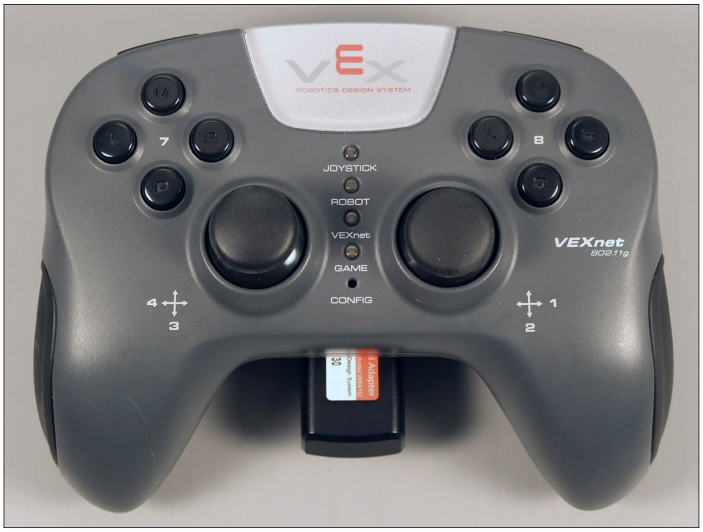
Remote Control Buttons

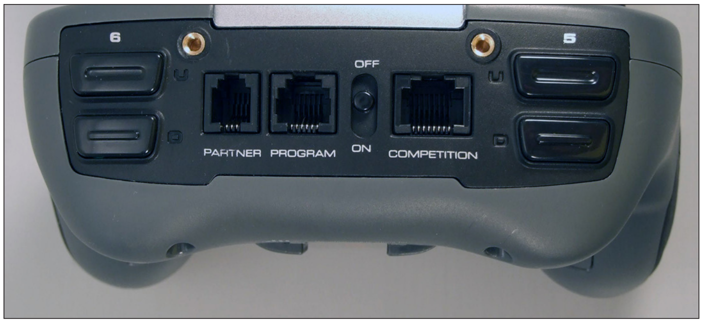
In this exercise you will:

1. Program the buttons on your remote controller.
2. Identify the names and locations of all buttons on the VEXnet Remote Control.

Remote Control Overview

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





Joysticks: Each remote control has two josticks. They are the round knobs that are labeled 1+2 and 3+4 on the picture on the left. To access the y-axis of right joystick the command would be "**`vexRT[Ch2]`**".

The joystick axis names are:

`Ch1`
`Ch2`
`Ch3`
`Ch4`

Note: ROBOTC has the capability of working with two remote controls at a time. Names for the second remote control are appended by Xmtr2. For example, to access the y-axis of right joystick on the second remote control, the command would be "**`vexRT[Ch2Xmtr2]`**".

Buttons: There are 12 programmable buttons on the remote control. The eight buttons on the front are broken into two groups of four, each having up, down, left, and right buttons. Two groups of up and down buttons make up the additional four buttons on the top of the remote control.

Accessing button values in ROBOTC is very similar to accessing joystick values. The vexRT[] command is still used, but now you use the letters "Btn", followed by the group number it belongs to, and finally the letter U, D, L, or R, depending on the buttons direction. For example, if you wanted to access the value of the down button on the front-left of the remote control you would use **`vexRT[Btn7D]`** short for vexRT Button Group 7 Down.

Note: Button names for the second remote control are also appended by Xmtr2. For example, to access the value of the down button on the front-left of the remote control, the command would be "**`vexRT[Btn7DXmtr2]`**".

Engineering Lab

Remote Control Review

This exercise assumes you understand how VEXnet communications works, that you have programmed your robot to move around using the joysticks, and that you have slowed down your robot's speed using different control mappings. If you have not completed those lessons, do so before beginning this exercise.

Things to Remember with Every Remote Control Program

1. Every remote control program will use the vexRT[] command to access the values of the joysticks and buttons.
2. Remote control commands for the robot must be placed in a while loop for the human operator to maintain smooth, continuous control.

Code Review

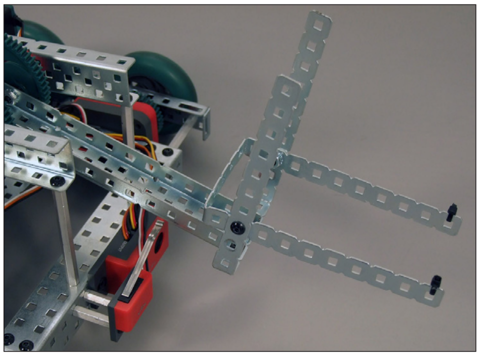
```
#pragma config(Motor, port2, rightMotor, tmotorNormal, openLoop, reversed)
#pragma config(Motor, port3, leftMotor,  tmotorNormal, openLoop)
//*!!Code automatically generated by 'ROBOTC' configuration wizard !!*//

task main()
{
  while(1 == 1)
  {
    motor[leftMotor] = vexRT[Ch3] / 2;
    motor[rightMotor] = vexRT[Ch2] / 2;
  }
}
```

Programming Remote Control Buttons

Remote control buttons behave the same way as the VEX touch sensors. When a touch sensor is pressed, it returns a value of "1". When it is not pressed, it returns a value of "0". Similarly, buttons return a value of "1" when pressed and a value of "0" when not pressed.

