Output Control

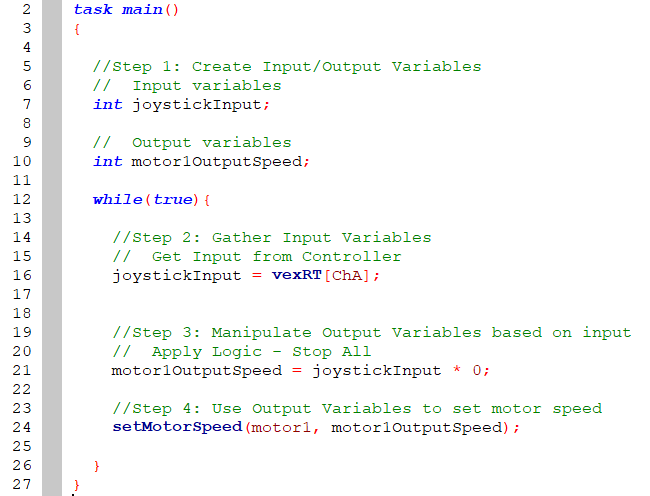
# **Description**

These tasks will act as an introduction to output control

# **Specific Requirements**

## **Basic Structure**

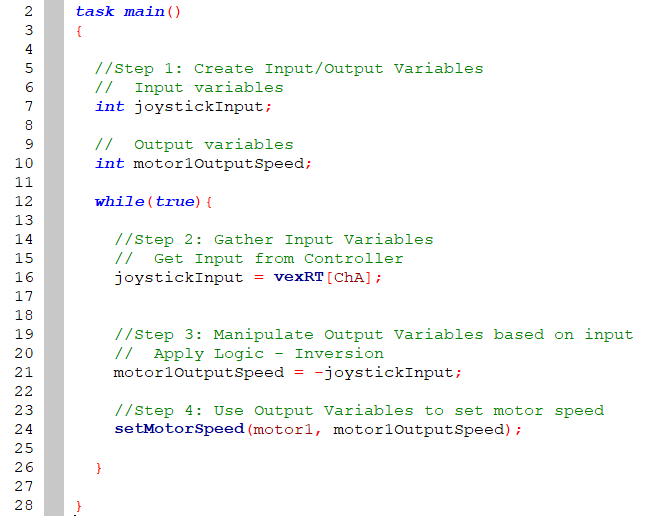
1. The basic structure of output control is as follows:
   1. **Step 1: Input and output variables are created**
   2. **Step 2: Inputs from the controller are stored in Input variables**
   3. **Step 3: Output variables are manipulated based on the Input**
   4. **Step 4: Output variables are used to set the motor speeds**
2. Code the following example for driving a motor with a joystick including comments!



1. The logic section states that joystickInput is always set to zero. The motor never moves! Let’s try some other logical operations on joystick input…

## **Inversion**

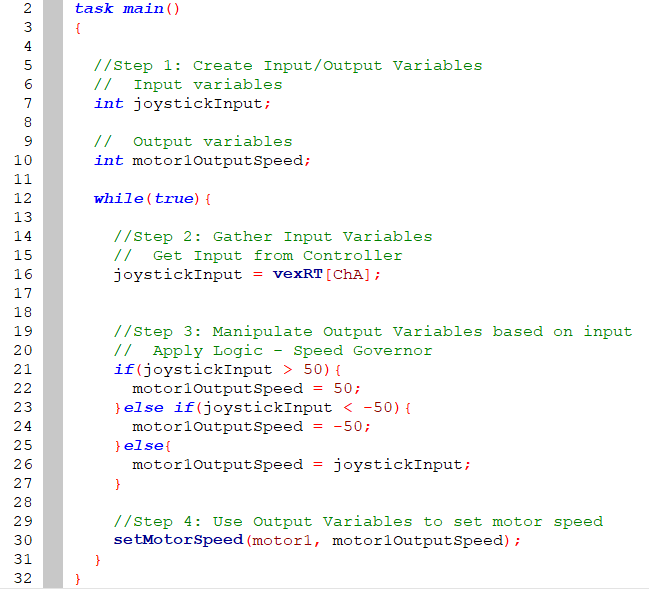
1. Inversion is useful when motors face opposite directions or when the joystick direction doesn’t match the motor direction.
2. To demonstrate, edit the logic section to invert the joystick control as follows:



1. Notice the joystick control is now reversed!

## **Speed Governors**

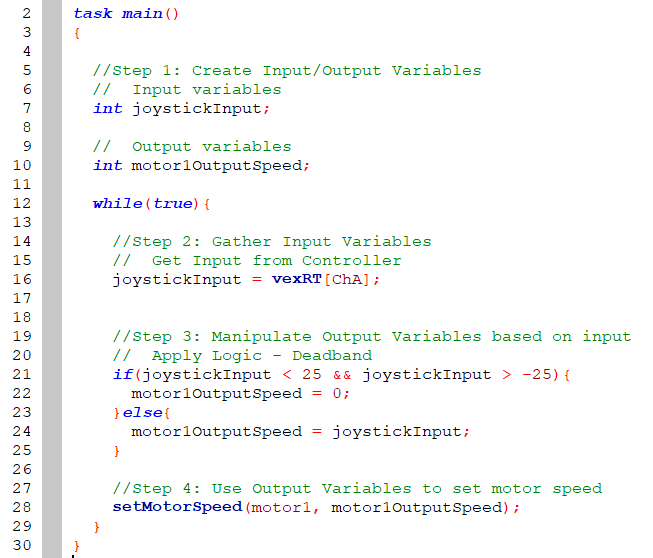
1. Speed governors are a safety measure used when vehicles should not travel faster than a certain speed – even if the vehicle is physically capable of doing so
2. Edit the logic section to check if the input is greater than 50. If so, set the speed to 50.  
   Additionally, check if the input is less than -50. If so, set the speed to -50:



1. Notice the motor speed is not capped (governed) at 50

## **Deadbanding**

1. Deadbands are used to prevent accidental or small outputs from causing motor output. This is sometimes useful if a joystick is sensitive and return non-zero values at the center position
2. To demonstrate, edit the logic section to check if the input is less than 25 and greater than -25.  
   If so, set speed to zero:



1. Notice that the motor speed must be greater than 25 or less than -25!