



In this second lesson of Unit 3, we'll develop an automatic light switch that responds to the 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 **Text**, a set of quotations marks, and the text "Auto Light: Press enter".
3. Initialize the variable **b** by adding the statement **b:=2**.
4. Add a **While...EndWhile** loop with the condition **b>1**. (The brightness value is very low.)

```

1.1 1.2 1.3 *Brightnes...ter RAD 5/6
* bright2
Define bright2()=
Prgm
local b
Text "Auto Light: Press enter"
b:=2
While b>1
[]
EndWhile
EndPrgm

```

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

5. In the **While** loop body, add **Send "READ BRIGHTNESS"** and **Get b** from the **Hub** menu as shown.
6. Add **DispAt 1, b** to see the reading

```

1.1 1.2 1.3 *Brightnes...ter RAD 8/9
* bright2
local b
Text "Auto Light: Press enter"
b:=2
While b>1
Send "READ BRIGHTNESS"
Get b
DispAt 1, b
[]
EndWhile
EndPrgm

```

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

```

If <condition> Then
  <do this when true>
Else
  <do this when false>
EndIf

```

You can add more blank lines anywhere by pressing **enter**.

```

1.1 1.2 1.3 *Brightnes...ter RAD 9/13
* bright2
Send "READ BRIGHTNESS"
Get b
DispAt 1, b
If Then
[]
Else
[]
EndIf
EndWhile
EndPrgm

```

Teacher Tip: Note that Else and EndIf belong on their own lines.



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 a **Request** statement for this 'trigger' value. Just be sure to use the **Request** statement before the **While** loop starts.

The '<' (less than) operator is accessed by pressing **ctrl + =**.

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 and off.

Optional:

Add **DispAt 2**, "**Light is ON**" when the light is on and **DispAt 2**, "**Light is OFF**" when the light is off *in the appropriate places in the program*.

To stop the loop (and the program), cover the light sensor completely so that the brightness reading falls below 1.

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

```

1.1 1.2 1.3 *Brightness...ter RAD
* bright2 8/13
Send "READ BRIGHTNESS"
Get b
DispAt 1, b
If b<25 Then
:
:
Else
:
:
EndIf
EndWhile
EndPrgm

```

```

1.1 1.2 1.3 *Brightness...ter RAD
* bright2 11/13
Send "READ BRIGHTNESS"
Get b
DispAt 1, b
If b<25 Then
  Send "SET LIGHT ON"
Else
  Send "SET LIGHT OFF"
EndIf
EndWhile
EndPrgm

```

```

1.2 1.3 1.4 *Brightness...ter RAD
Light is ON Done
bright2()
0.299091
Light is ON Done

```

Teacher Tip: The light will be on at the end of the program because you have to get the brightness level below 1 so the program will be in the 'darkness' stage. Add **Send "SET LIGHT OFF"** just before the end of the program.