Basic Motor Control

Contents

Introduction

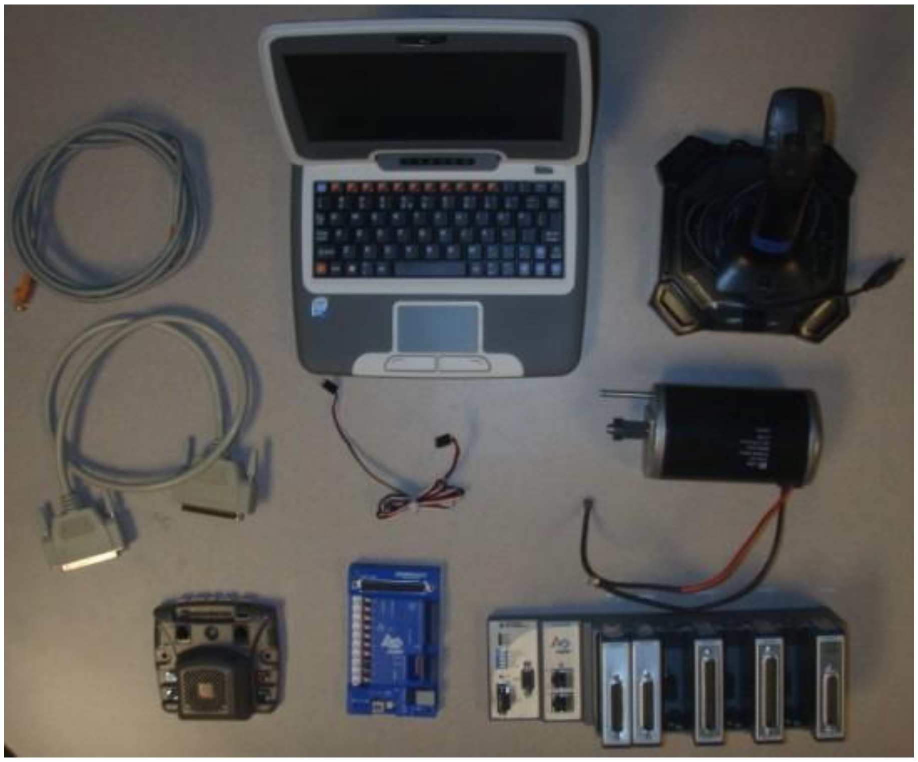
This tutorial demonstrates how to get the cRIO-FRC up and running a motor. It walks through the hardware setup and the programming necessary to control a motor with a joystick, as well as covering basics such as setting up the FRC Control System, necessary network configuration, and setting up an FRC Robot Project.

Getting Started

If you haven't already, go ahead and install the software that came in your kit. This will install a FRC specific version of LabVIEW along with the WPI Robotics Library. You will also need to have your cRIO imaged and your Driver Station firmware up to date. You can refer to the Training Material and Resources page and the *LabVIEW Robotics Programming Guide for the FIRST Robotics Competition* for information and instruction.

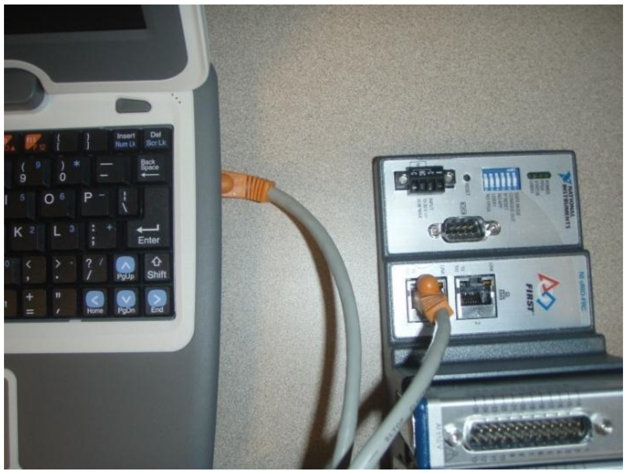
The following is a list of the hardware we are going to be using in this tutorial:

- FRC2 cRIO and the three included modules
- Driver Station/Netbook
- Motor
- Jaguar motor controller
- Digital Side Car
- 12 V battery
- SH37 68 pin cable
- R/C cable
- Two Ethernet Cables
- 14 gauge wire
- Optional: USB Joystick



