

Creating a subsystem

Creating a subsystem

Subsystems are classes that encapsulate (or contain) all the data and code that make a subsystem on your robot operate. The first step in creating a robot program with the RobotBuilder is to identify and create all the subsystems on the robot. Examples of subsystems are grippers, ball collectors, the drive base, elevators, arms, etc. Each subsystem contains all the sensors and actuators that are used to make it work. For example, an elevator might have a Jaguar speed controller and a potentiometer to provide feedback of the robot position.

Creating a subsystem by dragging from the palette

Creating a subsystem by dragging from the palette

Drag the subsystem icon from the palette to the Subsystems folder in the robot description to create a subsystem class.

Creating a subsystem by using the context menu on the Subsystem folder

Creating a subsystem by using the context menu on the Subsystem folder

Right-click on the Subsystem folder in the robot description to add a subsystem to that folder.

Name the subsystem

Name the subsystem

After creating the subsystem by either dragging or using the context menu as described above, simply type the name you would like to give the subsystem. The name can be multiple words

Creating a subsystem

separated by spaces, RobotBuilder will concatenate the words to make a proper Java or C++ class name for you.

Drag actuators and sensors into the subsystem

Drag actuators and sensors into the subsystem

There are two steps to adding components to a subsystem:

1. Drag actuators or sensors from the palette into the subsystem as required.
2. Give the actuator or sensor a meaningful name
3. Edit the properties such as module numbers and channel numbers for each item in the subsystem.

RobotBuilder will automatically use incrementing channel numbers for each module on the robot. If you haven't yet wired the robot you can just let RobotBuilder assign unique channel numbers for each sensor or actuator and wire the robot according to the generating wiring table.

This just creates the subsystem in RobotBuilder, and will subsequently generate skeleton code for the subsystem. To make it actually operate your robot please refer to: [Writing the code for a subsystem in Java](#) or [Writing the code for a subsystem in C++](#).