Smart Irrigation System: Challenge #4

In this activity you will use a Digital Temperature and Humidity (DHT) sensor, moisture sensor, and light level sensor to read and display measurements from various locations.

Goals:

1. Use the read command to read temperature.
2. Use the read command to read relative humidity.
3. Use the read command to read light intensity.
4. Use the read command to read moisture levels in soil
5. Use a While loop to repeat code.
6. Use an If-Then statements to make decisions

Background:

In the previous three challenges, you learned about working with several kinds of sensors:

- You used a light sensor to measure light intensity. You proposed ideal levels for watering crops. Would these levels change based on location or crop?
- You used a soil moisture sensor to detect the moisture level of different types of soil. The soil type led to different infiltration rates, water retention and erosion. Would soil type influence the parameters you used for restriction on a smart watering system?
- The Digital Temperature and Humidity sensor measured both temperature and humidity. You learned to minimize the net transfer of liquid water to vapor, ideal watering would occur when the temperature was cool and the humidity high.

Now, you will use the combination of all of the above, to create a monitoring system that will display all the relative information to allow a user to determine if a crop should be watered. The goal of this challenge is to get all the sensors working, and displaying data. In the next and final challenge, not only will you use sensors to monitor light, moisture, humidity and temperature; you'll use these values to control a water pump to water only when necessary, ie. a "smart" watering system!

Command	Example	Behavior
CONNECT <type> <number> TO <port>	Send "CONNECT LIGHTLEVEL 1 TO IN1"	Associates the first LIGHTLEVEL object with a light sensor plugged into port IN1 on the Hub.
SET <type> <number> TO <value>	Send "SET ANALOG.OUT 1 TO 128"	Turns on an analog.out1 object, such as a pump, to a power setting of 128
RANGE <type> <number> <min value> <max value>	Send "RANGE LIGHTLEVEL 1 0 100"	Scales the measured values read from LIGHTLEVEL 1 to return in the range 0 to 100.
READ <type> <number>	Send "READ MOISTURE 1"	Reads one measurement from the first moisture sensor.
Get <variable>	Get m	Stores the moisture measurement into the variable named m. *Note a get command must immediately follow a read command. The value stored will contain the measurement from the immediately preceding READ command."

DispAt <line #> , <"text"> , <variable name>	DispAt 3, "Moisture level = ", m	When variable m has a value of 26, "Moisture level = 26" is displayed on line 3 of the calculator.

While <Boolean expression>
  <statements>
EndWhile
	key:=" "
	While key ≠ "esc"
	  Send "READ MOISTURE 1"
	  Get m
	  DispAt 3,"Moisture level = ",m
	  key:=getKey()
	EndWhile
		The commands inside the While structure are looped until the escape key is pressed. The loop continues while logical expression key ≠ "esc", is true. The variable *key* is initially assigned an empty value so the while loop will execute at least once. The getKey() function monitors the keypad and returns a string with the name of the last key pressed.

If <Boolean expression 1> Then
  <statements 1>
ElseIf <Boolean expression 2> Then
  <statements 2>
Else
  <statements 3>
EndIf
	If t >=40 Then
	  DispAt 3,"It is Hot"
	ElseIf t >=25 and t < 40 Then
	  DispAt 3,"It is Warm"
	Else
	  DispAt 3,"It is Cool"
	EndIf
		The example decision tree has two mutually exclusive Boolean expressions and two corresponding execution statements. It also has an Else condition that executes corresponding statements when neither of the first two conditions are true. This final Else condition ensures that a set of statements will always be executed. When this decision tree executes, focus proceeds from top-down. If the first Boolean expression is true, the corresponding statements are executed and the decision tree is immediately exited. In the example, if t=30 then the first expression is false and the <statements 1> are skipped, the second expression is true and <statements 2> are executed and the tree is exited. Additional ElseIf statements may be inserted if needed.

The DHT sensor requires a few seconds to "warm up" and begin communicating with the Hub. During this period, the temperature is reported to be absolute zero (-273 **°C)**.	Send "CONNECT DHT 1 TO IN 2"
	temperature:=−273
	DispAt 3,"DHT is Warming up"
	While temperature<−270
	  Send "READ DHT 1
	  TEMPERATURE"
	  Get temperature
	  Wait 2
	EndWhile
	DispAt 3,"DHT is now ready"
		The program uses a While loop to continuously read the DHT every two seconds until it reports a value greater than absolute zero. You will need to include this code in your project.

Challenge:

Write a program named *C4* using a While loop to continuously measure and display a dashboard of all sensor value readings. The user should be able to stop the monitoring by pressing the ESC key. (Don't forget to warm-up the DHT sensor in your code.)