Configuring the Network Connection

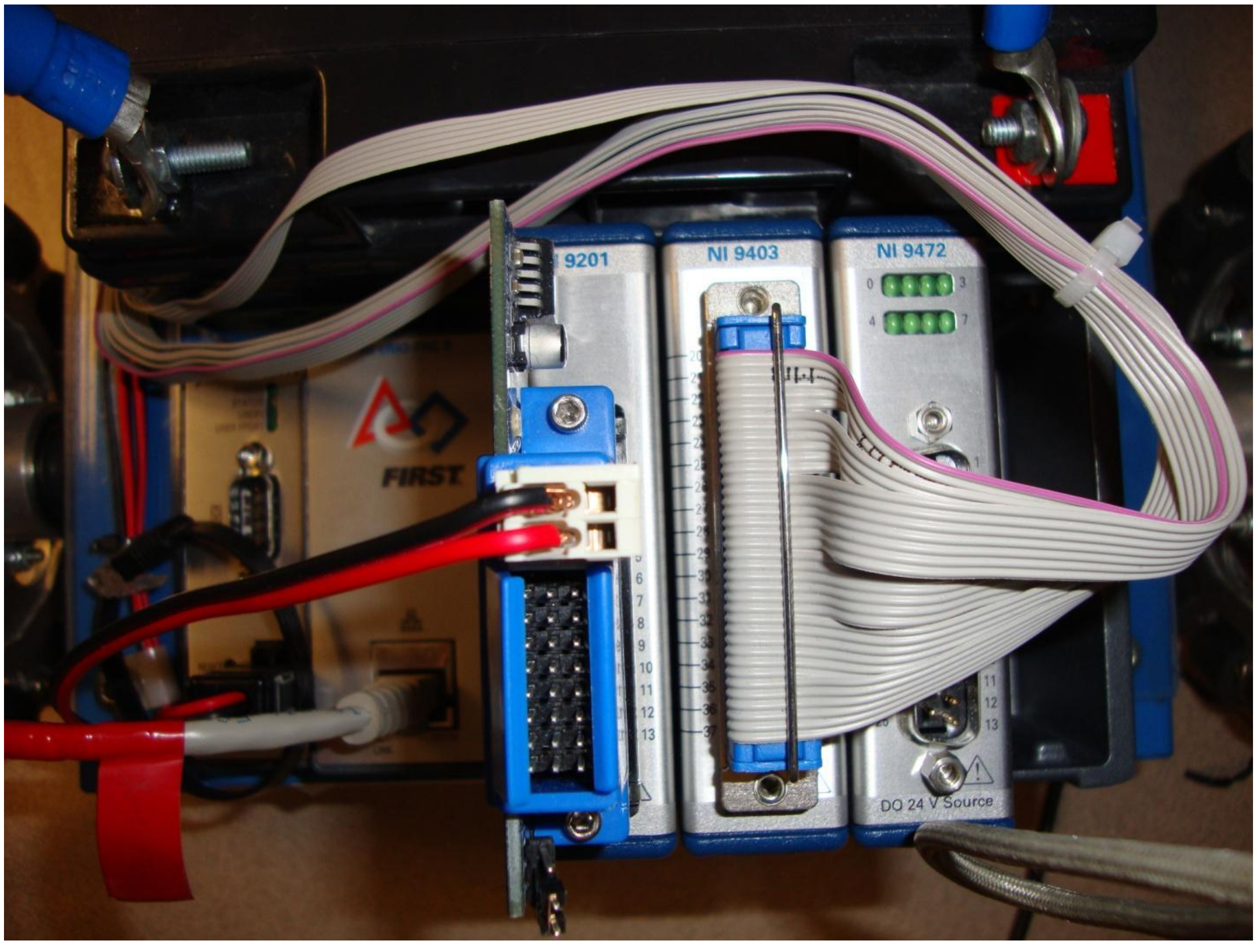
Connect the cRIO to the driver station using an Ethernet cable. This could also be done with a wireless router without changing any code so that the robot is un-tethered.



Power the driver station and configure it using your team number (See the FRC Driver Station Tutorial). This will set up the IP address of the driver station and the computer.

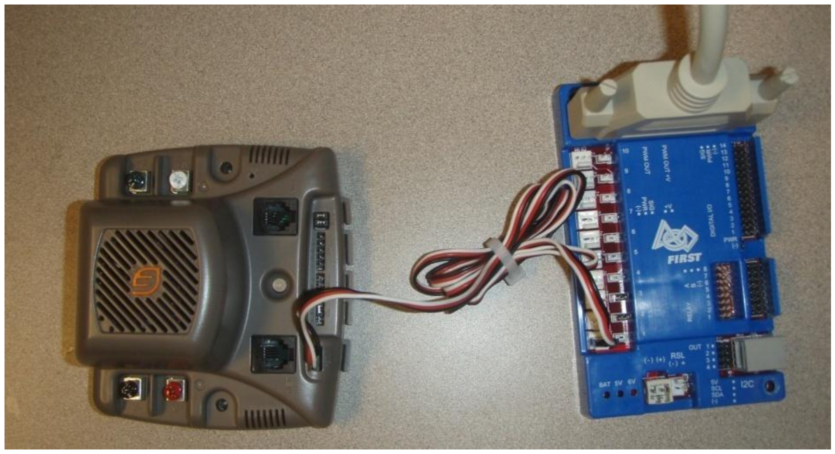
Connecting the Hardware

Remember to keep the modules in the same slots as they were shipped in or the modules will not work properly. Connect the Digital Side Car to the NI 9403 digital input output module in slot 2 using the SH37 cable.



The Digital Side Car is a breakout board that provides several signal interfaces, one of which is pulse width modulation (PWM). This application requires one PWM channel to send commands to our motor.

When connecting the Jaguar motor controller, the controller first needs to be connected to PWM Channel 1 on the Digital Side Car using an R/C cable. Make sure that the black wire goes to ground (-), the red wire goes to power (PWR) and the white wire goes to signal (SIG).

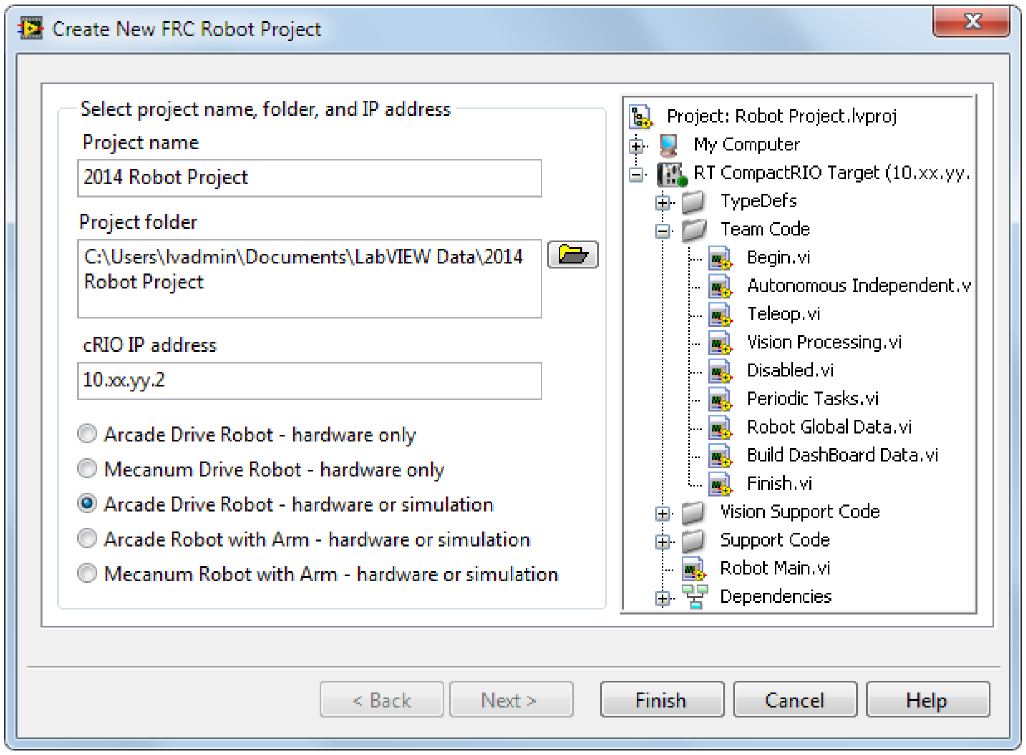


Next connect the motor to the M+/M- terminals on the Jaguar controller. Red is positive and black is negative. Now connect the Jaguar motor controller's V+/V- terminals to the 12 volt battery through the power distribution board. The controller fan should also be connected to the V+/V- terminals to prevent overheating.   Before continuing make sure the fan is on and that the motor is secured so that it does not move while running.

