



## Challenge: Design a Digital Mood Ring

### TI-NSPIRE™ CX

## TI-INNOVATOR™ STEM PROJECT

### TEACHER NOTES

#### Overview:

In this project, students build and program an introductory example of a feedback and control loop. This engineering principle is central to many industrial systems and consumer products. Feedback and control principles are also used in many Texas Instruments TI-Innovator™ Hub projects. During the project students will engage with fundamental concepts in programming, such as variables, loops, conditional statements and Boolean operators.

The project is presented as a series of challenges that build the conceptual knowledge and skills needed for the final open-ended challenge.

#### Goals:

Students will:

- create and edit a TI-BASIC program on the calculator.
- connect the Hub with a sensor to the calculator.
- write programs that include many commonly used Hub and calculator commands.
- build a basic feedback and control system.

#### Background:

Command	Example	Behavior
SET COLOR <r g b>	Send "SET COLOR 255 0 0"	Turns the color LED on with the color red.
DispAt <line #> , <"text"> , <variable name>	DispAt 3, "Temperature = ", t	When variable <i>t</i> has a value of 26, the following line is displayed on the calculator: Temperature = 26
CONNECT <type> <number> TO <port>	Send "CONNECT TEMPERATURE 1 TO IN1"	Associates the first TEMPERATURE object with a temperature module plugged into port IN1 on the Hub.
READ <type> <number>	Send "READ TEMPERATURE 1"	Reads a measurement from the first temperature object.
Get <variable>	Get t	Stores the temperature measurement into a variable named <i>t</i> . The value stored will contain the measurement from the immediately preceding READ command. Note: a Get command must immediately follow a READ command.
For <counter variable>,<start value>,<end value>,<step value>] <statements> EndFor	For n,1,10 DispAt 3,n EndFor	Runs For loop 10 times, starting at 1 and ending at 10. Executes the statement in the block each time, displays the value for the counter variable on row 3.
While <Boolean expression> <statements> EndWhile	key:="" While key!="esc" READ "TEMPERATURE 1" Get t DispAt 3,"Temperature= ",t key:=getKey() Wait 1	The statements in the While loop are executed until the escape key is pressed. The While loop continues as long as the Boolean expression evaluates to "true". The variable <i>key</i> is set to an initial value of an empty string using the assign function, ":=". getKey() is a function that returns a string with the name of the last key pressed while a programming is running. In this program the value of getKey() is stored to the variable <i>key</i> .

	<b>EndWhile</b>	
If <Boolean expression> Then <statements> EndIf	<b>If</b> $t > 26$ <b>Then</b> Send "SET COLOR 255 128 0" <b>EndIf</b>	Checks to determine if the value of variable $t$ is greater than 26. If the statement is true then the block of commands in the If..EndIf block are executed. Otherwise the block is skipped. In this case, when the temperature is greater than 26 the calculator will send a command to the TI-Innovator to set the COLOR to be yellow.
<Boolean expression> and <Boolean expression>	<b>If</b> $t \geq 22$ and $t < 24$ <b>Then</b> Send "SET COLOR 0 255 0" <b>EndIf</b>	When both expressions are true the "and" function is "true" and the statements are executed. Otherwise, the function returns false and the statements are skipped.

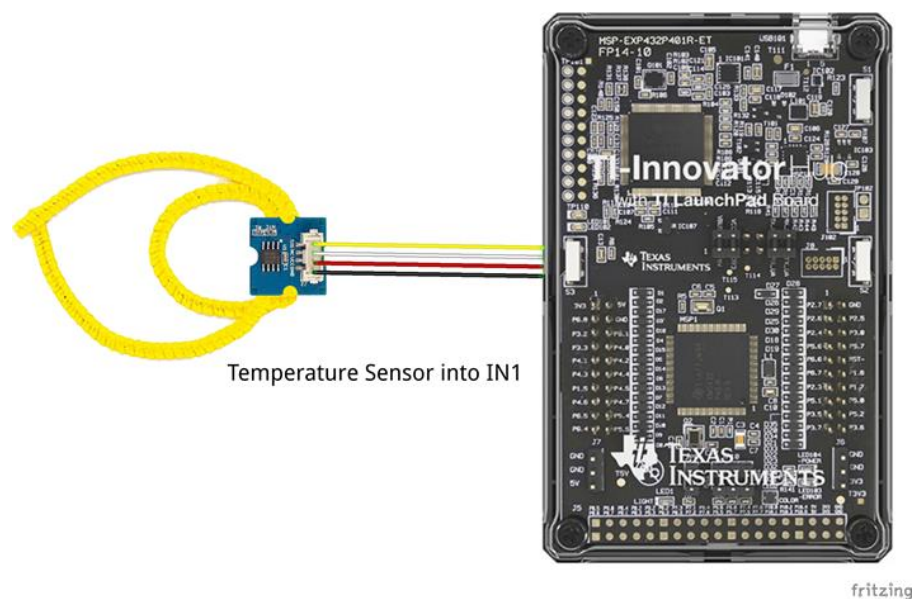
See TI-Innovator Technology eGuide for more background [https://education.ti.com/html/webhelp/EG\\_Innovator/EN/index.html](https://education.ti.com/html/webhelp/EG_Innovator/EN/index.html)

