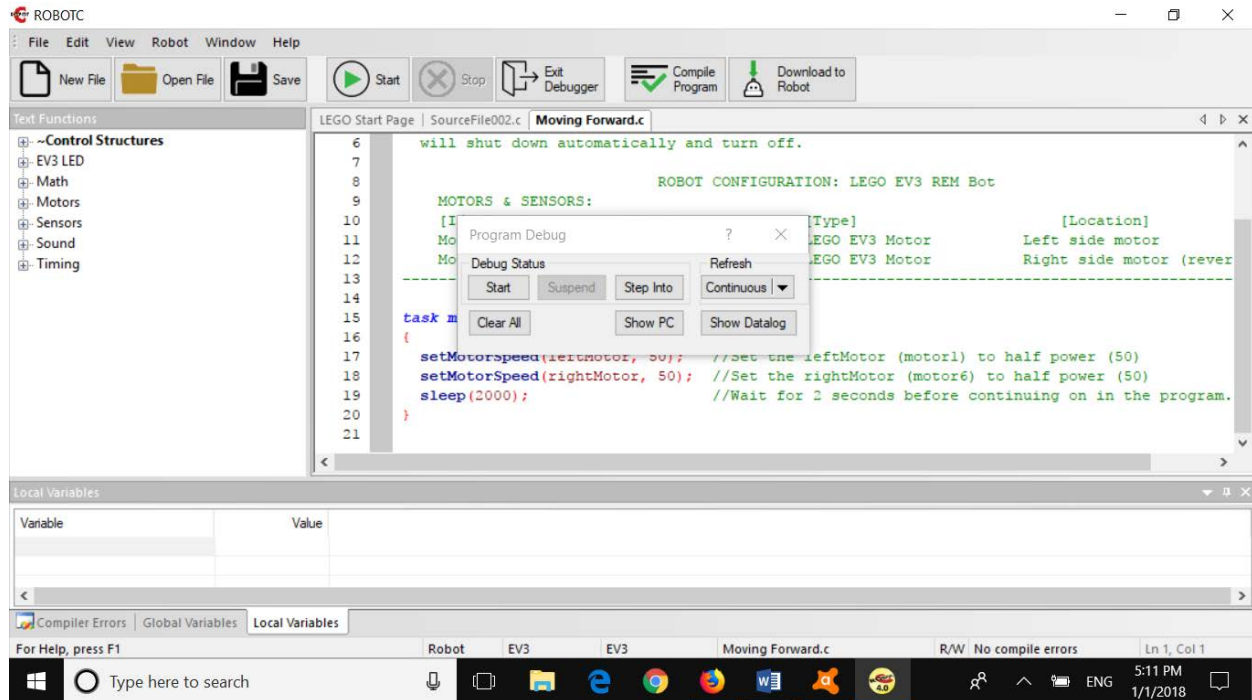
12. Look at the file, as in the screen shot below. The file contains function calls to functions that are specific to the EV3.

RobotC is a C based language, so C constructs (if, while etc) can be used. For specific RobotC functions, you can see the Function library. The left-hand side of the interface contains the Function Library panel has functions organized under type. These functions can be dragged and dropped into the text editor.

The code below calls the function `setMotorSpeed` that takes two parameters, a name of a motor and a speed (0 is off, 100 is highest speed, -100 is highest speed backward). The code also calls a function `sleep`, that takes one parameter, an integer that corresponds to the number of milliseconds that the robot stays in its state.