Finally, power the Digital Side Car by connecting it to the 12V battery through the power distribution board.

Creating the LabVIEW Project

Now that the hardware is configured the next step is to write a software VI to control the motor. Open up LabVIEW FRC and create a new FRC cRIO Robot Project. Name the project, set the save path, and enter the cRIO-FRC's IP address (10.0.0.2), then click **Finish**.



Create New FRC Robot Project

Select project name, folder, and IP address

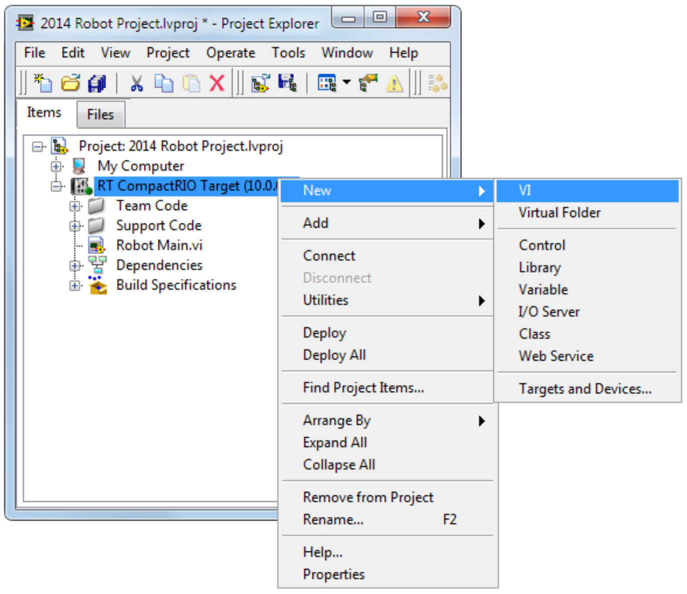
Project name: 2014 Robot Project

Project folder: C:\Users\lvadmin\Documents\LabVIEW Data\2014 Robot Project

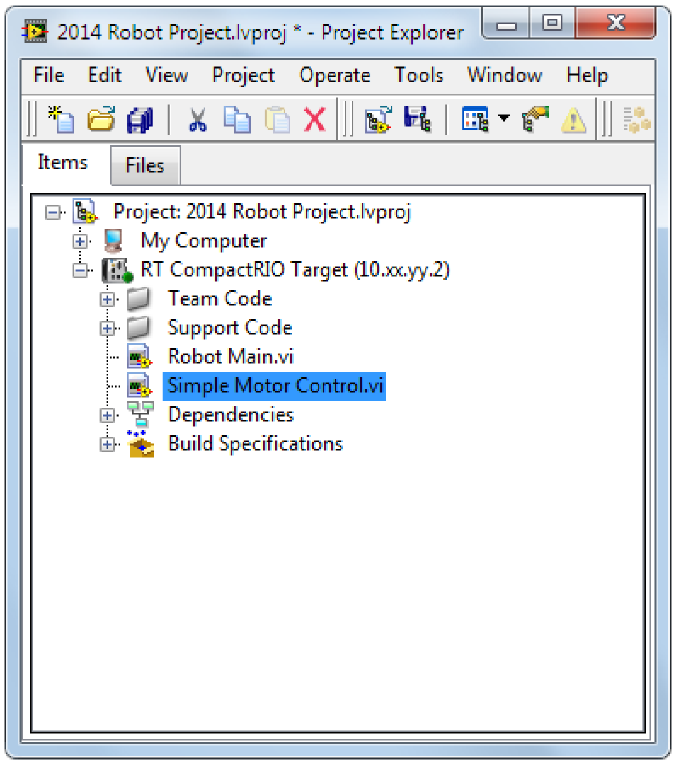
cRIO IP address: 10.xx.yy.2

- ○ Arcade Drive Robot - hardware only
- ○ Mecanum Drive Robot - hardware only
- ● Arcade Drive Robot - hardware or simulation
- ○ Arcade Robot with Arm - hardware or simulation
- ○ Mecanum Robot with Arm - hardware or simulation

In the project window, right-click on **RT CompactRIO Target (10.0.0.2)** and select **New»VI**



