

# How to Use the OPEN-ROBOT TCP Class Library

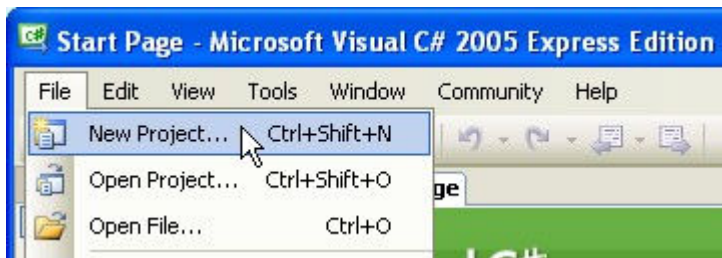
## Download and Install Microsoft® C# Express:

<http://www.microsoft.com/express/vcsharp/>

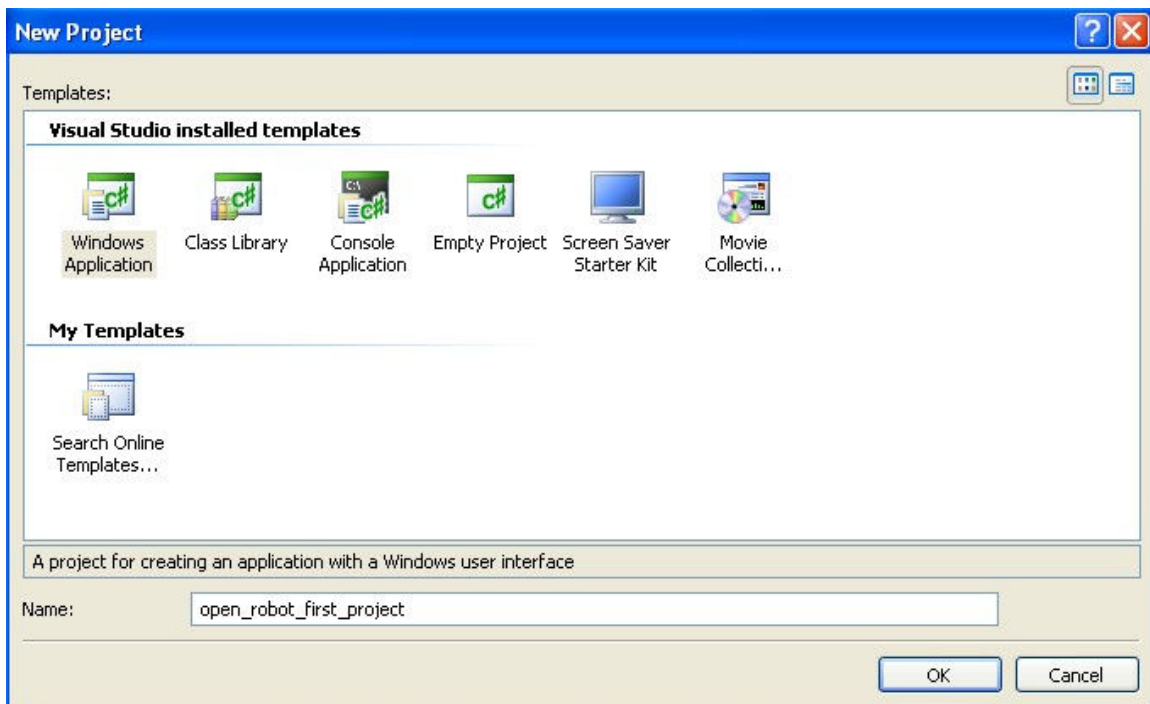
*Note:* I have only tested this using Visual C# Express 2005.

Run C# Express and Create a New Project:

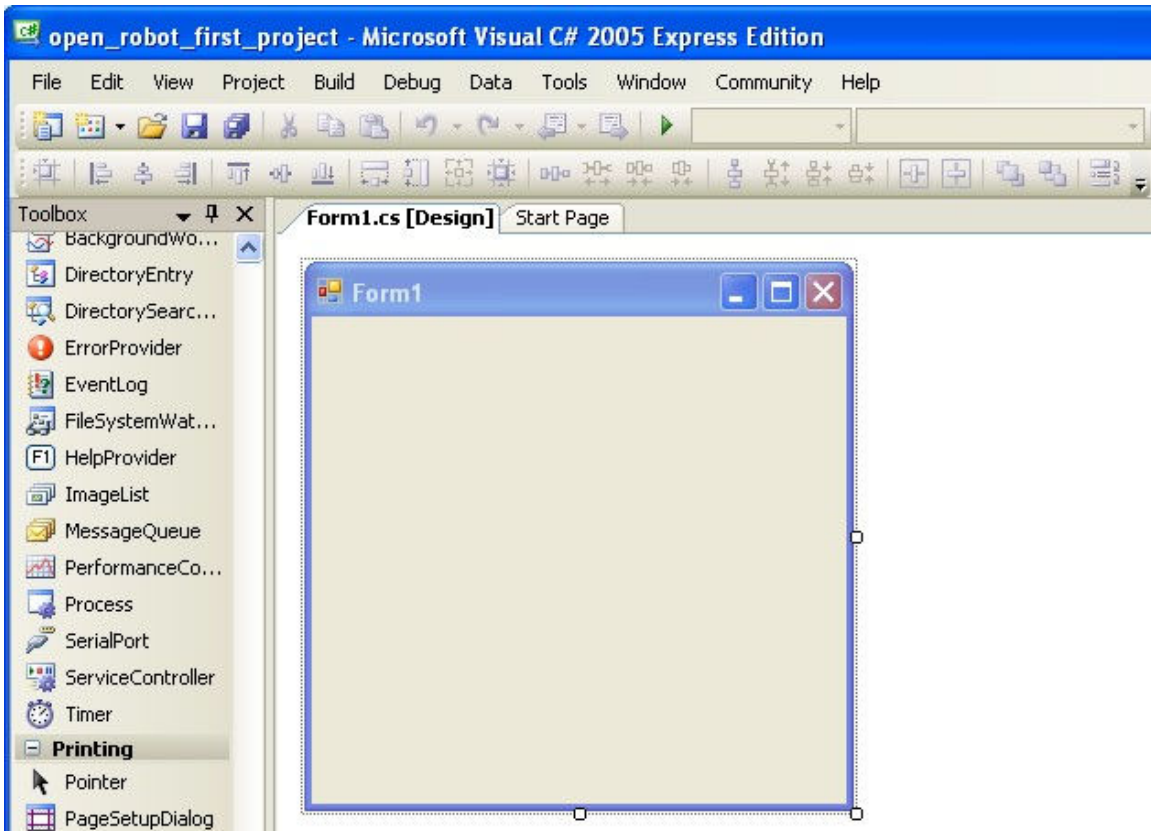
After installing C# Express, go ahead and run the application so that we can create a new project.



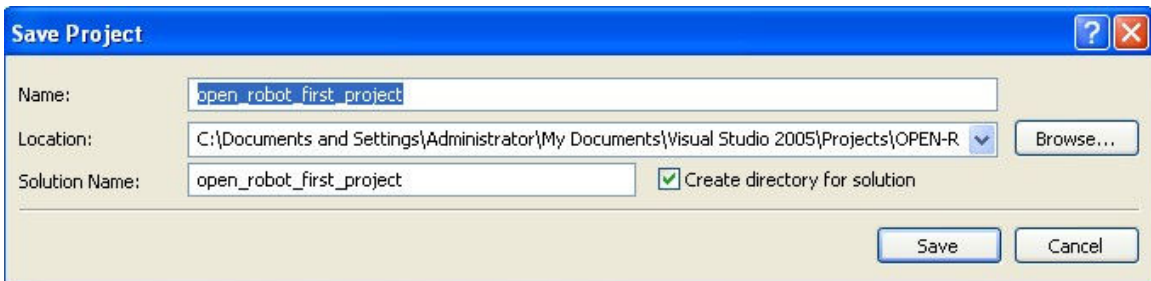
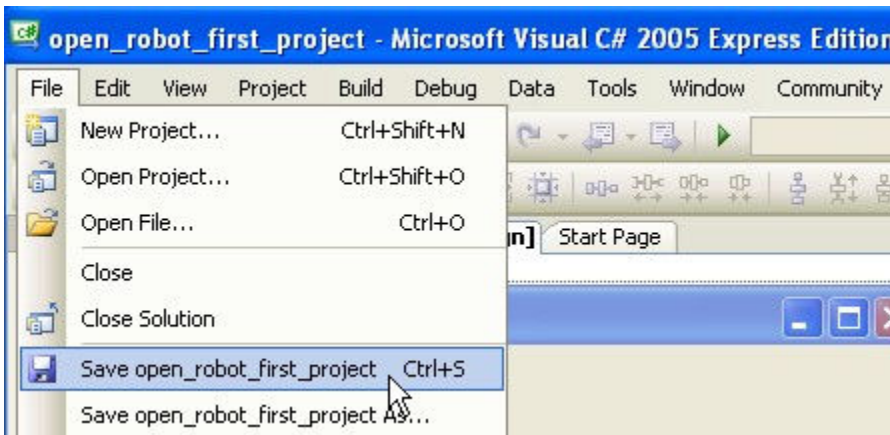
Enter a name for the new project. I'll be using **open\_robot\_first\_project**.