We will be using the remote control buttons to move the arm connected to the VEX robot. If you have not built the arm attachment, build it now using the instructions found in the Setup Section.



Once you have the robot arm attached, we will program two new behaviors. The new behaviors will be "Lift the Arm" and "Lower the Arm". To activate the arm, we will use the Up and Down buttons in Group 6, on the top of the remote control.

Engineering Lab

Programming Remote Control Buttons

```
 1  #pragma config(Motor, port2, rightMotor, tmotorNormal, openLoop, reversed)
 2  #pragma config(Motor, port3, leftMotor,  tmotorNormal, openLoop)
 3  #pragma config(Motor, port6, armMotor,   tmotorNormal, openLoop)
 4  //*!!Code automatically generated by 'ROBOTC' configuration wizard   !!*//
 5
 6  task main()
 7  {
 8    while(1 == 1)
 9    {
10      motor[leftMotor] = vexRT[Ch3] / 2;
11      motor[rightMotor] = vexRT[Ch2] / 2;
12
13      if(vexRT[Btn6U] == 1)
14      {
15        motor[armMotor] = 40;
16      }
17      else if(vexRT[Btn6D] == 1)
18      {
19        motor[armMotor] = -40;
20      }
21      else
22      {
23        motor[armMotor] = 0;
24      }
25    }
26  }
```

The values of the remote control buttons are accessed using reserved words in the form:

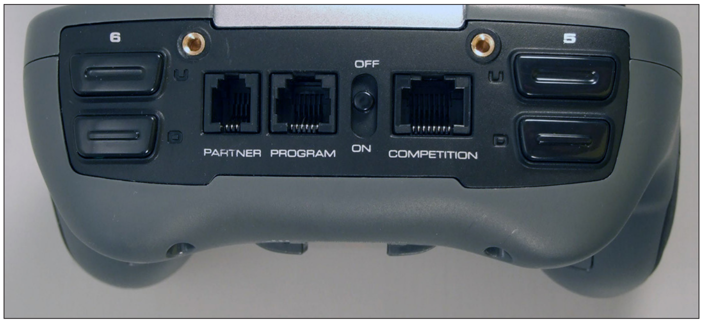
Btn#X

"Btn" is short for button.

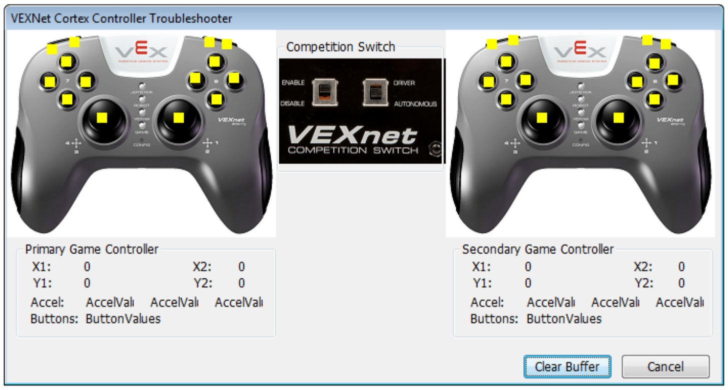
"#" is replaced by the button group number (5, 6, 7, or 8).
"X" is replaced by the direction of the button - U, D, R, or L (Up, Down, Right and Left).

Line 13 references remote control button 6U, and line 17 references to remote control button 6D.

What do lines 13 through 24 do? Experiment by commenting that section of code out and seeing what happens.



Copy the program above into ROBOTC and download it to your robot. Make sure that the motor for the arm is connected to Port 6. When you press buttons 6U and 6D, the arm will move up and down.



Primary Game Controller
X1: 0 X2: 0
Y1: 0 Y2: 0
Accel: AccelVal AccelVal AccelVal
Buttons: ButtonValues

Secondary Game Controller
X1: 0 X2: 0
Y1: 0 Y2: 0
Accel: AccelVal AccelVal AccelVal
Buttons: ButtonValues

Note: If you have your robot connected to the computer over VEXnet, you're able to view all of the remote control values using the Remote Control Troubleshooter. To open the Troubleshooter, go to Robot > Remote Control Troubleshooter > VEXnet Cortex Controller.