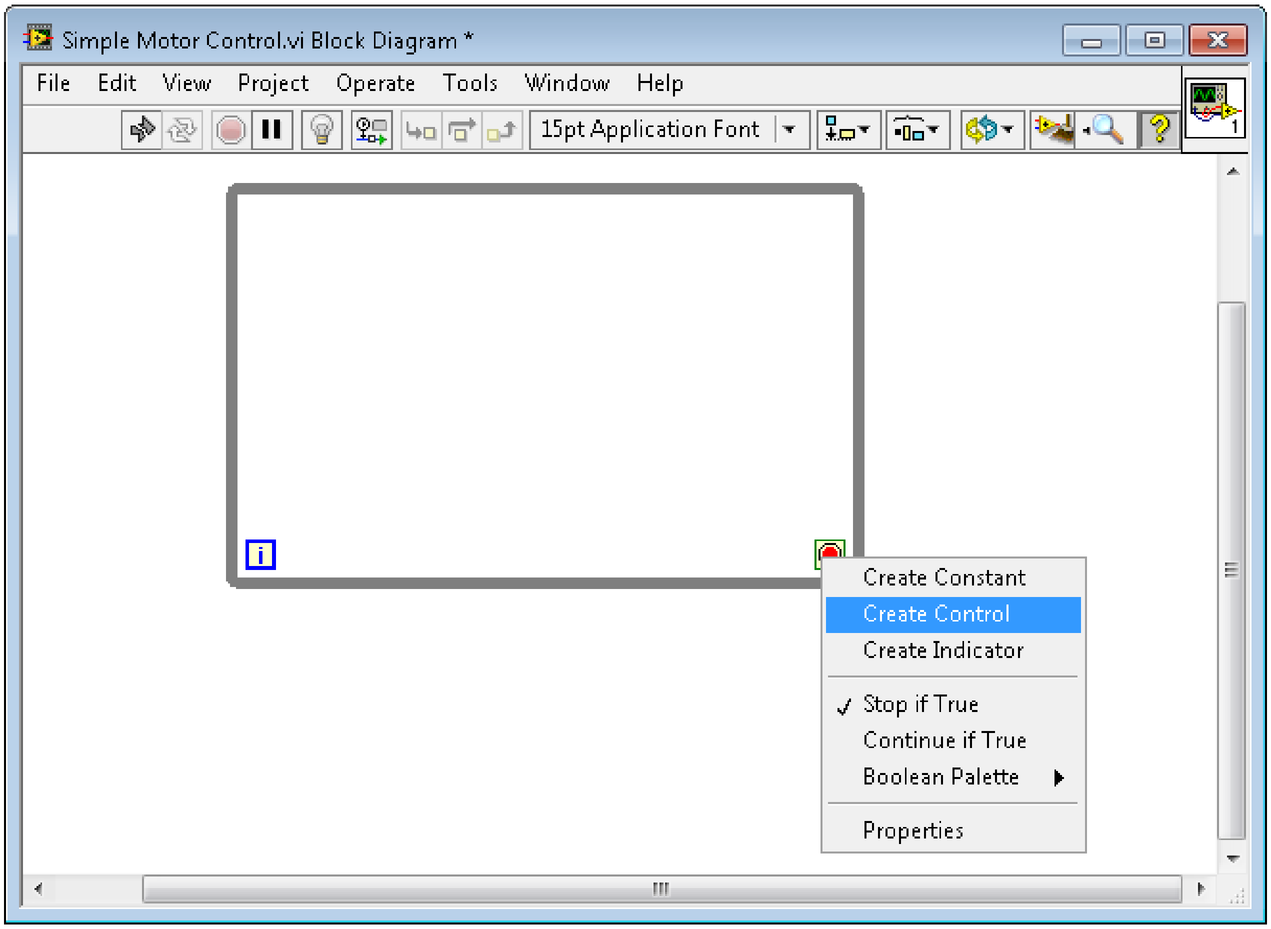
Save the new VI as *Simple Motor Control*. Notice that LabVIEW places the new VI inside the cRIO target tree, which means the VI will run on the cRIO.



Programming Motor Control in LabVIEW

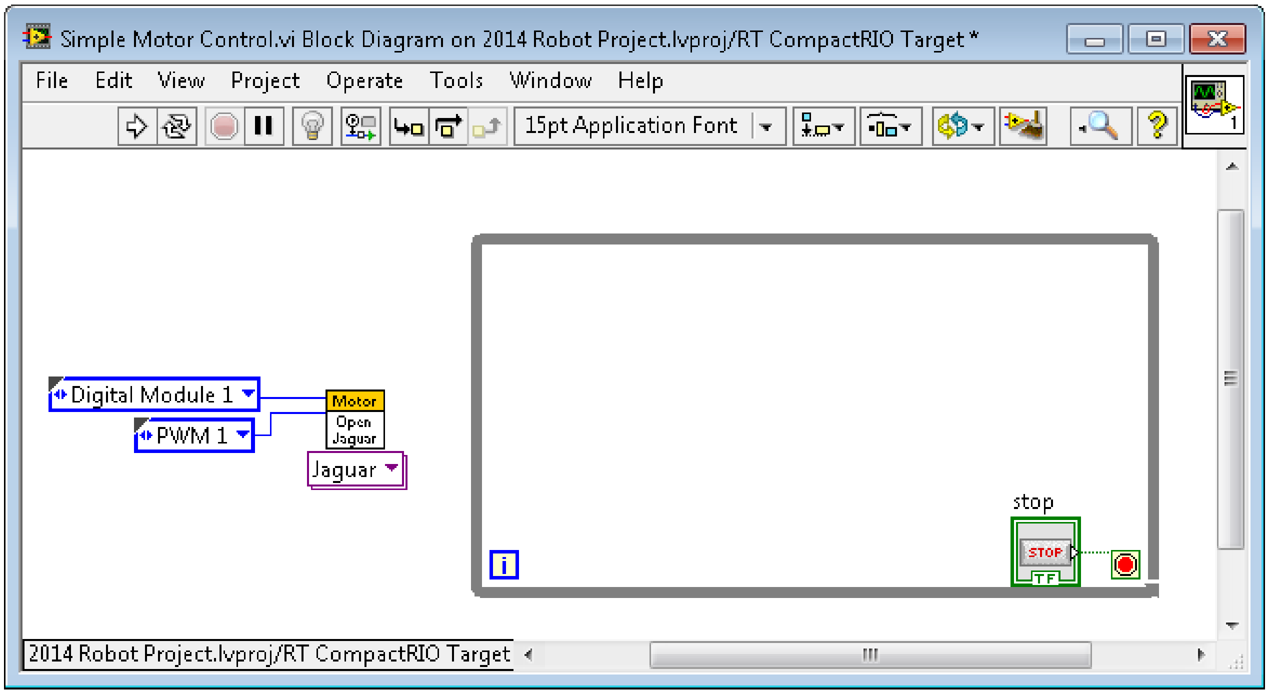
Double-click the **Simple Motor Control** VI to open it. Next, go to the block diagram by selecting **Window»Show Block Diagram**. Now we will begin the actual coding required to get the motor moving. First create a while loop by right-clicking anywhere in the white space on the block diagram and selecting **Programming»Structures»While Loop**. Click, drag, and release on the block diagram to specify the size of the while loop.

The while loop allows all code inside of it to run continuously until a condition or set of conditions are met. Right-click the **conditional terminal** in the bottom right corner of the while loop and select **create control**. Notice that a Stop button appears on the front panel.