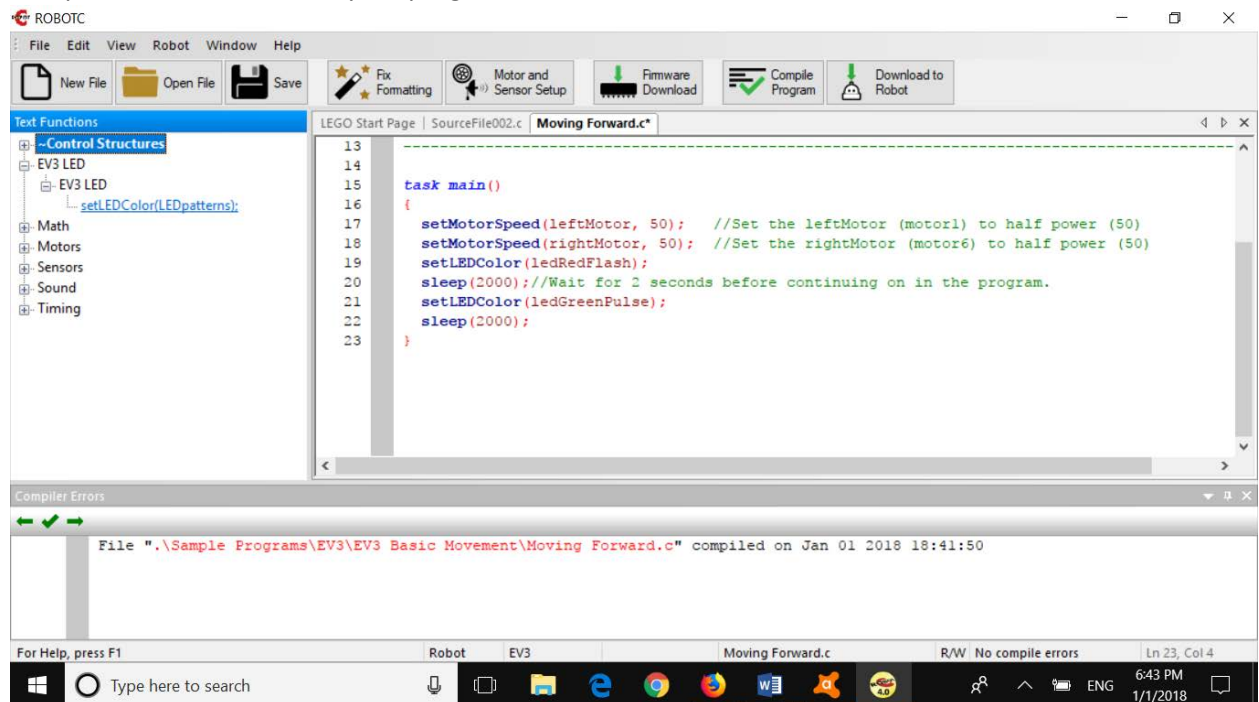


13. Click on the Compile Program button, the Download to Robot button, and then click the run button in the small window to execute your code.



- Let us see how to explore other functions. In the Function Library Pane, click on EV3LED and click on setLEDCoior. Let you mouse hover over it to get information. Then add this function to your sample program, by dragging it into the text editor. An example is below.

Compile, download and run your program.



15. Here are some other examples of functions that can be used with the EV3. Try them out (and check the boxes as you go along!) as follows:

☐ displayTextLine

For example: `displayTextLine(3, "Hi");` displays the text string Hi! on line 3 of the display. There are 8 lines on the display. **Write a program that displays a message on one of the lines of the display.**

☐ getTouchValue

For example: `if (getTouchValue(S1) == 1)` This checks the value of the touch sensor (0 not touched, 1 touched). The parameter is the name of the sensor that was set during the sensor setup.

Using an if statement, display one message if the touch sensor is true, another message if it is false.

☐ playSound

For example: `playSound(soundBeepBeep);`

You can see the names of all the built-in sounds at:

http://help.robotc.net/WebHelpMindstorms/index.htm#Resources/topics/LEGO_EV3/ROBOTC/Sounds/playSound.htm

Using a loop, play any sound of your choice 10 times.

☐ If the sonar sensor (ultrasonic) is set up using S4 as above we can do the following:

`int currentDistance; //declare a variable`

`//then give currentDistance the value of the sensor`

`currentDistance = SensorValue[S4];`

(currentDistance < 10) will return true if the ultrasonic sensor is facing an object within 10 centimeters.

Use a while loop to move forward until your robot is within 5 cm of an object.

□ resetGyro(nDeviceIndex) and getGyroDegrees(nDeviceIndex) Resets the Gyro sensor to 0 so that the current angle is 0. getGyroDegrees gets the current degree (angle) based on the previous reset.

For example:

Type in the following. It uses the resetGyro and getGyroDegrees and displayTextLine

```
task main()
{
    int x;
    resetGyro(S2);
    x = getGyroDegrees(S2);

    displayTextLine(3,"%d",x); //should display a 0
    sleep(4000);
}
```

For example:

```
//Keep looping until the gyro sensor reads greater
//than 90 degrees from its current position
repeatUntil(getGyroDegrees(S2) > 90)
//To fine tune this, the speed of the motors can be set to high
until this function returns 70, then after 70, the motors can be
set to a low speed)
```

16. OK! Now we can combine all the functions above to write a useful RobotC program. Assume that your robot is starting at one side the room! You wish to get across the room, but there are some obstacles in your way!
- Start going straight across the room (use a while or repeat loop)!
 - If you are within 30 cm of an obstacle (use the ultrasonic sensor in the condition of the while or repeat loop) turn right 90 degrees (use the gyro sensor for this)!
 - Once robot has turned 90 degrees, go straight for 2 seconds, turn back to robot's original direction! If the obstacle is still there, turn back 90 degrees and go straight for 2 seconds again, until you can get around the obstacle.
 - Continue going across the room!
 - E.C. Note that there might be multiple obstacles in your way, as you go across the room!

