

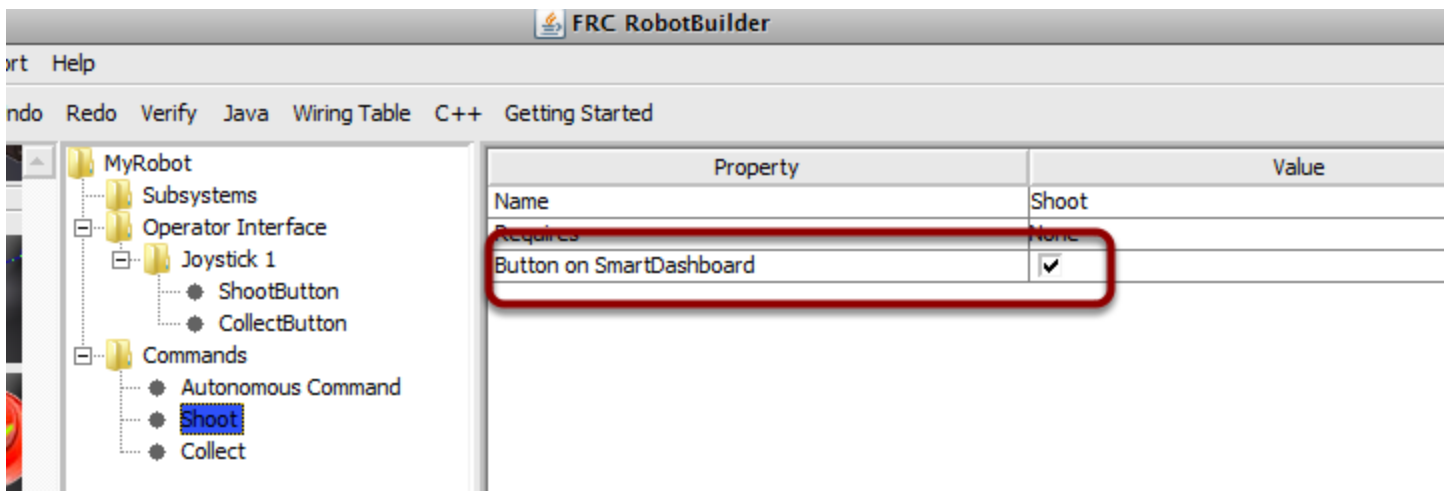
Adding a button to SmartDashboard to test a command

Adding a button to SmartDashboard to test a command

Commands are easily tested by adding a button to the SmartDashboard to trigger the command. In this way, no integration with the rest of the robot program is necessary and commands can easily be independently tested. This is the easiest way to verify commands since with a single line of code in your program, a button can be created on the SmartDashboard that will run the command. These buttons can then be left in place to verify subsystems and command operations in the future.

This has the added benefit of accommodating multiple programmers, each writing commands. As the code is checked into the main robot project, the commands can be individually tested.

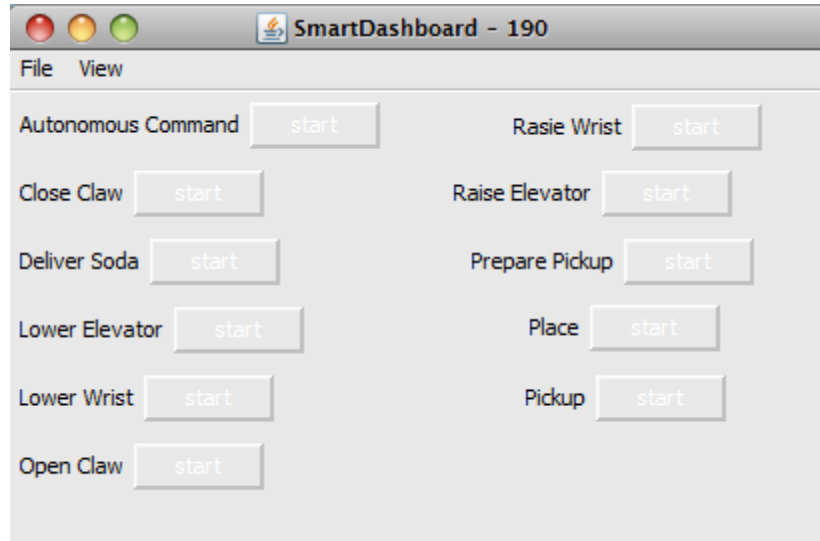
Creating the button on the SmartDashboard



The button is created on the SmartDashboard by putting an instance of the command from the robot program to the dashboard. This is such a common operation that it has been added to RobotBuilder as a checkbox. When writing your commands, be sure that the box is checked, and buttons will be automatically generated for you.

Adding a button to SmartDashboard to test a command

How to operate the buttons on the SmartDashboard



The buttons will be generated automatically and will appear on the dashboard screen. You can put the SmartDashboard into edit mode, and the buttons can then be rearranged along with other values that are being generated. In this example there are a number of commands, each with an associated button for testing. The button is labeled "Start" and pressing it will run the command. As soon as it is pressed, the label changes to "Cancel" and pressing it will interrupt the command causing the Interrupted() method to be called.

Adding commands manually

```
SmartDashboard::PutData("Autonomous Command", new AutonomousCommand());
SmartDashboard::PutData("Open Claw", new OpenClaw());
SmartDashboard::PutData("Close Claw", new CloseClaw());
SmartDashboard::PutData("Lower Wrist", new LowerWrist());
SmartDashboard::PutData("Rasie Wrist", new RasieWrist());
SmartDashboard::PutData("Lower Elevator", new LowerElevator());
SmartDashboard::PutData("Raise Elevator", new RaiseElevator());
SmartDashboard::PutData("Prepare Pickup", new PreparePickup());
SmartDashboard::PutData("Pickup", new Pickup());
SmartDashboard::PutData("Place", new Place());
SmartDashboard::PutData("Deliver Soda", new DeliverSoda());
```

Commands can be added to the SmartDashboard manually by writing the code yourself. This is done by passing instances of the command to the PutData method along with the name that should be associated with the button on the SmartDashboard. These instances are scheduled whenever the button is pressed. The result is exactly the same as RobotBuilder generated code, although clicking the checkbox in RobotBuilder is much easier than writing all the code by hand.