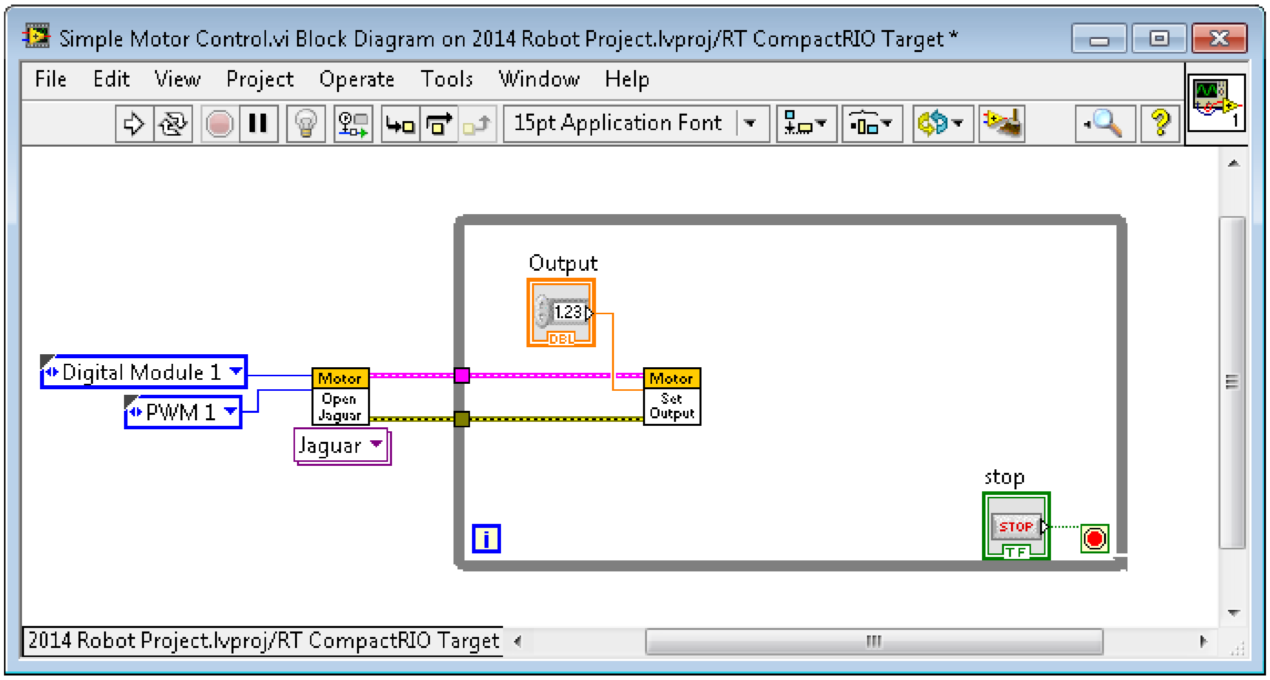
Now right-click anywhere in the white space of the block diagram and navigate to the **WPI Robotics Library»RobotDrive»Advanced»Motor Control** palette and place the **Open Jaguar** VI on the block diagram to the left of the while loop. Select **Jaguar** as the motor type. Right-click on the DIO Module input of the **Open** VI and create a constant. Set the value to Digital Module 1. This selects the digital module (9403) in the second slot of the cRIO.

Next, right-click on the **PWM Channel** input of the Open VI and create a constant. Set the value to PWM 1. This selects the PWM channel that our motor controller is connected to.

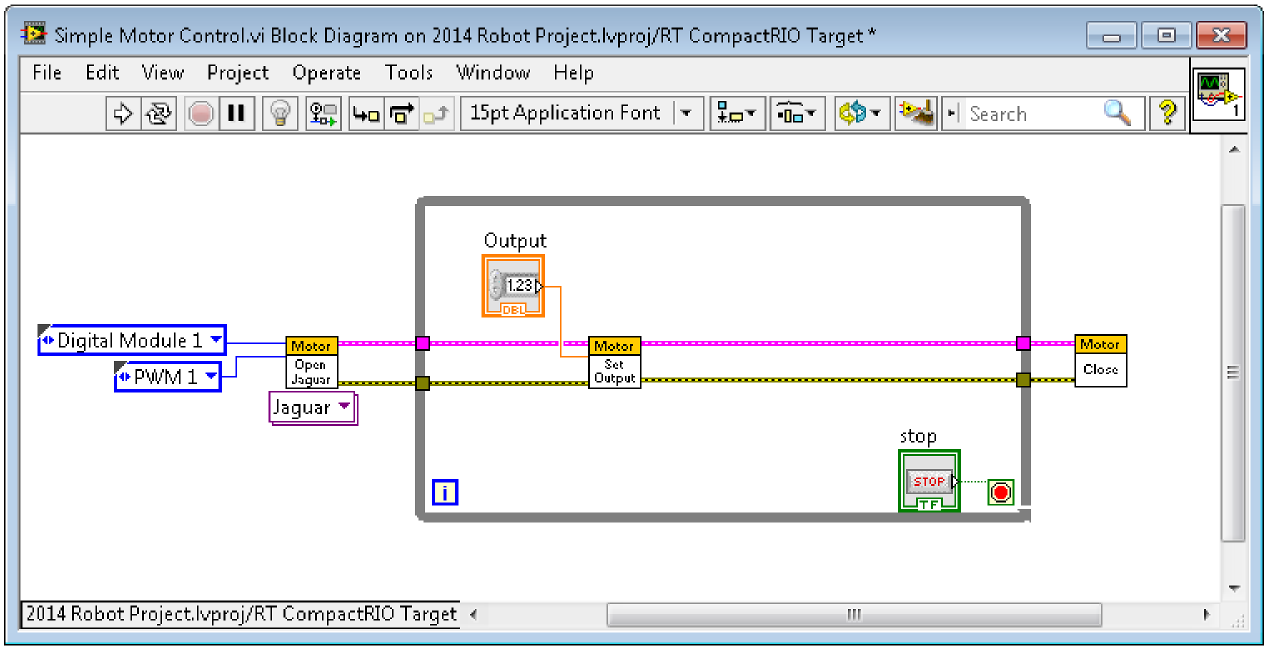

Simple Motor Control.vi Block Diagram on 2014 Robot Project.lvproj/RT CompactRIO Target *
File Edit View Project Operate Tools Window Help
15pt Application Font
Digital Module 1
PWM 1
Motor
Open Jaguar
Jaguar
stop
STOP
TF
2014 Robot Project.lvproj/RT CompactRIO Target

Now place the **Set Output** VI from **WPI Robotics Library»RobotDrive»Advanced»Motor Control** in the middle of the while loop. Wire the **MotorControlDevRef** and **error** terminals of the Open Jaguar VI to the inputs of the Set Output VI. Right-click the **output** input of the Set Output VI and select **Create»Control**.

