

# FRC WPI Robotics Library Overview

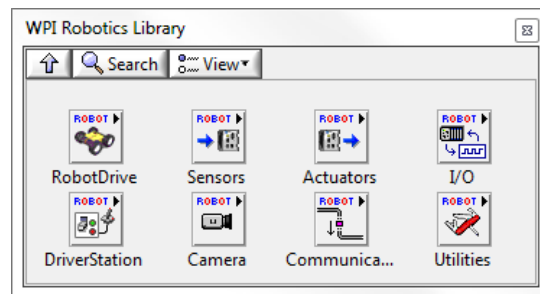
---

## Contents

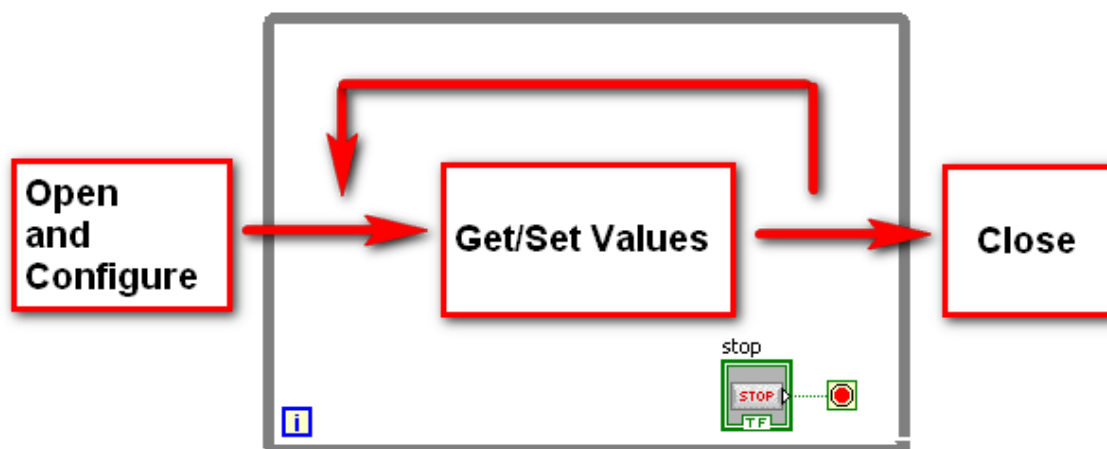
- 1.1 Introduction
- 1.2 RobotDrive
- 1.3 Sensors
- 1.4 Actuators
- 1.5 I/O
- 1.6 Driver Station
- 1.7 Compressor
- 1.8 Camera
- 1.9 Utilities
- 1.10 Conclusion

## Introduction

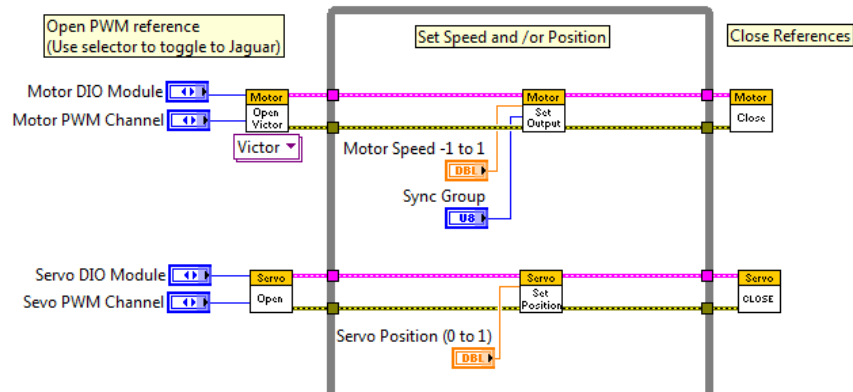
In this tutorial we will explore and give an overview of the functionality in the WPI Robotics Library in LabVIEW. The LabVIEW FRC VIs are located under **WPI Robotics Library** in the Functions Palette.



