



Unit 3: BRIGHTNESS, IF, and WHILE

Skill Builder 1: BRIGHTNESS measurements

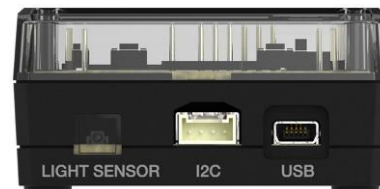
In this first lesson of Unit 3, we investigate the onboard light sensor, BRIGHTNESS, and introduce the **Output**(statement in TI-Basic to illustrate a technique for efficiently displaying numbers of different lengths using **Output**(.

Objectives:

- Read the BRIGHTNESS sensor
- Introduce the **While** loop
- Use the **Output**(statement
- Introduce **toString**(and concatenation

In the previous lessons, we have only been sending instructions to the TI-Innovator™ Hub to have an impact on its built-in devices (LIGHT, COLOR, and SOUND).

In this unit, we will work with the onboard light sensor and use the value in our program to create a 'light meter'. The light sensor produces values in the range 0 to 100 in decimal form.



Obtaining the light level value from the TI-Innovator Hub requires TWO statements:

- **Send**("READ BRIGHTNESS")
- **Get**(<var>)

Setting up the Program

1. Start a new program, and name it BRIGHT1.
2. Add the commands **ClrHome** and **Disp** to display the title as shown to the right.
3. Press **[prgm]**, and arrow over to the **HUB** menu.
4. Select **Send**("READ..." and then select **BRIGHTNESS**.
5. Press **[prgm]**, and arrow over to the **HUB** menu.
6. Select **Get**(. Enter the variable **B**, and the right parenthesis.

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:BRIGHT1
:ClrHome
:Disp "LIGHT SENSOR"
:
:Send("READ BRIGHTNESS")
:Get(B)
:
:
:
:
```

How it works:

- **READ BRIGHTNESS** tells the TI-Innovator Hub to read the brightness level and store that value in an onboard 'buffer'.
- **Get(B)** is a calculator command to get a value from the TI-Innovator Hub. This statement transfers the value in the buffer on the TI-Innovator Hub into the variable **B** in the TI-84 Plus CE. B can be replaced by any of the calculator's numeric variables, A...Z and Θ (theta).

Teacher Tip: A 'buffer' is a memory slot on the TI-Innovator Hub that temporarily stores a value. It is updated whenever another *READ* command is processed, so it is highly recommended to *READ* and immediately *Get* the value from the TI-Innovator Hub into a variable on the calculator. It is possible to collect a set of data from the TI-Innovator Hub and store that data in a list for future data analysis, but that is beyond the scope of this introduction.

While Loop

The **While...End** loop (**[prgm]** CTL menu) is used to process a block of code while a *condition* is true. A *condition* is a logical statement that can be evaluated as true or false. The relational operators and the logical operators are found on the **test** menu of the calculator (**[2nd]** **[math]**).

- The relational operators are =, \neq , <, >, \leq , and \geq .



10 Minutes of Code

TI-84 PLUS CE WITH THE TI-INNOVATOR™ HUB

UNIT 3: SKILL BUILDER 1

TEACHER NOTES

- The logical operators are **and**, **or**, **not**, and **xor**.
- These operators can be used together to build compound conditions such as **x>0 and y>0**.

Teacher Tip: There's another menu in the test menu: CONDITIONS. These are used in building piecewise functions with the piecewise function template. See [m](#) piecewise(.

```

NORMAL FLOAT AUTO REAL RADIAN MP
" { X^2; -3≤X and X≤0 "→Y1
  { X^3; 0<X and X<3

```

We are going to use a simple **While** loop that stops when the brightness value is less than 1. To terminate the program, simply cover the light sensor on the end of the TI-Innovator Hub with your hand.

Another way of 'breaking' or terminating a running program is to press the **ON** key. You will see the error message: **ERROR: BREAK** at the top of the screen and have the option of 1:Quitting to the Home screen or 2:Goto the program editor at the point where the action of pressing **ON** stopped the program. This is also a convenient way to continue editing the program.