This creates a control on the front panel that allows us to specify the speed of the motor. The value of this control can be between 1 and -1. Because we have the Set Output VI in a while loop, we can change the motor speed from the front panel while the motor is running. This VI will continue to run until your push the stop button.

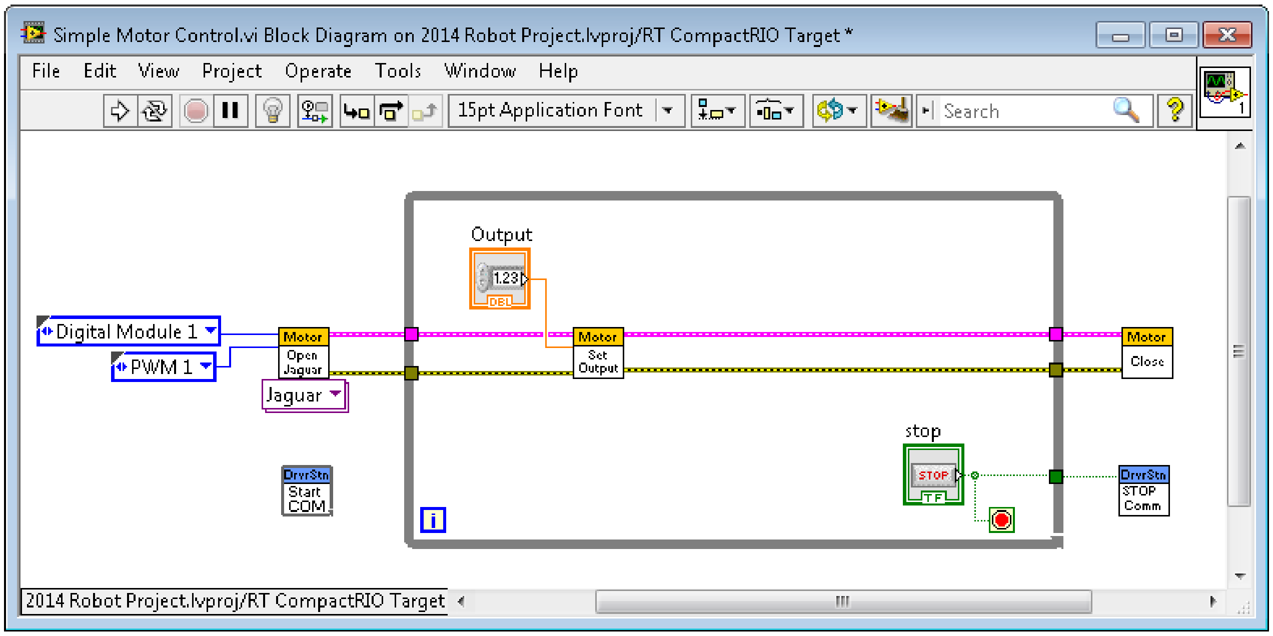

Simple Motor Control.vi Block Diagram on 2014 Robot Project.lvproj/RT CompactRIO Target *
File Edit View Project Operate Tools Window Help
15pt Application Font
Output
1.23
DBL
Digital Module 1
PWM 1
Motor
Open Jaguar
Jaguar
Motor
Set Output
stop
STOP
TF
2014 Robot Project.lvproj/RT CompactRIO Target

Finally, place the **Close** VI from **WPI Robotics Library»RobotDrive»Advanced»Motor Control** palette to the right of the while loop and then wire the **reference** and **error** outputs of the Set Output VI to the inputs of the Close VI.



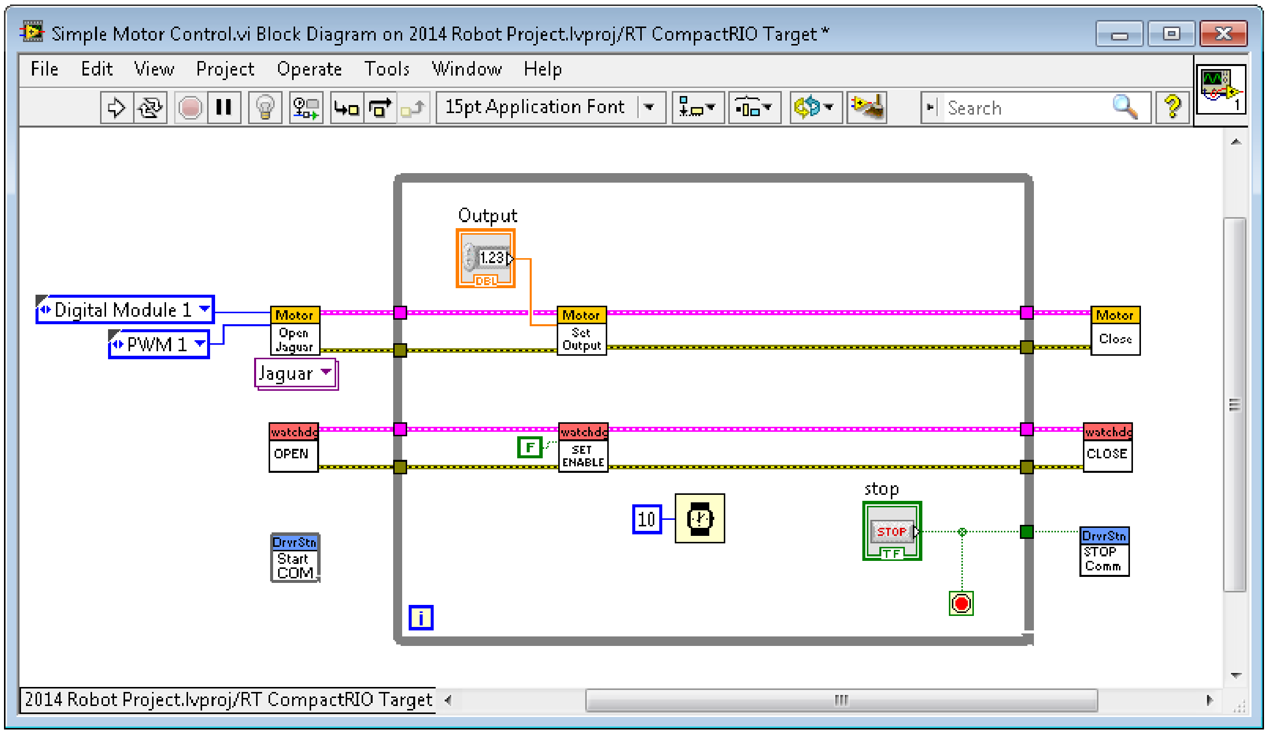
Next place the **Start Communication.vi** from **WPI Robotics Library»Driver Station** outside of the while loop. The **Stop Communication.vi** from **Robotics Library»Driver Station** also needs to be placed to the right of the while loop. Connect the stop button we created earlier to the input of the **Stop Communication** VI.