### Setup Project:

Students may work in groups of two or three.

### Supplies:

- Calculator
- Unit to Unit Cable
- Hub
- Temperature sensor
- Grove Cable
- Chenille Wire



Note: Students will need to push the wire through the holes on the temperature sensor board to create a ring. The sensing element is on the flat side of the sensor board. The student document shows a pipe cleaner, but it is not as pictured here. Expect (and encourage!) trial and error as students attempt to determine the most effective ring design.

**Student Activity:**

Sit in small groups with your calculator and supplies for this activity. Practice the guidance modeled by your teacher.

**Teacher Notes:**

Review and introduce the following calculator and Hub commands needed for this activity with students:

- Start a new document
- Insert a program
- Attach Hub

*Note: If students are new to the technology and/or coding, refer students to step-by-step lessons in the 10 Minutes of Code with TI-Innovator lessons to get started, especially Unit 1 Skill Builder 1: Color and Unit 1 Skill Builder 2: Sound.*

<https://education.ti.com/en/activities/ti-codes/nspire/10-minutes-innovator>

*Note: The set of videos at this link provide a detailed, step-by-step, demonstration of the project and the challenges.*

[https://www.youtube.com/playlist?list=PL17Fe0ZmhCR\\_i09b202PUi0wfaRYSLQka](https://www.youtube.com/playlist?list=PL17Fe0ZmhCR_i09b202PUi0wfaRYSLQka)

If some students are unfamiliar with Mood Rings or how they work, have students do a quick search on the internet for resources describing how a mood ring works. You may want to begin with this YouTube video describing how a mood ring works. <https://www.youtube.com/watch?v=gvgI51xOKTU&feature=youtu.be>

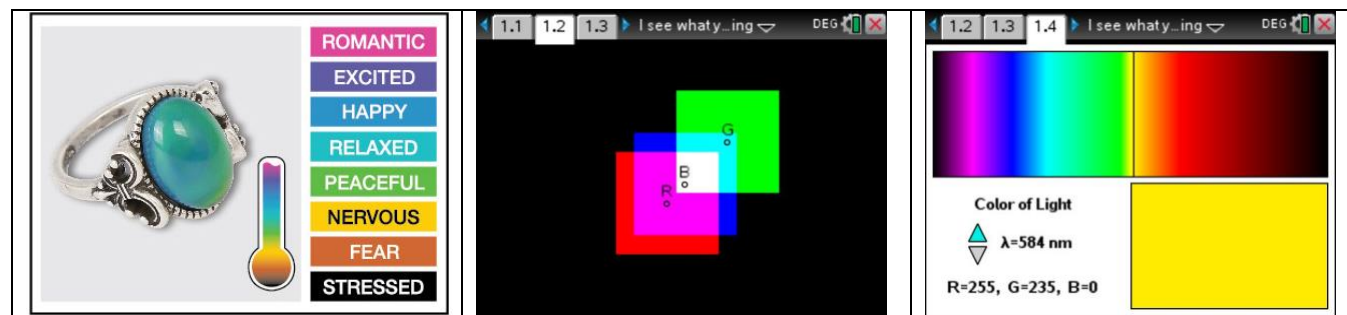
Optional: Distribute the associated Student Activity which contains background info and the Challenges listed here in this document. By doing so, student groups can work at their own pace through the Challenges, Alternatively, you may choose to display the Challenges one at a time in another way in your classroom.