```

NORMAL FLOAT AUTO REAL RADIAN MP
ERROR: BREAK
1:Quit
2:Goto
Action is Stopped.

```

Adding a While Loop

- Before the **Send(** statement in your program, add the statements:
2→B
While B>1 (Use the [test] menu for >.)
 - The statement **2→B** initializes the loop. As long as the condition **B>1** is true, the loop continues reading the light sensor. Once it becomes false, such as when there is no more light coming into the sensor, the loop and the program terminate.
- The **End** of the **While** loop must also be entered. Below the **Get(** statement, add an **End** statement for the end of the **While** loop. **End** is located in the **CTL** menu. (Press [prgm](#) to access the **CTL** menu.)

```

NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:BRIGHT1
:Disp "LIGHT SENSOR"
:2→B
:While B>1
:Send("READ BRIGHTNESS")
:Get(B)
:
:
:End
:

```

Teacher Tip: It's always a good idea to give your programs a way to get out of a loop. Recall that **End** is not the end of the program but rather it is the end of a control structure. (**If...Then...End**, **For...End**, **While...End**, and **Repeat...End**). In larger programs, there will be many instances of the End statement. The program processor in the calculator 'knows' which End belongs with which control structure, but it is up to the programmer to design the correct code.



10 Minutes of Code

TI-84 PLUS CE WITH THE TI-INNOVATOR™ HUB

3. Add the **Output(** statement *after* the **Get(** statement and *before* the **End** of the loop, as shown, by pressing `[prgm]`, arrowing over to **I/O**, and selecting **Output(**.
 - The **Output(** statement gives you great control over *where* on the HOME screen something should be displayed. The structure of the statement is: **Output(<line#>, <column#>, <string or variable>)**.

Examples:

- **Output(3,7,"HELLO")** will place the letter "H" at line 3, column 7 of the HOME screen and the rest of the word will follow the letter "H".
 - **Output(5,10,B)** will place the first digit of the value of the variable **B** starting at line 5, column 10 of the Home screen, and the rest of the digits follow.
4. Run the program with the TI-Innovator Hub attached.
 - You should see the title message at the top of the screen and a value in the middle of the screen that changes depending on the light intensity being read by the sensor. *Note that the value shown is not a standardized value.*
 - To end the program, cover the light sensor on the end of the TI-Innovator Hub so that the brightness value displayed is less than 1.

The **Output(** statement does not erase trailing digits when a shorter number is displayed after a longer number. For example, if one value displayed is 1.23456 and the next value displayed is 55, then you will see 5523456. Our final modification to this program is a tricky way of correcting this problem.

To correct the 'trailing digits' issue, we convert the variable **B** into a string and add some spaces to the end of it to completely erase the previous value displayed.

toString(is found on the `[prgm]` **I/O** menu.

The final correct **Output** statement is: **Output(5,10,toString(B)+" ")**

There are about 10 spaces between the quotes above.

Now, when you run the program, you will see that some values are shorter than others because the spaces that we added to the string representation of **B** erase the previous digits.

Concatenation:

The "+" sign used in the **Output(** statement is *not* used for addition; it is used to 'concatenate' (combine) two strings (sets of characters). The spaces (in quotation marks) are appended *to the end* of the string representation of the variable **B**.

Teacher Tip: String manipulation is not required for these lessons, but it should be clear from this example that knowing what features are available in a programming language greatly enhances the coding experience.

UNIT 3: SKILL BUILDER 1

TEACHER NOTES

```

NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:BRIGHT1
:Disp "LIGHT SENSOR"
:2→B
:While B>1
:Send("READ BRIGHTNESS")
:Get(B)
:
:Output(5,10,B)
:End
:

```

```

NORMAL FLOAT AUTO REAL RADIAN MP
LIGHT SENSOR

25.254712

```

```

NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:BRIGHT1
:Disp "LIGHT SENSOR"
:2→B
:While B>1
:Send("READ BRIGHTNESS")
:Get(B)
:Output(5,10,toString(B)+"      ")
:End
:

```

```

NORMAL FLOAT AUTO REAL RADIAN MP
LIGHT SENSOR

25.254712

```