These VIs are necessary to establish a connection with the cRIO when the program starts and to close that connection when the program stops.

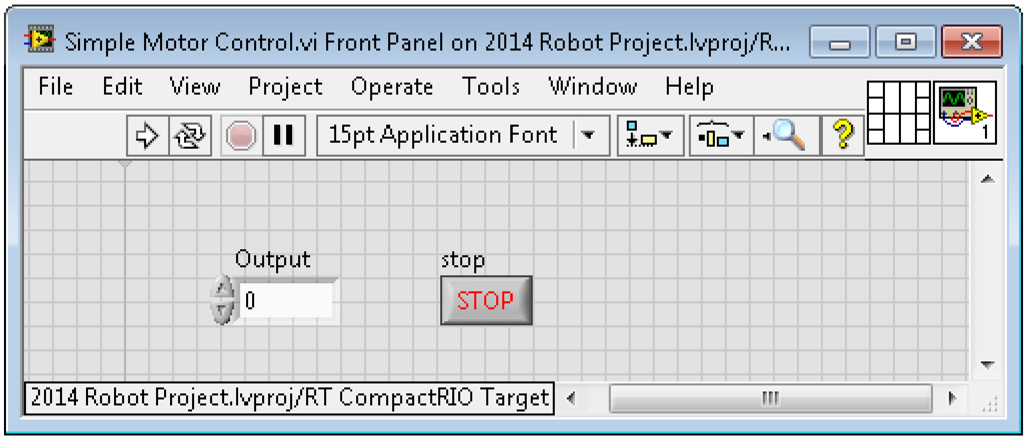


Finally create some additional room in your while loop, and then place the Watchdog **Open**, **Set Enabled**, and **Close** VIs located at **WPI Robotics Library»Utilities»Watchdog** on the block diagram as shown in the figure below. Right-click on the **enable watchdog** input and create a false constant. This disables the optional user watchdog.

A watchdog allows you to stop all robot processes if an error occurs in your program. This can be a very useful safety feature. You will also need to add a time delay of 10 ms to the while loop. This wait statement causes the VI to allow other processes to execute. Without this statement, your program will use 100% of the processor resources.



Now go to the front panel, save the VI, and click the run button. The VI will now download to the cRIO.


Simple Motor Control.vi Front Panel on 2014 Robot Project.lvproj/R...
File Edit View Project Operate Tools Window Help
15pt Application Font
Output
0
stop
STOP
2014 Robot Project.lvproj/RT CompactRIO Target

Once the program has finished downloading, try changing the value of the speed control from 0 to 0.1. The motor should now start moving slowly.

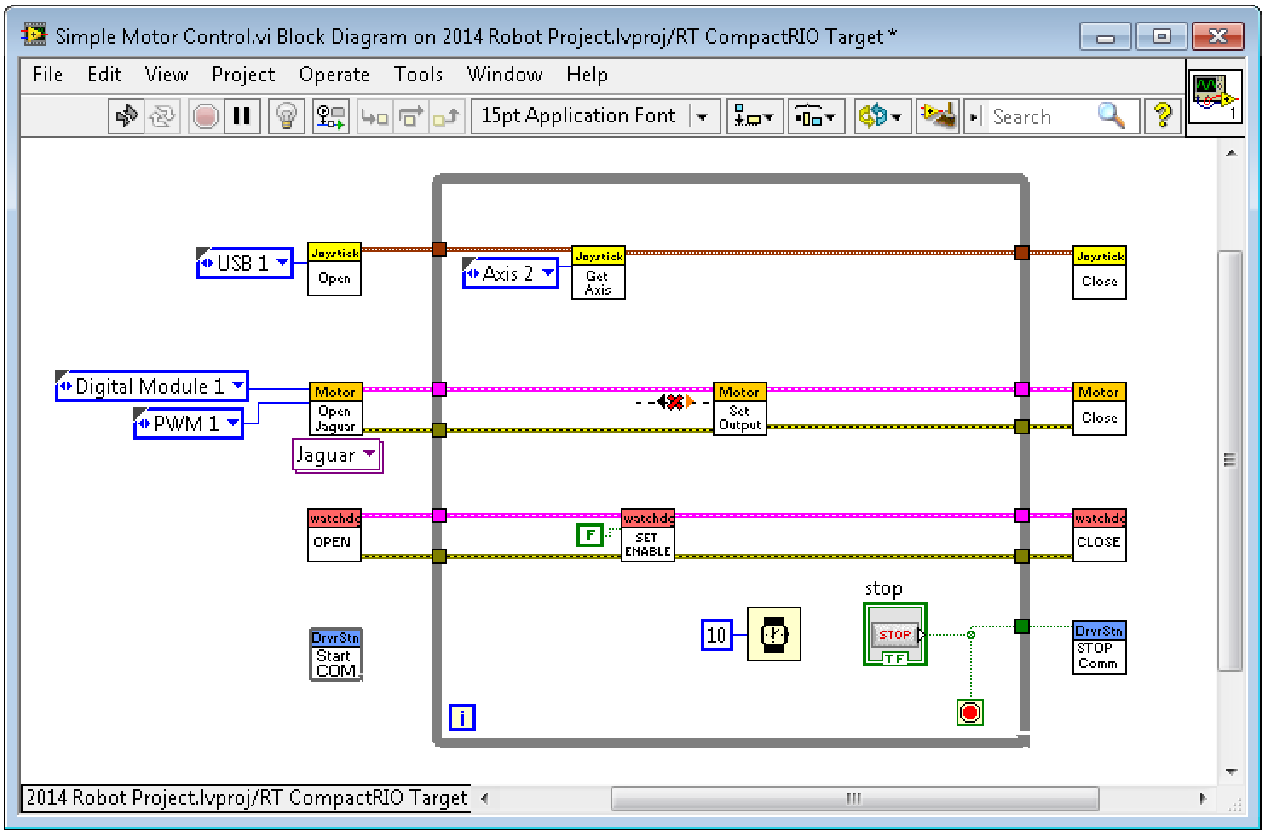
Keep increasing this number up to 1, at which point the motor will spin at full speed. Notice that inputting negative values for speed will spin the motor in the opposite direction. Press the stop button on the front panel to stop the program.

