Activity 3.1.4 While and If-Else Loop Systems – VEX

Introduction

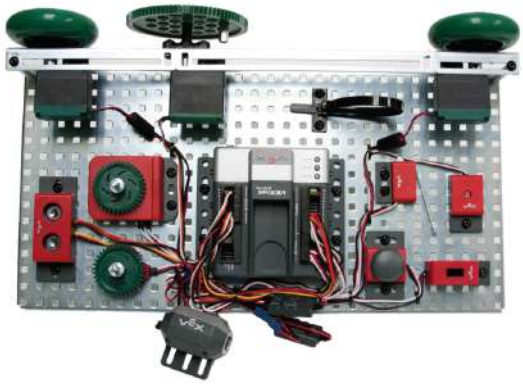
One of the powerful attributes of a computer program is its ability to make decisions. Although it can be argued that only humans are capable of decision making, computers are able to make decisions using criteria. They are able to compare two values and determine whether one is larger than the other. They can determine whether a statement is true or false, based on empirical data.

Equipment

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

Procedure

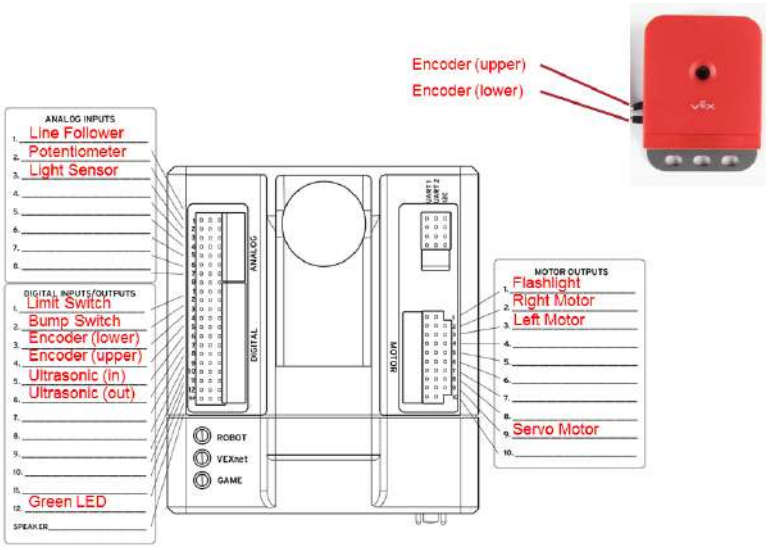
1.
2. Connect the POE VEX testbed Cortex to the PC.



POE VEX Testbed

3. 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_4_Part1.
4. In this activity you will use all of the testbed input and outputs. 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



5. A while loop is a structure within ROBOTC which allows a portion of code to be run over and over, as long as a certain condition remains true.

Below is the pseudocode outline of a while loop.

```
while(condition)
{
  // repeated-commands
}
```

(condition)
Either **true** or **false** (see Reference > Boolean Logic).

Repeated commands
Commands placed here will run over and over as long as the (condition) is **true** when the program checks at the beginning of each pass through the loop.

Below is an example of a program using an infinite While Loop.

```
task main()
{
 bMotorReflected[port2]=1;
 while(1 == 1)
 {
  motor[port3]=127;
  motor[port2]=127;
 }
}
```

The condition is true as long as 1 is equal to 1, which is always.

While the condition is true, both motors will receive full power.

6. Copy and paste or create the program below in the `task main()` section of the program between the curly braces. Note that the light threshold will vary depending on ambient light.

```
while(1 == 1)
{
   turnFlashlightOn(flashlight, 127);
   untilDark(450, lightSensor);
   turnFlashlightOff(flashlight);
   untilLight(450, lightSensor);
}
```

7. 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.
8. Press Start to run the program and observe the behaviors.
9. Document what this program would look like as pseudocode simple behaviors.

10. Open the PLTW ROBOTC template. Click File, Save As, select the folder that your teacher designated, and then name the file A3_1_4_Part2.
11. Write a program that performs the behavior below. Refer to the while loop structure and Boolean Logic table below when developing the program.