After clicking the **OK** button a new project will be created.



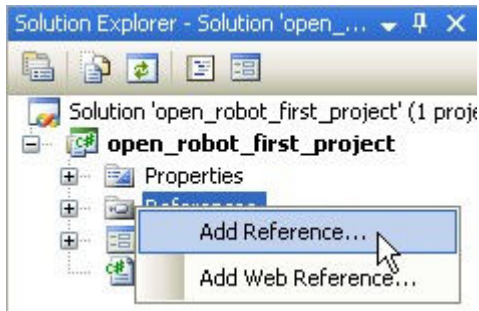
Be sure to save your newly created project so that a folder is created.



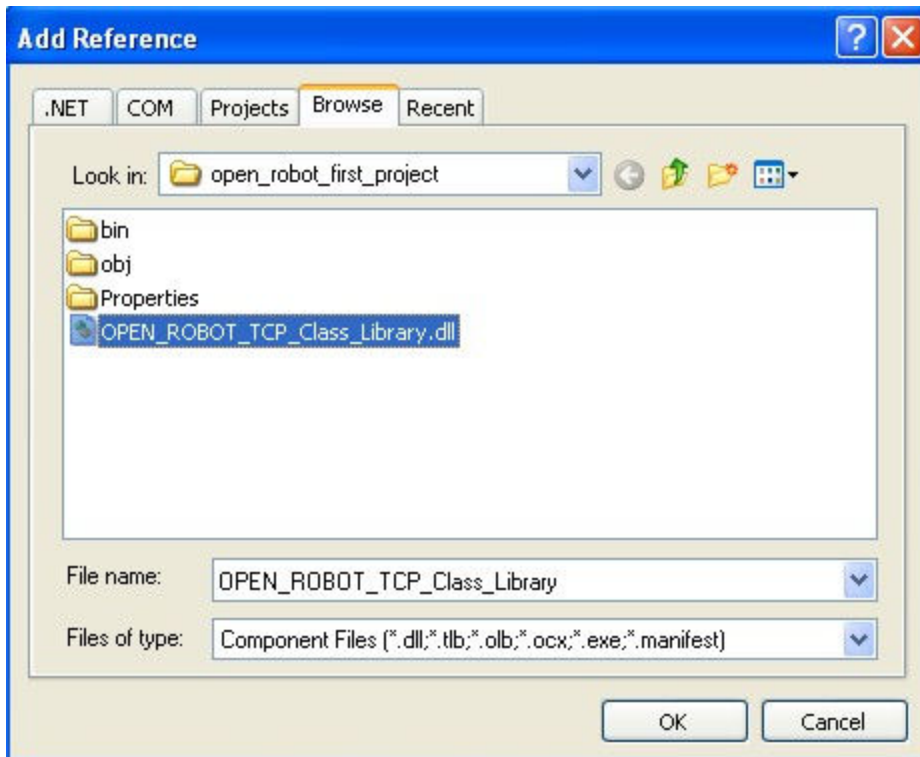
Now you need to copy the **OPEN\_ROBOT\_TCP\_Class\_Library.dll** to the folder of your newly created project so that we can later add it to the project.



Now go back to your C# project and right click on **References** under the **Solution Explorer** and select **Add Reference**.



