

---

# Activity 3.1.5 Variables and Functions – VEX

---

## Introduction

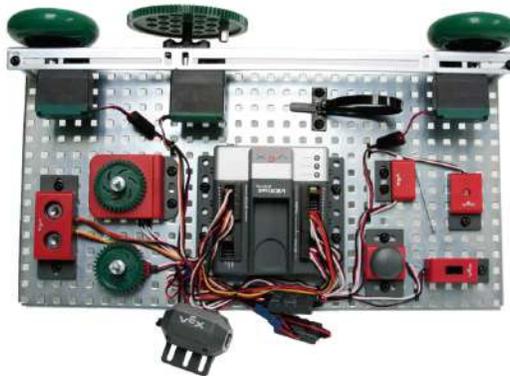
A program can accomplish a given task in any number of ways. Programs can quickly grow to an unmanageable size so variables and functions provide a technique to reduce the size of the program. A variable is a space in your robot's memory where you can store data, such as whole numbers, decimal numbers, and words. Functions group together several lines of code, which can then be referenced many times in task main, or even other functions. The use of variables and functions allows for complex control of a system.

## Equipment

- Computer with ROBOTC software
- POE VEX® testbed
- PLTW ROBOTC template

## Procedure

1. Connect the POE VEX testbed Cortex to the PC.

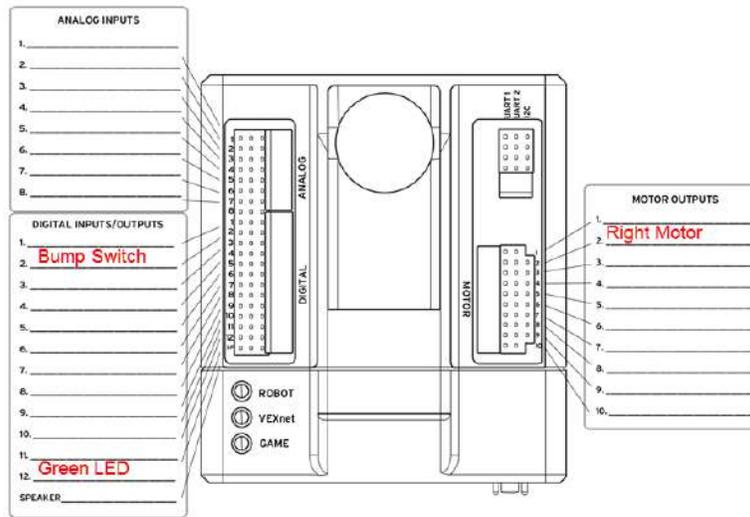


POE VEX Testbed

2. Open the PLTW ROBOTC template. Click File, Save As, select the folder that your teacher designated for you to save your ROBOTC programs in, then name the file A3\_1\_5\_Part1.
3. In this activity you will use the green LED, right motor, and bumper switch. Leave the previously connected motors and sensors wired to the Cortex. Go to the Motors and Sensors Setup window. Configure the Motors and Sensors Setup to reflect the inputs and outputs to be used. Note that additional motors and

sensors that are physically attached may be configured; however, these are not required to be configured. Click OK to close the window.

### Cortex Wiring Diagram



4. Copy and paste or create the program below in the `task main()` section of the program between the curly braces.

```
int motorcount;
motorcount = 0;
while (motorcount < 3)
{
    startMotor(rightMotor, 95);
    wait(2);
    stopMotor(rightMotor);
    wait(2);
    motorcount = motorcount + 1;
}
```

5. Save the program, power on the Cortex, compile, and download the program. If you have any errors, check with your instructor to troubleshoot your program.
6. Press Start to run the program and observe the behaviors.
7. Document what this program would look like as pseudocode simple behaviors.
8. Open the PLTW ROBOTC template. Click File, Save As, select the folder that your teacher designated, and then name the file A3\_1\_5\_Part2.
9. Write a program that performs the behavior below. Refer to the while loop structure and Boolean Logic table below when developing the program.

Create a count-based while loop that turns on the LED for 3 seconds, then off for 3 seconds, 5 times.

10. Test the program and troubleshoot until the expected behavior has occurred. Save the program.
11. Open the PLTW ROBOTC template. Click File, Save As, select the folder that your teacher designated, and then name the file A3\_1\_5\_Part3.
12. Recreate the LED Control Function and run it on your testbed.

```
void LEDControl(); //Function prototype / Declaration
```

```
task main()  
{ // Program begins, insert code within curly braces  
  while (1==1)  
  {  
    LEDControl(); //Function call  
  }  
} //End of task main
```

```
void LEDControl() //Function definition  
{  
  if(SensorValue[bumpSwitch] == 1)  
  {  
    turnLEDOn(green);  
  }  
  else  
  {  
    turnLEDOff(green);  
  } //End of function
```

13. Test the program and troubleshoot until the expected behavior has occurred. Save the program.
14. Describe the behaviors observed.
15. Follow teacher direction and either print the programs or submit electronically with this activity.