    Using the Ambient Light Sensor, program the flashlight to turn on when the lights in the room go out (or the sensor is blocked), and turn back off when the lights are turned back on, while the limit switch is pressed.

```
while (condition)
{
  body
}
```

While Loop

ROBOTC Symbol	Meaning	Sample comparison	Result
==	"is equal to"	50 == 50	true
		50 == 100	false
		100 == 50	false
!=	"is not equal to"	50 != 50	false
		50 != 100	true
		100 != 50	true
<	"is less than"	50 < 50	false
		50 < 100	true
		100 < 50	false
<=	"is less than or equal to"	50 <= 50	true
		50 <= 100	true
		50 <= 0	false
>	"is greater than"	50 > 50	false
		50 > 100	false
		100 > 50	true
>=	Greater than or equal to	50 >= 50	true
		50 >= 100	false
		100 >= 50	true

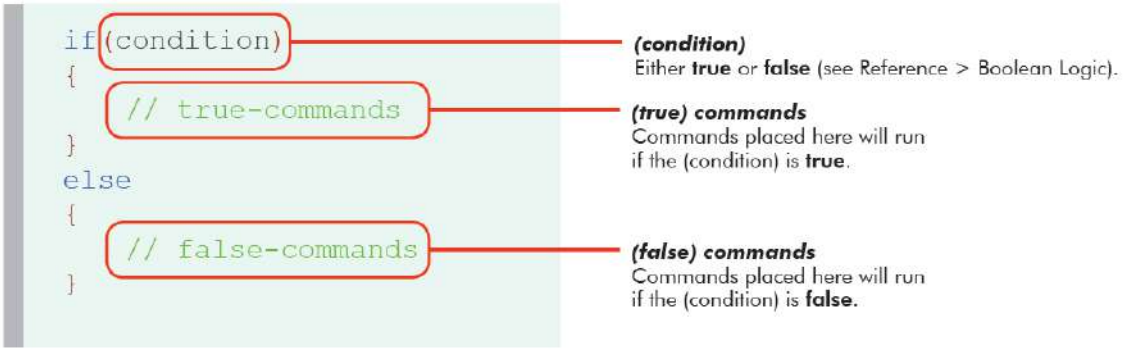
12. Test the program and troubleshoot until the expected behavior has occurred. Save the program.
13. Open the PLTW ROBOTC template. Click File, Save As, select the folder that your teacher designated, and then name the file A3_1_4_Part3.
14. Copy and paste or create the program below in the `task main()` section of the program between the curly braces.

```
ClearTimer(T1);
while (time1[T1] < 20000)     //Loop
program for 20 sec
{
  turnLEDOn(green);
  wait(2);
  turnLEDOff(green);
  wait(2);

}
```

15. Download and run the program. Observe the behaviors and document what this program would look like as pseudocode simple behaviors.

16. Modify your program to perform the following pseudocode. Program the greenLED to repeatedly turn on for 2 seconds, then off for 2 seconds, forever.
17. Test the program and troubleshoot until the expected behavior has occurred.
18. An if-else Statement is one way to allow a computer to make a decision. With this command the computer will execute one of two pieces of code, depending on whether the condition is true or false.



19. Open the PLTW ROBOTC template. Click File, Save As, select the folder that your teacher designated, and then name the file A3_1_4_Part4.
20. Copy and paste or create the program below in the `task main()` section of the program between the curly braces.

```
while (1 ==1)    //Loop program indefinitely
{
  if (SensorValue(bumpSwitch) == 1)
  {
    turnLEDOn(green);
  }
  if (SensorValue(bumpSwitch) == 0)
  {
    turnLEDOff(green);
  }
}
```

21. Download and run the program. Observe the behaviors and document what this program would look like as pseudocode simple behaviors.

22. Modify your program to use an if-else statement as shown below.

```
if (condition)
{
  statement;
else
{
  statement;
     If-Else Structure
```

23. Test the program and troubleshoot until the expected behavior has occurred. Add comments in your program to explain the purpose of each step. Print the program to submit with this activity. Save the program.
24. Follow the teacher direction and either print the programs or submit electronically with this activity.