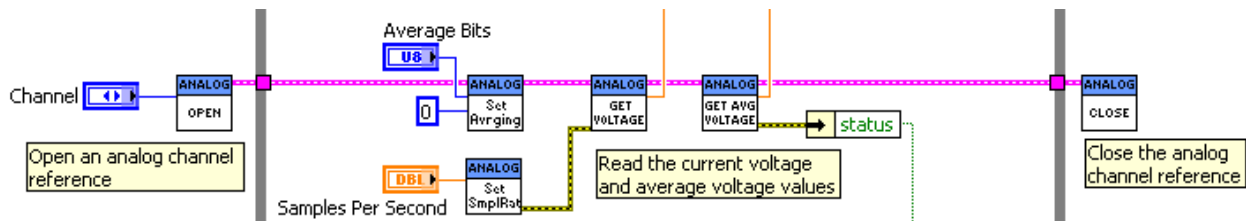
As you read through this tutorial, notice how the entire library follows the Open»Get/Set»Close paradigm.



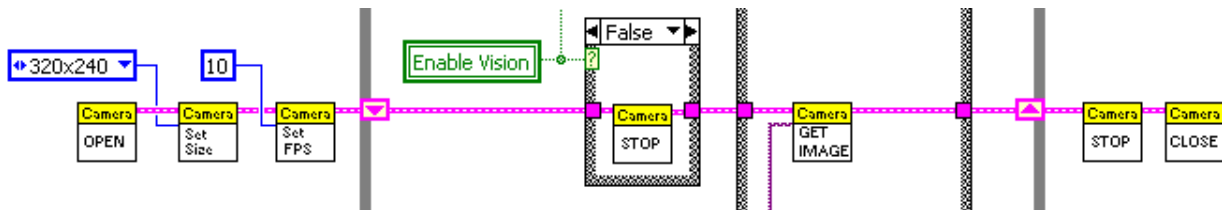
Each palette will contain open, get and/or set, and close VIs, whether you are using VIs from the RobotDrive palette,



The I/O palette,



Or the Camera palette,

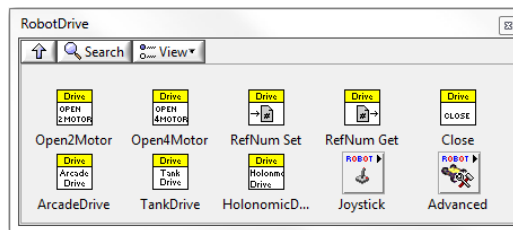


They will all follow the Open»Get/Set»Close paradigm.

Understanding and applying the Open»Get/Set»Close paradigm will be key in successfully programming your robot. See the examples in the NI Example Finder accessible from the lower right corner of the LabVIEW getting started window for additional examples of the Open»Get/Set»Close paradigm. You can also access the Example Finder by going to **Help»Find Examples**.

## RobotDrive

The RobotDrive palette provides VIs to control the drive motors on your robot. You have many options for designing and configuring the robot drive train. The RobotDrive VIs support two-wheel and four-wheel drive robots. VIs are provided for arcade, tank, holonomic and skid steering systems. See the [Basic Motor Control](#) tutorial for additional information.



### Arcade Drive

Arcade Drive accepts inputs from one of the joysticks connected to the Driver Station. This VI then mixes the x-axis and y-axis values to supply different signals to each motor for turning.

### Tank Drive

Tank Drive enables you to operate a robot with a tank drive motor configuration. Tank driving uses two joysticks. You control the left and right motors independently with the two joysticks. The two y-axis values from the joysticks supply signals to each motor for robot movement.

### Holonomic Drive

Holonomic Drive controls robots with special wheels that allow the robot to drive in any direction without first rotating or maneuvering the robot. Holonomic drive robots use omnidirectional wheels that have a center, large main wheel mounted on a drive shaft and several smaller rollers around the perimeter of the main wheel. The smaller rollers allow the robot to travel in a direction that is perpendicular to the direction of travel with the main wheel without having to first move the main wheel. The robot can travel in any direction by moving the main wheel and the rollers at different speeds.

### Joystick

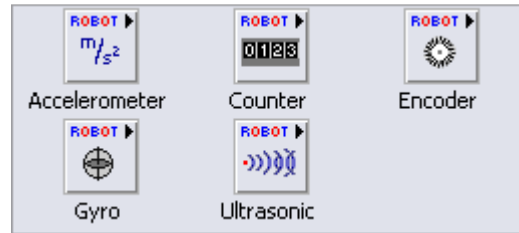
The Joystick palette is also included in the Driver Station palette. Use the Joystick VIs to return information from up to 4 joysticks connected to the driver station.

