Make a Decision

Submitted by Andy Lindsay on Thu, 03/21/2013 - 16:32

Lesson edited to work with **CodeBlocks** IDE by Jeff La Favre 3/12/26

*(Updated 2013-08-08 for SimpleIDE 0.9.4 and its Learn folder's Simple Libraries and Examples) [SimpleIDE is the IDE for use with the robot. This lesson is edited so that we can use **CodeBlocks** as the IDE, which does not require a robot – J. La Favre]*

A common program decision is what to do with an output, such as a light, speaker or motor, based on one or more inputs, such as measurements from sensors.

Next, we will look at some example programs that make decisions based on the values stored by certain variables. After you are through with this primer and into programs for simple circuits, you will see many examples of decisions based on sensor inputs.

This example program initializes and then displays the values of variables named `a` and `b`. After that, it checks if `a` is larger than `b`, and if so, it prints a message saying so.

- Start **CodeBlocks** (or the IDE of your choice).
- Open the **File** menu and select **New**. Then select **Empty File**.
- Click the mouse in the text window of **CodeBlcks** and use the keyboard to enter the following text: **#include <stdio.h>**
- Open the **File** menu and select **Save file**, which opens a **Save file** dialog box.
- In the dialog box, open the drop-down labeled **Save as type** and select **C/C++ files**. In the **file name** slot enter this name for the file: **make a decision.** At the top of the dialog box there is a **Save in** slot, which determines where the file will be saved. Make sure you know the location where you are saving your file so that you can find it later. Now click the **Save** button to save your program file.

- Copy the text in the box below and paste it into the text window of **Dev-C++** under the first line of text you have already entered. Alternatively, use the keyboard to enter the text.
- Click the **Save** button to save the code you just pasted or entered with keyboard.
- Examine the program. Do you expect it to display one message, or two?
- Run the program by opening the **Execute** menu and selecting **Compile and Run**. If there are no errors in the program, a new program window will open. Compare the actual output to your predicted output.
- What do you think will happen if you swap the values of `a` and `b`? Try it, and then re-run the program.

```
int main()
{  // marks the beginning of code block for main()
  int a = 25;
  int b = 17;
  printf("a = %d, b = %d\n", a, b);
  if(a > b)   //execute the printf below here only if variable a has a value larger than variable b
  {  // marks the beginning of code block for if(a > b)
    printf("a is larger \n");
  }  // marks end of code block for if(a > b)
}  // marks the end of code block for main()
```

How the Code Example Works

The program initializes `a` to 25 and `b` to 17 and displays both those values. Then, it uses `if(a > b)` to decide whether to execute the contents of a code block, which contains the `printf("a is larger \n")` statement. If `a` is greater than `b`, it will print the second message. If it's equal to or less than `b`, it will not print the message.

Did You Know?

The > symbol is called a *relational operator*. Here is a list of relational operators and their meanings:

==	Equal to
!=	Not equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to

In C language, the expression `a > b` returns 0 if it is false (`a` is not really greater than `b`) or 1 if it is true (`a` is greater than `b`). So, in `if(a > b){...}`, the block of code inside the braces gets executed when `a > b` evaluates to 1, as in `if(1){...}`. The code block does not get executed if `a > b` returns 0, as in `if(0){...}`.

Try This

Comparisons with negative numbers can be interesting.

- Try changing `a` to –25 and `b` to –17.
- Think about which number is larger.
- Re-run the program and check the result.

Your Turn

- Expand your program with more **`if`** code blocks that test each of the relational operators introduced in the Did You Know section.