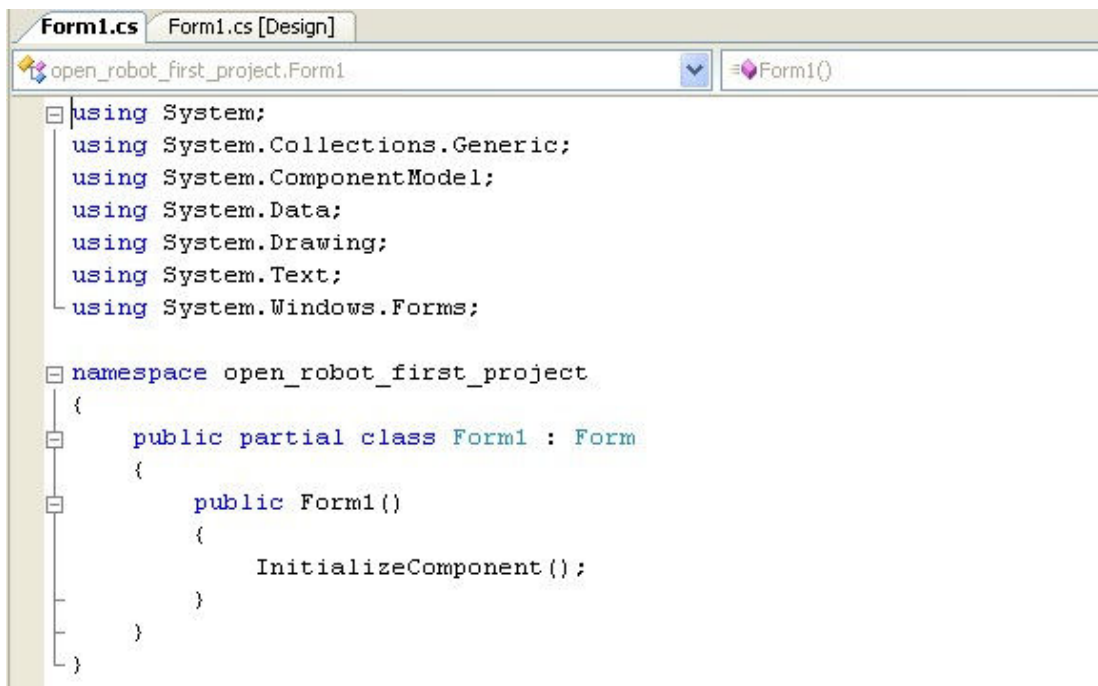
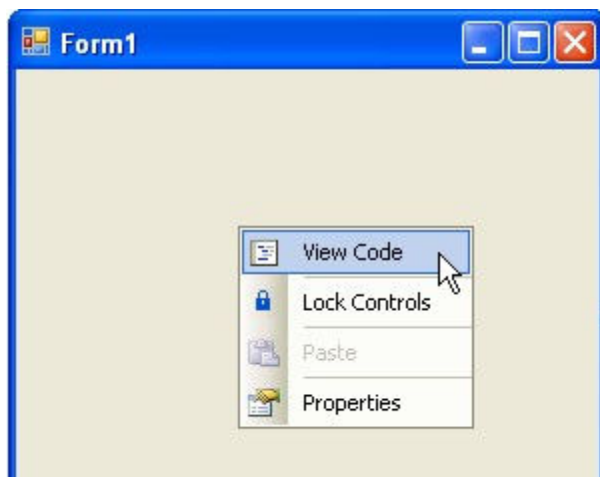
Click on the Browse tab and find the **OPEN\_ROBOT\_TCP\_Class\_Library.dll** file.



Once you've selected the file click the **OK** button. The class library will now show up under your project's References in the **Solution Explorer** window.



Right click on **Form1** and select **View Code**.



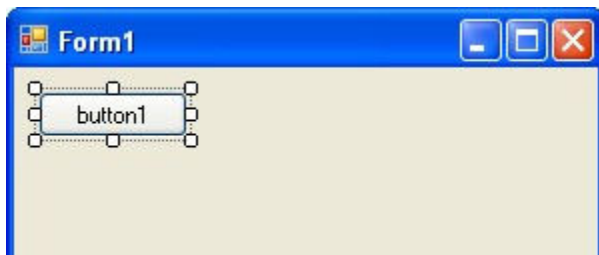
Now you need to create an instance of the **OPEN\_ROBOT\_TCP\_Class\_Library**.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

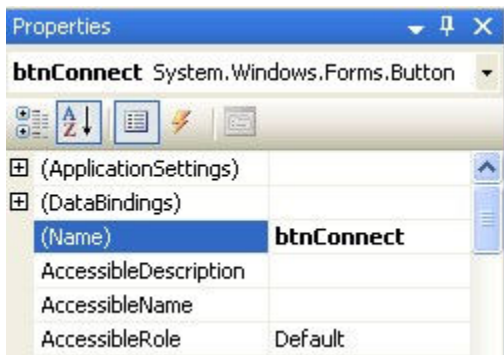
namespace open_robot_first_project
{
    public partial class Form1 : Form
    {
        OPEN_ROBOT_TCP_Class_Library.OPEN_ROBOT_TCP_Functions myOPEN_ROBOT = new OPEN_ROBOT_TCP_Class_Library.OPEN_ROBOT_TCP_Functions();