**Challenge 1:** Use SET COLOR to explore using the color LED. Try to find the RGB values of all the colors in the mood chart.

e.g. Send “SET COLOR 255 128 0” will make yellow.

**Teacher Guidance during Challenge 1:**

- Demonstrate the color mixing simulations in the Color TI-Nspire file to assist in student understanding of additive color mixing and RGB values. Download Color.TNS at this [link](#).





## Challenge: Design a Digital Mood Ring

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT  
TEACHER NOTES

- Example program:

```
Define cla()=
Prgm
Send "SET COLOR 255 128 0"
EndPrgm
```

- Extension: Write a program that displays each of the colors that the mood ring will display.

```
Define clb()=
Prgm
Send "SET COLOR 255 0 0"
Wait 2
Send "SET COLOR 255 0 255"
Wait 2
Send "SET COLOR 255 113 0"
Wait 2
Send "SET COLOR 0 0 0"
EndPrgm
```

- Wait is useful to pause the program for a specified number of seconds.
- Try setting the color value to 0 0 0
- Try setting the color value to 255 255 255
- Try setting the color value to 1 1 1

**Challenge 2:** Use DispAt command to display your name at several locations on the screen.

### Teacher Guidance during Challenge 2:

- Use the DispAt command (Program I/O menu). DispAt takes at least two inputs (also known as “arguments”. The first input is the row number to display on. The second input is the text string, value, variable, etc. to display.
- The colon is found on the menu of characters, the key in the lower right labeled with “?!”.
- Example program:

```
Define c2()=
Prgm
DispAt 3,"Hello, my name is:"
DispAt 4,"Hollingsworth"
EndPrgm
```



## Challenge: Design a Digital Mood Ring

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT  
TEACHER NOTES

**Challenge 3:** Use a For..EndFor loop to display the numbers 1 through 10.

### Teacher Guidance during Challenge 3:

- Loops are used to repeat a set of commands. A For loop repeats a specified number of times. In TI-Basic the programmer defines a For loop with four inputs: a counter variable, a beginning value for the counter variable, an ending value for the counter variable and an optional step value variable.

- Example program

```
Define c3()=
Prgm
For n,1,10
  DispAt 3,n
  Wait 0.5
EndFor
EndPrgm
```

- The value for step is assumed to be 1 if a step value input is not included in the command.
- The set of commands to be repeated are defined within the For..EndFor block.

For...EndFor is found on the Program Control Menu

**Challenge 4:** Connect a temperature sensor to the TI-Innovator Hub and display the temperature on the calculator.

### Teacher Guidance during challenge 4:

- The Hub Connect commands provide the hub with information about the type of device plugged into external ports and how to communicate with that device.
- The Read command instructs the Hub to read a value from a device plugged into a specified port. The Hub places the value in a memory location on the Hub, waiting for the calculator to “get” the value.
- The Get command gets the value from the Hub memory location for values read and stores the value to a specified variable.
- Note that multiple arguments can be used to build a message with the DispAt command.
- Example program:

```
Define c4()=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN 1"
Send "READ TEMPERATURE 1"
Get t
DispAt 4,"Temperature=",t
EndPrgm
```



## Challenge: Design a Digital Mood Ring

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT  
TEACHER NOTES

**Challenge 5:** Use a loop to read and display temperature.

### Teacher Guidance during Challenge 5:

- Display the sample number (counter value) in addition to the temperature.
- Note that multiple inputs can be used to build a message with the DispAt command. Make sure to separate the arguments with commas. Note: Find the degree symbol in the symbol menu (ctrl-catalog book symbol key above divide).
- Example program:

```
Define c5()=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN 1"
For n,1,10
  Send "READ TEMPERATURE 1"
  Get t
  DispAt 3,"Sample Number=",n
  DispAt 4,"Temperature=",t,"°C"
  Wait 2
EndFor
EndPrgm
```

**Challenge 5 Extension:** Use a loop to read, display and log into list arrays for time and temperature. Graph the result.

