



Unit 3: BRIGHTNESS, IF, and WHILE

Skill Builder 2: BRIGHTNESS & LIGHT with IF, WHILE

In this second lesson of Unit 3, we'll develop an automatic light switch that responds to ambient light, turning on when darkness falls and turning off when the amount of light increases.

Objectives:

- **READ BRIGHTNESS**
- Use a **While** loop
- Use **If...Then...Else...End** to turn the LIGHT on or off depending on BRIGHTNESS

Now write a program that detects the BRIGHTNESS value and turns a light on when it gets 'dark'. When the room lighting gets brighter, the light goes off. This is exactly how many automatic light switches and night lights work.

Our program will read the light sensor on the TI-Innovator™ Hub and turn the onboard **LIGHT ON** whenever the brightness value falls below a certain level and turn it off when the brightness is above that level.

Use your brightness meter from the previous lesson to determine a suitable 'middle value' between 'lightness' and 'dark'.

Setting up the Program

1. Start a new program, and name it BRIGHT2.
2. Add the commands **ClrHome** and **Disp** and an informational title after **Disp**, as shown to the right.
3. Initialize the variable **B** by adding the statement **2→B**.
4. Add a **While** loop with the condition **B>1**. (The brightness value is very low.)
5. Add **End** to close the while loop.

```
NORMAL FLOAT AUTO REAL Radian MP
PROGRAM: BRIGHT2
:ClrHome
:Disp "AUTO LIGHT!"
:2→B
:While B>1
:
:
:
:End
:
```

To terminate the loop and the program, cover the light sensor.

6. In the **While** loop body, add **Send("READ BRIGHTNESS")** and **Get(B)** from the **prgm** HUB menu as shown.

```
NORMAL FLOAT AUTO REAL Radian MP
PROGRAM: BRIGHT2
:ClrHome
:Disp "AUTO LIGHT!"
:2→B
:While B>1
:Send("READ BRIGHTNESS ")
:Get(B)
:
:
:
:
```

If Statements

If statement will have two 'blocks' of code: one for when the condition is true and another for when the condition is false.

The structure of the *multi-line* statement is

```
NORMAL FLOAT AUTO REAL Radian MP
PROGRAM: BRIGHT2
:Send("READ BRIGHTNESS ")
:Get(B)
:If
:Then
:
:Else
:
:End
:End
```

If <condition>

Then

<do this when true>

Else

<do this when false>

End



You can add more blank lines by pressing **[ins]** **[enter]** or by using the **F5** (a %) menu option **Insert Line Above**.

Writing the Condition

The brightness value is stored in the variable **B** and ranges from 0 to 100.

What is a good 'dark' value? We chose 25, but you can change it to any value between 0 and 100. Use your brightness meter from the previous lesson to decide on a lightness-darkness boundary value.

You could improve the program by using an **Input** statement for the 'trigger' value. Just be sure to use the **Input** statement before the **While** loop starts.

The '<' (less than) operator is located in the **[test]** menu (**[2nd]** **m**).

1. Turn the **LIGHT ON** or **LIGHT OFF** in the **Then** and **Else** blocks as shown.
2. Run the program with the TI-Innovator Hub attached.
3. Control the light hitting the sensor, and watch the LIGHT (the red LED on the Hub) turn on or off.

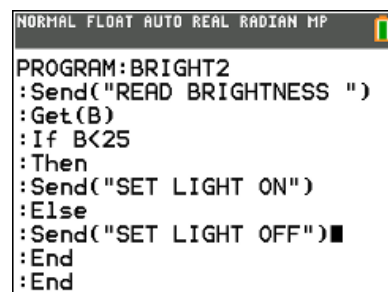
It might help to add an **Output**(statement from the previous lesson to display the value of **B** on the calculator screen and add statements in the If Then Else blocks to show whether the light is "ON" or "OFF". You can use something similar to:

Output(9,1,"ON ") and **Output(9,1,"OFF")**

One final note: The light will be ON when the program ends. Why? Add a statement to make sure that the light is off.



```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:BRIGHT2
:Send("READ BRIGHTNESS ")
:Get(B)
:If B<25
:Then
:Else
:End
:End
```



```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:BRIGHT2
:Send("READ BRIGHTNESS ")
:Get(B)
:If B<25
:Then
:Send("SET LIGHT ON")
:Else
:Send("SET LIGHT OFF")
:End
:End
```