

Working with Commands and Subsystems

When using the Command-based framework Shuffleboard makes it easier to understand what the robot is doing by displaying the state of various commands and subsystems in real-time.

Displaying subsystems

To see the status of a subsystem while the robot is operating in either autonomous or telop modes, that is what it's default command is and what command is currently using that subsystem send a subsystem instance to Shuffleboard:

```
Java:
SmartDashboard.putData(subsystem-reference);

C++:
SmartDashboard::PutData(subsystem-pointer);
```

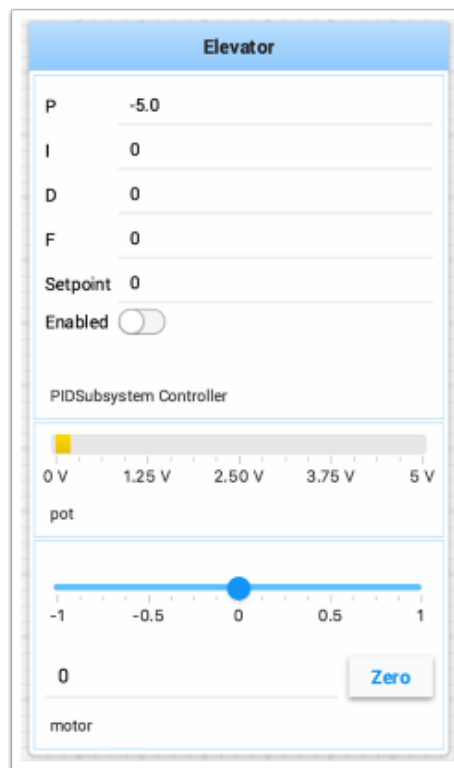
Shuffleboard will display the subsystem name, the default command associated with this subsystem, and the currently running command. In this example the default command for the Elevator subsystem is called "AutonomousCommand" and it is also the current command that is using the Elevator subsystem.



Working with Commands and Subsystems

Testing and viewing subsystems in Test mode

In Test mode (Test/Enabled in the driver station) subsystems maybe displayed in the LiveWindow tab with the sensors and actuators of the subsystem. This is ideal for verifying of sensors are working and seeing the values that they are returning. In addition the actuators can be operated, for example motors can be operated using sliders to set the speed and direction. For PIDSubsystems the P, I, D, and F constants are displayed along with the setpoint and an enable control. This is useful for tuning PIDSubsystems by adjusting the constants, putting in a setpoint, and enabling the embedded PIDController. Then the mechanisms response can be observed. Then the parameters can be changed, the PIDController re-enabled until a reasonable set of parameters is found.



More information on tuning PIDSubsystems can be found here: [Testing and tuning PID loops](#). Using RobotBuilder will automatically generate the code to get the subsystem displayed in Test mode. The code that is necessary to have subsystems displayed is shown below where *subsystem-name* is a string containing the name of the subsystem:

Working with Commands and Subsystems

```
setName(subsystem-name);
```

Displaying commands

Using commands and subsystems makes very modular robot programs that can easily be tested and modified. Part of this is because commands can be written completely independently of other commands and can therefore be easily run from Shuffleboard. To write a command to Shuffleboard use the `SmartDashboard.putData` method as shown here:

```
SmartDashboard.putData("ElevatorMove: up", new ElevatorMove(2.7);
```

Shuffleboard will display the command name and a button to execute the command. In this way individual commands and command groups can easily be tested without needing special test code in a robot program. In the image below there are a number of commands contained in a Shuffleboard list. Pressing the button once runs the command and pressing it again stops the command. To use this feature the robot must be enabled in teleop mode.