### Teacher Guidance during Challenge 5 Extension:

- **Note:** This challenge is optional. Logging sensor readings to list variables for later analysis is useful. However, logging the data is not required for a working digital mood ring.
- Example program:

```
Define c5e()=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN 1"
time:={}
temp:={}
For n,1,10
  Send "READ TEMPERATURE 1"
  Get t
  DispAt 3,"Time=",n*0.5,"sec"
  DispAt 4,"Temperature = ",t,"°C"
```



## Challenge: Design a Digital Mood Ring

### TI-NSPIRE™ CX

## TI-INNOVATOR™ STEM PROJECT

### TEACHER NOTES

```
time[n] := n * 0.5
temp[n] := t
Wait 0.5
EndFor
EndPrgm
```

- The temperature value readings are stored to a variable *t*. Then during each For loop cycle the value for *t* is stored to the position *n* of the list variable named *temp*.
- The value of time is calculated by multiplying the number of For loop cycles, *n*, by the wait period, 0.5.
- The variables *time* and *temp* are initialized as blank lists in the beginning section of the program, using { }
- The variable *n* is used to set the position of the value within the list variables, *time[n]* and *temp[n]*.
- The value for *n* is incremented by 1 during each cycle of the While loop with this statement *n:=n+1*. The variable *n* is initially set to 1, the first position in TI-Basic lists, before the While loop begins.

**Challenge 6:** Use a While..EndWhile loop along with getKey() command to monitor temperature and make a decision to display a message, “Hot” or “Cold”.

Then modify your program to include a 3<sup>rd</sup> level between hot and cold, “Nice”.

#### Teacher Guidance during Challenge 6:

- While loops are useful when you would like a set of commands to be executed when a certain condition is true.
  - The While loop test checks to see if the variable named *key* is not equal to “esc”. If *key* is not equal to “esc” then the commands in the loop are run. If *key* is equal to “esc” then the program exits the While loop and moves to the next command after the EndWhile statement. In the example, the variable named *key* is initially set to be a blank string. This assures that the While loop will run at least once.
  - In the While loop a function called getKey() is used to update the value stored in the variable named *key*. getKey() is a TI-Nspire function that is useful for keyboard controlled While loops. getKey() stores the name of the last key that you pressed while a program is running. The key names are text string values, which are enclosed in quotes. For example, the While loop test checks for “esc” not esc.

- Example program:

```
Define c6a()=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN 1"
key:="" "
```



## Challenge: Design a Digital Mood Ring

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT  
TEACHER NOTES

```
While key≠"esc"
  Send "READ TEMPERATURE 1"
  Get t
  DispAt 4,"Temperature=",t,"°C"
  If t<22 Then
    Send "SET COLOR 0 0 255"
    DispAt 5,"It is cold"
  EndIf
  If t≥22 Then
    Send "SET COLOR 255 0 0"
    DispAt 5,"It is hot"
  EndIf
  key:=getKey()
  Wait 0.5
EndWhile
EndPrgm
```

- **Extension to include three levels of temperature:**
- The Boolean operator “and” is used to define levels of temperature. The “and” function can be found on the Catalog menu, the key with the open book just above the divide key. See the description of the “and” function in the top section of this document. The “or” function is another commonly used Boolean operator. “or” can also be found on the Catalog menu.

```
Define c6b()=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN 1"
key:=""
While key≠"esc"
  Send "READ TEMPERATURE 1"
  Get t
  DispAt 4,"Temperature=",t,"°C"
  If t<22 Then
    Send "SET COLOR 0 0 255"
    DispAt 5,"It is cold"
```





## Challenge: Design a Digital Mood Ring

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT  
TEACHER NOTES

```
EndIf
If t≥22 and t<25 Then
    Send "SET COLOR 0 255 0"
    DispAt 5,"It is nice"
EndIf
If t≥25 Then
    Send "SET COLOR 255 0 0"
    DispAt 5,"It is hot"
EndIf
key:=getKey()
Wait 0.5
EndWhile
EndPrgm
```

**Final Challenge:** Build a mood ring to repeatedly read the temperature sensor, determine the mood of the person, display the temperature value and display the mood.