        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

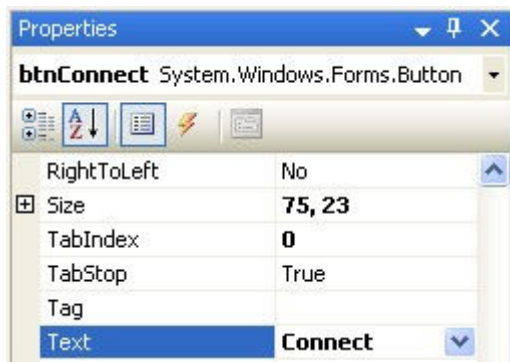
Go back to the Form1 Design window by clicking on the **Form1.cs [Design]** tab. Drag a Button control over to Form1.



We will use this button to create a connection with the OPEN-ROBOT. Right click on **button1** and select Properties. Edit the **Name** and change it to **btnConnect**.



Scroll down and change the **Text** property to **Connect**. Now the button will display Connect.



Double click the Connect button to create and display button click event code.

```
private void btnConnect_Click(object sender, EventArgs e)
{
}
}
```

This button click event code will be executed every time a user clicks the **Connect** button. This is where we will open a TCP/Socket connection with OPEN-ROBOT. Always use a Try/Catch to ensure that we handle any errors that may occur.

```
private void btnConnect_Click(object sender, EventArgs e)
{
    try
    {
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return;
    }
}
```

Use the **OpenSocketConnection** method to open a socket connection. Enter your robot's IP Address and be sure to use 10001 as the Port. If the call to **OpenSocketConnection** generates an exception it will be handled and so we need to let the user know what happened and then return. If the connection is successful then we should let the user know.

```

private void btnConnect_Click(object sender, EventArgs e)
{
    try
    {
        myOPEN_ROBOT.OpenSocketConnection("192.168.0.200",10001);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return;
    }

    MessageBox.Show("Connected to Robot!");
}

```

Now go back to the **Form1 Design** window and create a **Close** button. Name this button as **btnClose** and be sure to set the Text property to **Close**. We will use the **CloseSocketConnection** method. Now add the following code to the button click event code. This code will close the open socket connection.

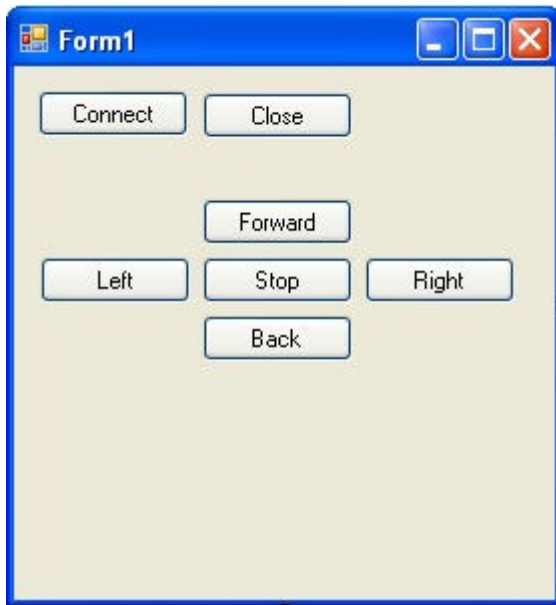
```

private void btnClose_Click(object sender, EventArgs e)
{
    try
    {
        myOPEN_ROBOT.CloseSocketConnection();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return;
    }

    MessageBox.Show("Connection to Robot is Closed!");
}

```

Again go back to the **Form1 Design** window but this time create five buttons with the following names: **Forward**, **Backward**, **Stop**, **Left**, and **Right**.



Then add the following code to the corresponding button click event code.

```
private void btnForward_Click(object sender, EventArgs e)
{
    try
    {
        myOPEN_ROBOT.SetOpenLoopVelocity(25,25,500,0);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return;
    }
}
```

For the **Forward** button click event we will use the **SetOpenLoopVelocity** method to make the robot move forward.

```
private void btnBack_Click(object sender, EventArgs e)
{
    try
    {
        myOPEN_ROBOT.SetOpenLoopVelocity(-25, -25, 500, 0);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return;
    }
}
```

For the **Back** button click event we will use the **SetOpenLoopVelocity** method to make the robot move backward.

```
private void btnLeft_Click(object sender, EventArgs e)
{
    try
    {
        myOPEN_ROBOT.SetOpenLoopVelocity(-25, 25, 500, 0);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return;
    }
}
```

For the **Left** button click event we will use the **SetOpenLoopVelocity** method to make the robot spin left.

