CE-8 Lab 3: Small Introduction to Programming

Lab Objectives

By the end of this lab you should be able to:

1. **Identify** the different conditional operators and how to use them properly in the context of a program.
2. **Identify** what a variable is and how one is used in a program.
3. **Create** a program that properly uses `IF, THEN, ELSE` commands to direct the behavior of your robot.

Finally, you are in control of your robot. This lab is relatively short if you get the idea.

Variables

When making a program, it is very useful to keep track of and use values that you generate or get from a sensor. The way this is done in programs is by using variables. Variables in computer programming are a bit different than what you might be used to, they are not the same as variables in a math equation. Although, variables in programs do have names like in math equations. In math equations, variables represent unknowns, in a program the value of the variable is known and can be used to keep track of some information that we are gathering. For example, I could have a variable named `robotdistance` that would keep track of how far my robot has traveled in the course of a program. Where `robotdistance` is the name of the variable.

The first step in using variables in programs is to *declare* your variable, we do this by using the command `VAR`. Lets for example make a variable called `xyz`:

```plaintext
xyz VAR Word
```

This command tells the program to make a variable called 'xyz', the 'Word' part I'll explain later. It is important that the name you choose not be the same as another command or variable that is already in use, for example you can't make a variable called 'END'. The 'Word' argument tells the program how much space to use for our variable, there are a few options here and they all specify the size in terms of bits. Here are your options:

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>1-bit</td>
</tr>
<tr>
<td>Nib</td>
<td>4-bits</td>
</tr>
<tr>
<td>Byte</td>
<td>8-bits</td>
</tr>
<tr>
<td>Word</td>
<td>16-bits</td>
</tr>
</tbody>
</table>
In the chip that we are using for our robot, we can have a maximum of 13 'Word' size variables in use at any given time. The maximum value that a 16-bit variable can take is 65,535, if you try to set a variable to be bigger then this you'll start running into problems.

You can use variables in expressions to set their values. Its important to remember when working with expressions that the right hand side of the expression is evaluated first. For example, if want our variable 'xyz' to be equal to 5 plus 3 we would simply write.

\[ xyz = 5+3 \]

This statement first computes 5 plus 3 and then sets the variable 'xyz' to that value. We can also use variables in the right hand side equation, even if its the same variable as the one on the left hand side. For example:

\[
xyz = 5+3 \\
xyz = xyz*2
\]

What do you suppose the value of 'xyz' is? Lets look at these two commands in the way our robot will look at them. The first thing it does will add 5 and 3, it will then set 'xyz' to be equal to this value which is 8. It will then take the value of 'xyz', which is 8, and multiply it by 2, which is 16. It will then set 'xyz' to be this value, so we've effectively doubled the value of 'xyz' with this command.

**IF...THEN**

One of the main tools used by programmers is the **IF...THEN** statement. The **IF...THEN** statement is a simple way to direct the behavior of our robot based on the truth of some statement. The basic syntax of the **IF...THEN** statement is as follows:

```
IF condition THEN
    some action we want the robot to perform
ENDIF
```

We can read the syntax of the **IF...THEN** statement as the same way we tell someone of some action we might perform. For example, 'If it's sunny outside then I will go for a walk'. Notice the similarities in this sentence and that of our **IF...THEN** statement. What is the condition that I will go for a walk in this sentence? The condition could be anything we want it to be, in our robot we usually will use a condition based on one of our sensors or perhaps what the robot is currently doing or even the value of some variable. The ways in which we test a condition is by using comparison operators, the main operators that we use are.
<table>
<thead>
<tr>
<th>Comparison Operator Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equal</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Not Equal</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater Than</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less Than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater Than or Equal To</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less Than or Equal To</td>
</tr>
</tbody>
</table>

You might recognize some of this operators from your algebra class. Here is an example of a working `If...Then` statement. The variable 'xyz' will be the value of one of our light sensors, when the light value get above 10,000 we will have one of our LED's turn on.

```
IFF xyz > 10000 THEN
   HIGH 9
ENDIF
```

You can find more about the `If...Then` statement in the help documentation of the PBASIC editor.

**Exercises**

1. Using a `Do...Loop`, keep track of how many times the loop executes and use the `DEBUG` command to print out the number of executions, this should be a very short program.
2. Using a `Do...Loop`, keep track of how many times the loop executes and turn on the center LED after the loop executes 2,000 times.
3. Using a `Do...Loop`, keep track of how many times the loop executes and turn on the center LED after the loop executes 20 times but this time have the outer lights blink until the center light turns on. (Hint: Lookup 'ELSE' in the help documentation)
4. Using a `Do...Loop`, let your LED’s blink from the left to right, and the right to left like the KITT in Knight Rider.

For next time, you will learn to move your robot. It would be helpful for you to read documents provided before you come to the lab.