### Teacher Guidance during Final Challenge:

- Have the students use the skills from the previous challenges.
- How many moods will the students have their programs check for?
- What values will the students use to define each mood level?
- Example basic version of the Digital Mood Ring program:

```
Define mood()=
Prgm
Send "CONNECT TEMPERATURE 1 TO IN 1"
key:=""
DispAt 8,"Press esc key to quit"
While key≠"esc"
    key:=getKey()
    Send "READ TEMPERATURE 1"
    Get t
    DispAt 4,"Temp =",t,"°C"
    If t<24 Then
        Send "SET COLOR 0 0 0"
        DispAt 5,"You are STRESSED"
```



## Challenge: Design a Digital Mood Ring

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT

TEACHER NOTES

```
EndIf
If t≥24 and t<25 Then
  Send "SET COLOR 255 0 0"
  DispAt 5,"You are NERVOUS"
EndIf
If t≥25 and t<26 Then
  Send "SET COLOR 255 150 50"
  DispAt 5,"You are UNSETTLED"
EndIf
If t≥26 and t<27 Then
  Send "SET COLOR 0 255 0"
  DispAt 5,"You are ACTIVE"
EndIf
If t≥27 and t<28 Then
  Send "SET COLOR 0 255 150"
  DispAt 5,"You are RELAXED"
EndIf
If t≥28 and t<29 Then
  Send "SET COLOR 0 255 255"
  DispAt 5,"You are LOVEABLE"
EndIf
If t≥29 and t<30 Then
  Send "SET COLOR 0 0 255"
  DispAt 5,"You are ROMANTIC"
EndIf
If t≥30 and t<31 Then
  Send "SET COLOR 255 100 75"
  DispAt 5,"You are HAPPY"
EndIf
If t≥31 and t<32 Then
  Send "SET COLOR 255 255 0"
  DispAt 5,"You are TIRED"
```



## Challenge: Design a Digital Mood Ring

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT  
TEACHER NOTES

```
EndIf
If t ≥ 32 Then
  Send "SET COLOR 255 0 255"
  DispAt 5, "You are CALM"
EndIf
Wait 0.5
EndWhile
Send "SET COLOR 0 0 0"
EndPrgm
```

- Example of an advanced program, which includes many optional extensions. See explanations below.

```
Define moodext()=
Prgm
Local a
a:=1
Local t
t:=0
Local room_tempsum
Room_tempsum:=0
Local room_temp
room_temp:=0
Local key
key:=""
Local n
n:=1
time:={ }
temp:={ }
Send "CONNECT TEMPERATURE 1 TO IN 1"
For a,1,8
DispAt a," "
EndFor
DispAt 1,"The Mood Ring"
```



## Challenge: Design a Digital Mood Ring

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT

TEACHER NOTES

```
DispAt 8,"Hit esc key to quit"
For a,1,10
Send "READ TEMPERATURE 1"
Get t
room_tempsum:=room_tempsum+t
Wait 0.1
EndFor
room_temp:=room_tempsum/10
DispAt 2,"Room Temperature=",round(,1),"°C"
While key≠"esc"
  key:=getKey()
  Send "READ TEMPERATURE 1"
  Get t
  DispAt 4,"Temperature=",round(t,1),"°C"
  If t<room_temp+1 Then
    Send "SET COLOR 0 0 0"
    DispAt 5,"You are STRESSED"
  ElseIf t≥room_temp+1 and t<room_temp+2 Then
    Send "SET COLOR 255 0 0"
    DispAt 5,"You are NERVOUS"
  ElseIf t≥room_temp+2 and t<room_temp+3 Then
    Send "SET COLOR 255 150 50"
    DispAt 5,"You are UNSETTLED"
  ElseIf t≥room_temp+3 and t<room_temp+4 Then
    Send "SET COLOR 0 255 0"
    DispAt 5,"You are ACTIVE"
  ElseIf t≥room_temp+4 and t<room_temp+5 Then
    Send "SET COLOR 0 255 150"
    DispAt 5,"You are RELAXED"
  ElseIf t≥room_temp+5 and t<room_temp+6 Then
    Send "SET COLOR 0 255 255"
    DispAt 5,"You are LOVEABLE"
```



## Challenge: Design a Digital Mood Ring

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT

TEACHER NOTES

```

ElseIf t≥room_temp+6 and t<room_temp+7 Then
  Send "SET COLOR 0 0 255"
  DispAt 5,"You are ROMANTIC"
ElseIf t≥room_temp+7 and t<room_temp+8 Then
  Send "SET COLOR 255 100 75"
  DispAt 5,"You are HAPPY"
ElseIf t≥room_temp+8 and t<room_temp+9 Then
  Send "SET COLOR 255 255 0"
  DispAt 5,"You are TIRED"
ElseIf t≥room_temp+9 Then
  Send "SET COLOR 255 0 255"
  DispAt 5,"You are CALM"
EndIf
Wait 0.5
time[n]:=n*0.5
temp[n]:=t
n:=n+1
EndWhile
Send "SET COLOR 0 0 0"
EndPrgm

