

Engineering Lab

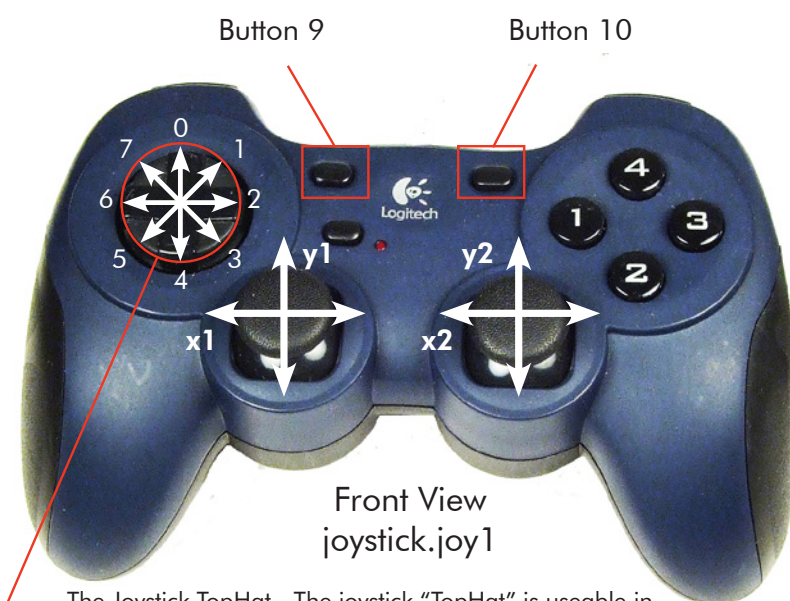
Remote Control Buttons

In this lesson you will:

1. Learn how to program the buttons on your remote controller.
2. Review the "JoystickDriver" include file.
3. Review the function "getJoystickSettings(joystick)".
4. Identify the names and locations of all buttons on the joystick.

Remote Control Button Overview

The joystick remote controller is a very powerful tool that a programmer can use to achieve finite control of their robot. Each button can be programmed to control a specific behavior, for example - goStraight, rightTurn, leftTurn, openGripper, closeGripper - allowing limitless options.



The Joystick TopHat - The joystick "TopHat" is useable in your program and accessed with the reserve word "joystick.joy1_TopHat". The TopHat returns values from -1 to 7. The -1 value signifies that nothing is pressed.



Joysticks: The joystick controller is named joystick.joy1. There are two joystick axes on the joystick.joy1 controller. They are the round knobs that are labeled x1/y1 and x2/y2 on the picture on the left. To access the "y-axis" of joy_1 the command would be "joystick.joy1_y1".

The joystick axis names are:

```
joystick.joy1_y1
joystick.joy1_x1
joystick.joy1_y2
joystick.joy1_x2
```

Note: ROBOTC has the capability of working with two joystick controllers at a time. There is also a joystick.joy2 remote controller that can be programmed with ROBOTC.

Buttons: There are 10 programmable buttons on the joystick controller. They are found in three locations.

Buttons 1-4 are located on the front right side of the joystick controller.

Buttons 5-8 are located on the top of the joystick controller.

Buttons 9 and 10 are above the joysticks.

Joystick buttons values are accessed: joy1Btn(1) through joy1Btn(10)

Note: There is a set of joy2Btn(#) that map to the second joystick controller.

Engineering Lab

Remote Control Lesson Background

This lesson assumes you understand how Bluetooth communications works, that you have programmed your robot to move around using joystick control, that you have optimized your code by creating a dead zone, and that you have slowed down your robot's speed through the use of variable math. If you have not completed those lessons, complete them before beginning this one.

Things to Remember with Every Remote Control Program

1. Every Remote Control Program will need to include the "JoystickDriver.c" include file.
2. The "getJoystickSettings(joystick)" function is built into the "JoystickDriver" include file. The getJoystickSettings function updates the joystick variables so that the program has the current conditions of the joystick.