Adding a Joystick

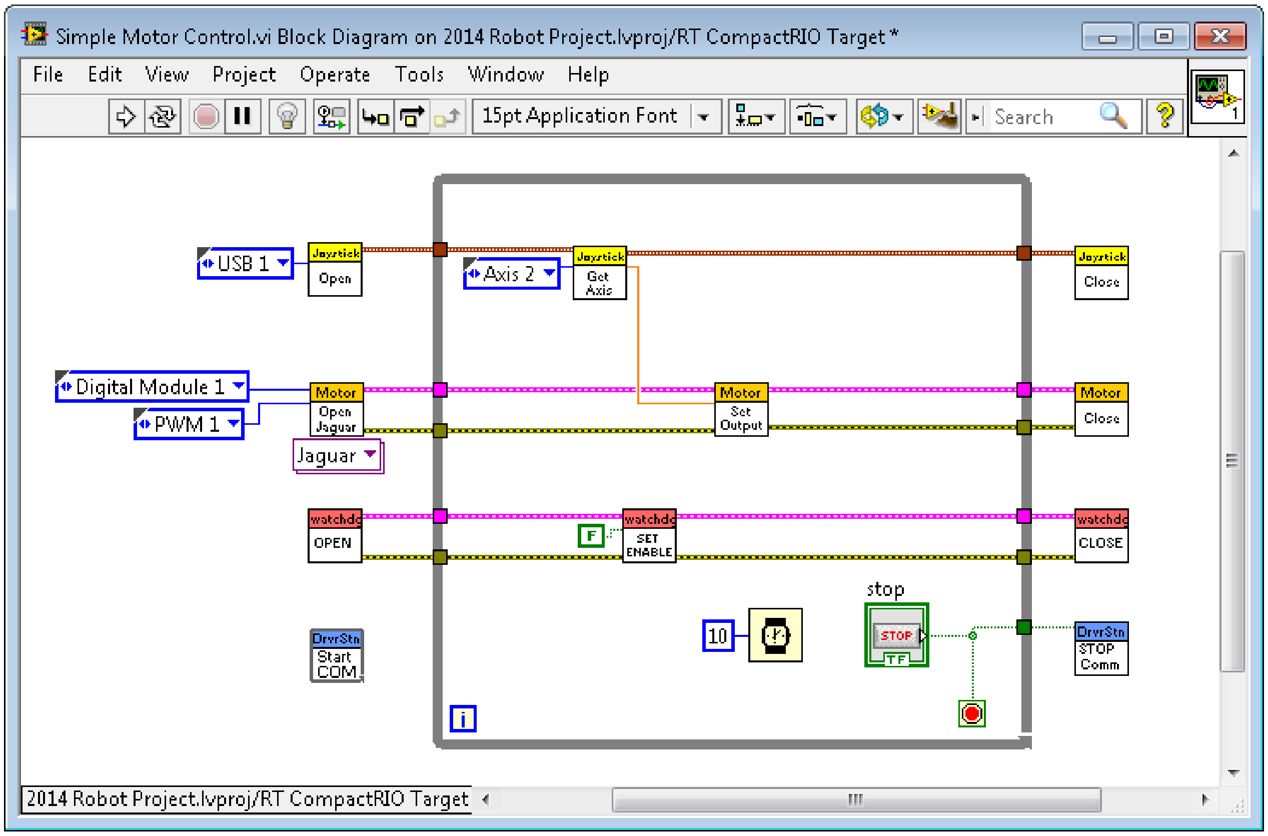
In order to control the motor with a joystick, first plug the joystick into a USB port 1 on the driver station and set it to Joystick 1 in the Driver Station. Then go back to the block diagram and place the joystick **Open** VI located at **WPI Robotics Library»Driver Station»Joystick** to the left of the while loop. Right-click on the joystick device input and create a constant. Select a value of USB 1.

Now place the **Get Axis** VI from **WPI Robotics Library»Driver Station»Joystick** inside the while loop and connect the **JoystickDevRef** output of the joystick Open VI to the reference input of the Get Axis VI. Right-click on the Axis input of the Get Axis VI and create a constant. Click on the constant and select Axis 2. On our joystick, Axis 2 is the forward/backward axis.

If you are using a different joystick, you may find that your joystick has different mappings for the axis. Delete the speed control made earlier by clicking the control and hitting the delete key on the keyboard.



Remove the broken wires by going to **Edit»Remove Broken Wires**. Next, connect the **value** output from the joystick Get Axis VI to the **output** input of the PWM Set Speed VI. Finally, close the joystick reference with a Close VI.



Click the run button to download the VI to the cRIO. Once the code downloads, move the joystick forwards and backwards to move the motor with varying speed and direction. Press the stop button when finished.

Conclusion

Congratulations, now you're up and running with a simple motor example. Explore the rest of the Training Material and Resources page to learn about acquiring data from sensors, taking and processing images from the camera, and integrating the data you read from both the sensors and the camera into a project to move motors. Have fun programming and use the skills and techniques you learn in these tutorials to build a better robot!