```

- Example of an advanced program, which includes many optional extensions. See explanations below.

```

Define moodext()=
Prgm
Local a:a:=1
Local t:t:=0
Local room_tempsum:Room_tempsum:=0
Local room_temp:room_temp:=0
Local key:key:=" "
Local n:n:=1
time:={ }
temp:={ }
Send "CONNECT TEMPERATURE 1 TO IN 1"

```



## Challenge: Design a Digital Mood Ring

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT

TEACHER NOTES

```
For a,1,8
DispAt a," "
EndFor
DispAt 1,"The Mood Ring"
DispAt 8,"Hit esc key to quit"
For a,1,10
Send "READ TEMPERATURE 1"
Get t
room_tempsum:=room_tempsum+t
Wait 0.1
EndFor
room_temp:=room_tempsum/10
DispAt 2,"Room Temperature=",round(,1),"°C"
While key≠"esc"
  key:=getKey()
  Send "READ TEMPERATURE 1"
  Get t
  DispAt 4,"Temperature=",round(t,1),"°C"
  If t<room_temp+1 Then
    Send "SET COLOR 0 0 0"
    DispAt 5,"You are STRESSED"
  ElseIf t≥room_temp+1 and t<room_temp+2 Then
    Send "SET COLOR 255 0 0"
    DispAt 5,"You are NERVOUS"
  ElseIf t≥room_temp+2 and t<room_temp+3 Then
    Send "SET COLOR 255 150 50"
    DispAt 5,"You are UNSETTLED"
  ElseIf t≥room_temp+3 and t<room_temp+4 Then
    Send "SET COLOR 0 255 0"
    DispAt 5,"You are ACTIVE"
  ElseIf t≥room_temp+4 and t<room_temp+5 Then
    Send "SET COLOR 0 255 150"
```



```
DispAt 5,"You are RELAXED"
ElseIf t≥room_temp+5 and t<room_temp+6 Then
  Send "SET COLOR 0 255 255"
  DispAt 5,"You are LOVEABLE"
ElseIf t≥room_temp+6 and t<room_temp+7 Then
  Send "SET COLOR 0 0 255"
  DispAt 5,"You are ROMANTIC"
ElseIf t≥room_temp+7 and t<room_temp+8 Then
  Send "SET COLOR 255 100 75"
  DispAt 5,"You are HAPPY"
ElseIf t≥room_temp+8 and t<room_temp+9 Then
  Send "SET COLOR 255 255 0"
  DispAt 5,"You are TIRED"
ElseIf t≥room_temp+9 Then
  Send "SET COLOR 255 0 255"
  DispAt 5,"You are CALM"
EndIf
Wait 0.5
time[n]:=n*0.5
temp[n]:=t
n:=n+1
EndWhile
Send "SET COLOR 0 0 0"
EndPrgm
```

## Explanation of advanced program

- Variables can be declared as Local to the program. Variables declared as local can only be accessed from within the program that they are declared in. Global variables can be accessed from other programs and from any application in the calculator. Local variables are useful for avoiding inadvertent variable name conflicts and confusion. Also, local variables do not clutter the Vars menu with extra variables.
- Multiple program commands can be placed on the same line by separating the commands with a colon. In this case, the initial set up of variables is done on the same line with a Local command and an initial value.
- The program stores temperature and time values in lists for later analysis.



## Challenge: Design a Digital Mood Ring

TI-NSPIRE™ CX

TI-INNOVATOR™ STEM PROJECT

TEACHER NOTES

- The program uses a For..EndFor loop to assure blank rows for displaying the program outputs by displaying empty text strings to each of the 8 rows used by the DispAt command.
- The program takes into account the temperature in the environment by using a For..EndFor loop to calculate a mean temperature of 10 readings. The variable *room\_temp* is then used to shift the temperatures ranges of the mood categories.
- If-Then-Elseif commands are useful when there are multiple levels of mutually exclusive categories to check. Checking of the Elseif statements stops when an Elseif condition check is found to be true. An approach using a series of If..Then..EndIf statements will continue to check all of the statements in the loop. The Elseif construction saves the program from doing unnecessary checks.