### Advanced

The RobotDrive Advanced palette provides the ability to set the speed and direction for motors directly. It provides a lower level of control than the VIs on the RobotDrive main palette.

## Sensors

Use the Sensor VIs to send and receive information from the sensors you connect to the CompactRIO modules. See the [Sensor Basics](#) tutorial for additional information.



### Accelerometer VIs

The Accelerometer VIs are used to determine the acceleration of a robot or robot component as measured by an accelerometer. Accelerometers measure both dynamic motion, or vibration, and static motion, or gravity, by measuring x-axis and y-axis acceleration. The accelerometer in last year's kit of parts is a two-axis accelerometer. It provides acceleration data in the x and y-axis relative to the circuit board. An accelerometer can be used, for example, to measure the angle of a boom arm on your robot.

### Counter VIs

The Counter VIs are used to count the number of pulses in a digital signal. You can use the Counter VIs to count up or count down based on the rising or falling edges of the signal. The Counter VIs are especially useful when an algorithm or subroutine necessary to your program is not already incorporated in some of the higher level encoder VIs.

### Encoder VIs

The Encoder VIs are used to get position information measured by a quadrature encoder. The FPGA on the FRC cRIO uses 4x decoding. This setup will return a count value for every rising and falling edge as well as Boolean value that indicates direction. From this information and the specifications for your encoder, you can determine the angular position of a gear or motor.

### Gyro VIs

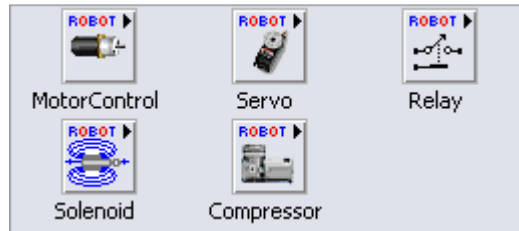
The Gyro VIs are used to set and monitor the heading, or direction, of the robot as measured by a gyroscope transducer. The Gyro VIs measure the instantaneous rate of rotation or yaw rate of the robot. You can use the Gyro VIs to make sure that your robot is moving on a desired heading by measuring the difference between the gyro reading and a desired set point, and then correcting for any error.

### Ultrasonic VIs

The Ultrasonic VIs are used to measure the distance between an ultrasonic sensor on the robot and the closest object in range of the sensor. This can be useful in developing a system that keeps your robot from running into objects.

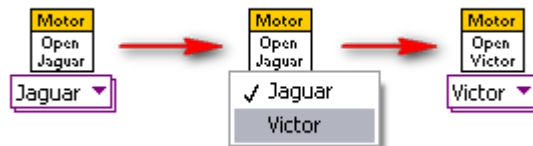
## Actuators

The VIs in the Actuators palette are used to control the behavior of actuators such as relays, motors, and solenoids.



### Motor Control VIs

The Motor Control VIs are used to set or get the speed of a motor using a Jaguar or Victor motor controller. The Motor Control VIs use preconfigured PWM range and deadband values specific to either the Jaguar or Victor motor controller, depending on which instance of the VIs you select, so you do not need to perform any conversions or scaling calculations on the raw PWM values.



### Relay VIs

The Relay VIs are used to manipulate relays. You can open and close relays to control the power source for a motor or other device.

### Servo VIs

The Servo VIs are used to set or get the angle and position of a servo motor. If you use a servo, use the Servo VIs instead of the PWM VIs. The Servo VIs use preconfigured PWM range and deadband values specific to the servo so you do not need to perform any conversions or scaling calculations on the raw PWM values.

### Solenoid VIs

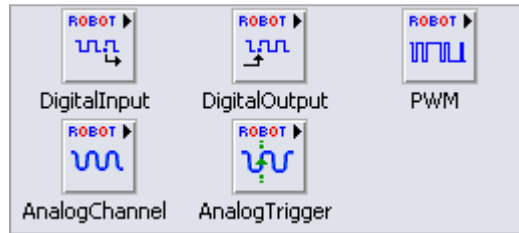
The Solenoid VIs are used to operate a solenoid. A solenoid is a pneumatic valve that can release air to move a part on the robot.