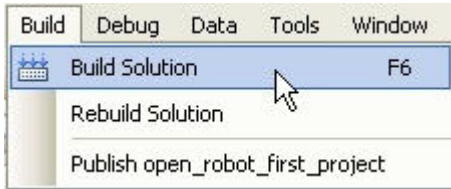
```
private void btnRight_Click(object sender, EventArgs e)
{
    try
    {
        myOPEN_ROBOT.SetOpenLoopVelocity(25, -25, 500, 0);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return;
    }
}
```

For the **Right** button click event we will use the **Set\_Open\_Loop\_Velocity** method to make the robot spin right.

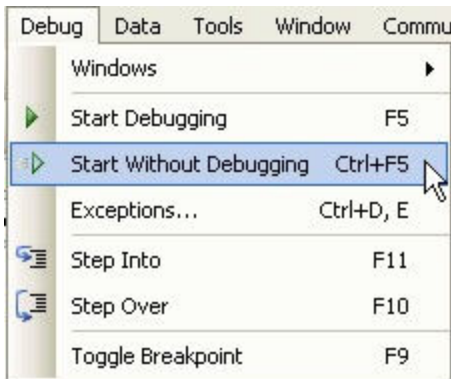
```
private void btnStop_Click(object sender, EventArgs e)
{
    try
    {
        myOPEN_ROBOT.StopRobot(500,0);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return;
    }
}
```

For the **Stop** button click event we will use the **StopRobot** method to make the robot halt or stop immediately.

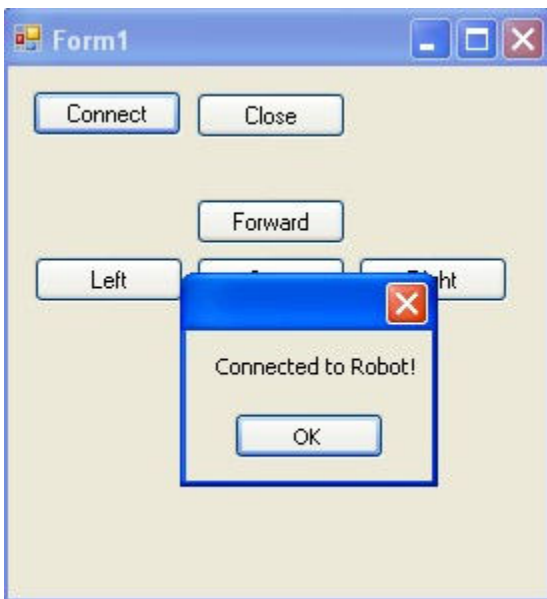
Now you should be able to compile and run your program to test it out. To compile/build your program select **Build Solution** from the **Build** menu.



To run the program, select **Start Without Debugging** from the **Debug** menu.

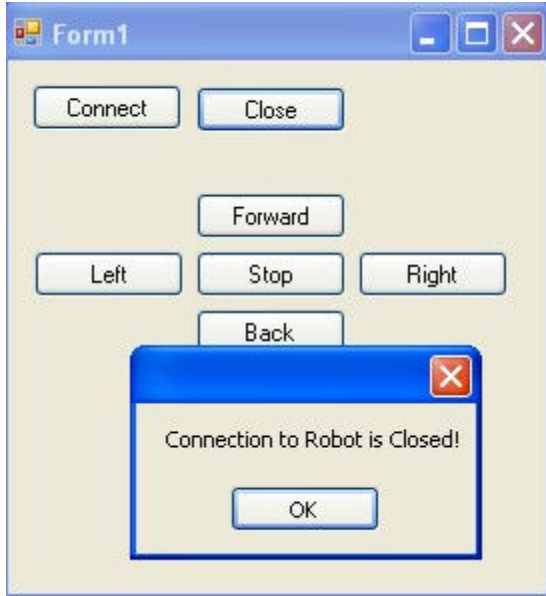


Your project should now be running and you can try connecting to your robot, but be sure to turn your robot on first.



Once you are connected the **Forward**, **Backward**, **Left**, **Right**, and **Stop** buttons can be used.

When you are done be sure to close the socket connection.



If you have any questions please send them to [abe@abotics.com](mailto:abe@abotics.com)