



Smart Irrigation System: Challenge #4

Goals:

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.

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."



Challenge: Design Smart Irrigation System
TI-84 PLUS CE

TI-INNOVATOR™ STEM PROJECT
STUDENT ACTIVITY

<p>Output(<row> ,<column>, <"text">)</p> <p>While <Boolean expression> <statements> End</p>	<p>Output(1,3,"Hello World")</p> <p>0→k While k ≠ 45 Send("READ MOISTURE 1") Get(m) Output(1,1,"Moisture = ") Output(1,12,m) getKey→k End</p>	<p>Displays "Hello World" on row 1 column 3.</p> <p>The commands inside the While structure are looped until the escape key is pressed. The loop continues while logical expression $k \neq 45$, is true. The variable k is initially assigned a zero so the while loop will execute at least once. The getKey function monitors the keypad and returns a number for the last key pressed.</p>
<p>If <Boolean expression> Then <statements 1> Else <statements 2> End</p>	<p>If $t \geq 40$ Then Output(6,1,"It is Hot") Else If $t \geq 25$ and $t < 40$ Then Output(6,1,"It is warm") Else Output(6,1,"It is Cool") End End</p>	<p>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 If statements may be inserted if needed.</p>
<p>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).</p>	<p>Output(3,1,"DHT WARMING UP") -273→T While $T < -270$ Send("READ DHT 1 TEMPERATURE") Get(T) Wait 2 End Output(3,1,"DHT IS NOW READY")</p>	<p>The program uses a While loop to continuously read the DHT every two seconds until it reports a value greater than absolute zero.</p>

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.)