### Compressor

Use the Compressor VIs to start and stop a compressor. The Compressor VIs manipulate a pressure switch connected to the high pressure side of the pneumatic circuit of the robot, controlling the compressor of the robot.

## I/O

The I/O VIs can be used to send and receive analog and digital data from a module on the cRIO. These VIs are meant for low-level applications—connecting to sensors, actuators, etc.



### Analog Trigger VIs

The Analog Trigger VIs are used to create digital signals from analog signals. You can use analog triggers as sources for the Counter VIs and Encoder VIs. The analog trigger circuit on the target produces three types of output: TriggerState, InWindow, and Pulse. Use the GetOutput VI to get the TriggerState and InWindow output from the trigger.

### Analog Channel VIs

Use the Analog Channel VIs to acquire an analog signal from a channel on a module on the cRIO. You can configure how the channel samples the signal by specifying the number of bits and the number of oversample bits. You can acquire the raw signal or the scaled voltage signal.

### Digital Input VIs

The Digital Input VIs are used to open or close a reference to a digital input and to acquire digital values that sensors send to the cRIO.

### Digital Output VIs

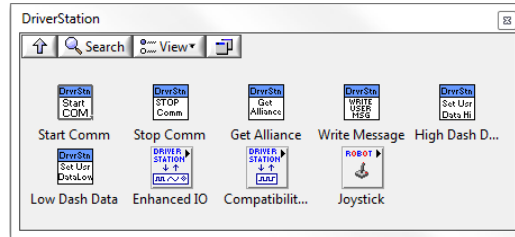
The Digital Output VIs are used to open or close a reference to a digital output and to set the value of the digital output.

### PWM VIs

The PWM VIs are used to generate a hobby-style pulse width modulation (PWM) signal. You can use PWM signals to drive motor controllers and servos. Use the VIs on this palette with custom motor controllers or servos. Otherwise, use the VIs on the Victor, Jaguar, or Servo palettes to use PWM for those devices.

## Driver Station

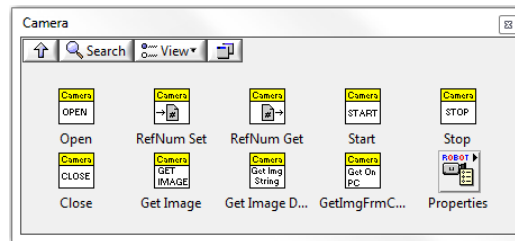
Use the DriverStation VIs to manipulate data flow between the robot, driver station, and laptop. The Driver Station VIs can receive, interpret, and display both analog and digital data from your various sensors and joysticks.



The Driver Station VIs can be used to relay information from the robot to the laptop, check the battery power of the robot, set the controls of the robot based on the current competition mode, and display information from the outputs you define. For more information, see the [Dashboard Coding](#) tutorial

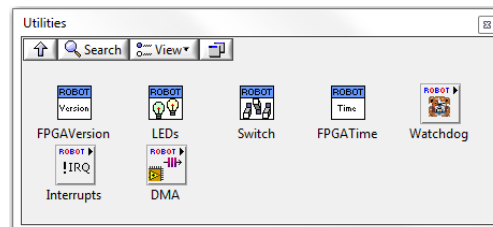
## Camera

You can use the Camera VIs to configure the Axis camera properties, acquire images directly on the cRIO, or to retrieve images from the cRIO in your Dashboard project. The camera VIs deal exclusively with image acquisition and setup. Explore the FIRST Vision palette and the [Image Processing](#) tutorial to learn more about the available image processing functionality.



## Utilities

Use the Utility VIs to return information about the cRIO such as the version of the FPGA, the status of LEDs and switches.



In previous years, Watchdog VIs were used to control the user watchdog if you chose to enable it. This was used to disable the PWM relay and solenoid outputs if the Feed Watchdog VI did not receive an input within a specified amount of time. If the watchdog timed out all of the motors and actuators were immediately disabled on the robot. This year, a new safety method is being used. See [this tutorial](#) for additional information.

## **Conclusion**

This concludes the FRC Robotics Library overview. You now know what tools are available to you and have a high level view of how they can be used to build a better robot! Please refer to the [NI LabVIEW FRC Help document](#) and the [Getting Started Programming with FRC Hardware: Videos and Tutorials](#) page for more detailed information regarding the contents of the WPI Robotics Library.