Review

<pre> 1 #include "JoystickDriver.c" 2 3 task main() 4 { 5 while(1 == 1) 6 { 7 getJoystickSettings(joystick); 8 motor[motorC] = joystick.joy1_y1; 9 motor[motorB] = joystick.joy1_y2; 10 } 11 }</pre>	<p>Include all of the contents of the file joystickDriver in this program.</p> <p>The "getJoystickSettings(joystick)" function updates the variables that store the joystick controller's position.</p> <p>These are variables that map from specific parts of the joystick controller and will be used to set the power of motorB and motorC.</p>
--	--

Programming Remote Control Buttons

Joystick buttons behave the same way as the NXT touch sensors do. When the NXT touch sensor is pressed, it returns a value of "1". When it is not pressed, it returns a value of "0". Similarly, joystick buttons receive a value of "1" when pressed and a value of "0" when not pressed.

To introduce programming remote control buttons, we will be using the gripper arm connected to the NXT robot. If you have not built the gripper arm attachment, build it now. You can find directions for the attachment under the using buttons video.



Once we have the robot gripper arm attached, we will want to program two new behaviors. The new behaviors will be "Open the Gripper Arm" and "Close the Gripper Arm". To activate the buttons, we will use two buttons on our joystick controller. There are 10 possible buttons available on the controller:

Buttons 1-4 are on the front right side of the remote control

Buttons 5-8 are on the top of the remote control

And buttons 9 and 10 are located in the middle, on the top of the remote control.

For our gripper program, we are going to program buttons 5 and 6.

Engineering Lab

The Remote Control Program to Control Buttons 5 & 6

```

1  #include "JoystickDriver.c"
2
3  task main()
4  {
5      while(1 == 1)
6      {
7          getJoystickSettings(joystick);
8
9          if(joy1Btn(6) == 1)
10         {
11             motor(motorA) = 25;
12         }
13         else if(joy1Btn(5) == 1)
14         {
15             motor(motorA) = -25;
16         }
17         else
18         {
19             motor(motorA) = 0;
20         }
21     }
22 }

```

Joystick controller buttons are programmed using the reserved word:

`joy1Btn(n)`

"joy1" references remote control number one.

"Btn" is short for button

"n" can be replaced by any number from 1 to 10. This references which button you want to read the value from.

Line 9 refers to joystick controller button 6, and line 13 refers to joystick controller button number 5.

What do lines 19 through 22 do? Experiment by commenting that code out and see what happens.



Copy the program above into ROBOTC and download it to your robot. Make sure that the motor on the robot gripper is connected to motorA. When you press buttons 5 and 6, your robot gripper will open and close.



Note: If you download your program, hit the start button on ROBOTC, and find your remote control is not working, it could be because you don't have the Joystick Controller Debug screen open.

The joystick remote controller gives you unlimited control and programming possibilities. You can also call functions using buttons. See the next page for an example using functions for the joystick TopHat.

Engineering Lab

Programming the Joystick TopHat

The Joystick TopHat - The joystick TopHat is programmable just like other buttons. The TopHat is accessed with the reserve word:

`joystick.joy1_TopHat`

The TopHat returns values from -1 to 7. The -1 value signifies that nothing is pressed.

See the program example below to see how the Joystick TopHat is programmed.



Front View
`joystick.joy1`

Program Example

```

1 void MoveForward()
2 void TurnRight()
3 void TurnLeft()
4 void Stop()
5
6 #include "JoystickDriver.c"
7
8 task main()
9 {
10     while(1 == 1)
11     {
12         getJoystickSettings(joystick);
13         if(joystick.joy1_TopHat == 0)
14         {
15             MoveForward();
16         }
17         if(joystick.joy1_TopHat == 2)
18         {
19             TurnRight();
20         }
21         if(joystick.joy1_TopHat == 6)
22         {
23             TurnLeft();
24         }
25         if(joystick.joy1_TopHat == -1)
26         {
27             Stop();
28         }
29     }
30 }

```

The functions MoveForward, TurnRight, TurnLeft, and Stop are not complete in this example. This is only to show what is possible using buttons, joystick axes, and the TopHat controls.

If the top switch of the TopHat is pressed, then call the function MoveForward.

If the right switch on the TopHat is pressed then call the function TurnRight.

If the left switch on the TopHat is pressed then call the function TurnLeft.

If nothing on the TopHat is pressed then call the function Stop.