2. CODE A COOLING BREEZE FOR A GREENHOUSE

If the weather is cold, bundle up before going outdoors. If it's hot, dress light. You just read two *conditional if-then statements* that use weather conditions to help you decide what to wear. Computer programs also use conditional if-then statements to make decisions. In this investigation, you will write a program that uses temperature measurements from inside a greenhouse as the condition to decide when to turn on a fan and how long it must run to keep a greenhouse at an ideal temperature. Besides temperature control, air movement near leaves helps the photosynthesis-related gases of carbon dioxide, oxygen, and water vapor mix with the surrounding air. Plants also need frequent air movement to help their stems grow stronger.

Goals

- Write code that uses real-time sensor input to evaluate conditional statements.
- Program a fan to control the temperature inside a greenhouse.

Materials and Equipment

- Data collection system
- //control.Node
- Greenhouse Sensor
- Greenhouse Sensor Module with cable and stopper
- Grow Light with Power Out cable and USB power adapter
- Power Output Module with Power Out cable
- USB Fan
- EcoChamber with included stoppers
- Wooden craft stick
- Shallow dish
- Zip seal sandwich bag
- Ice

Safety

Follow these important safety precautions in addition to your regular classroom procedures:

- Keep water away from sensor boxes, electrical plugs, and exposed electronic boards.
- Don't allow exposed electronic boards to contact a metallic or conductive surface.

CAUTION:

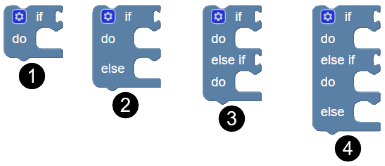
- *Don't look directly at the LEDs.*
- *Don't touch the LEDs.*

Research

Computer programs use the words *true* and *false* to check whether a condition is met.

The If Statement Block

The If statement block ❶ does actions if a condition has been met and is *true*. You can find the If statement block within the **Logic** block category. The gear ⚙ allows you to modify the block. Here is an example of how adding *else* ❷ can modify an if-then (or, if-do) statement: if it's cold, bundle up, *otherwise wear what you want* (if-do-*else*). An *else if* ❸ modifier does actions when the first If statement condition is not met: If it's cold, bundle up; *if it's hot, dress light* (if-do-*else if-do*). Statements are checked in order from top to bottom. You can combine several *else if* modifiers to define a series of conditions and set a different action for each true statement. The last block shown ❹ is an if-do-*else if-do*-else statement: If it's cold, bundle up; *if it's hot, dress light;* otherwise, wear what you want.



Add a Condition to an If Statement

Let's look at two ways to add conditions to an **If** statement.

1. **Comparator block** = : Use this block to compare two values. Choose from the following: =; ≠; <; ≤; >; ≥
2. **Bitwise block** and : Use this block with "and" to evaluate whether two statements are both true; change "and" to "or" to evaluate whether at least one of two statements is true.

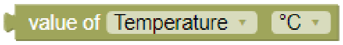
If you need to evaluate more than one condition before the next line of code, try nesting conditional statements.

Inputs and Outputs

The Greenhouse Sensor captures environmental information your code can use to make decisions. Greenhouse Sensor measurements are program *inputs* because sensor data can be fed into your code to evaluate whether a conditional statement is true. The **Hardware** category provides access to all connected PASCO inputs such as Greenhouse Sensor measurements and outputs such as the Grow Light and USB Fan. The Grow Light and USB Fan are *outputs* because you define what their action will be once a statement is found to be either true or false. The **Code Output** category includes blocks to create text or number outputs to be shown in a data display.

The Sensor Measurement Value Block

Greenhouse Sensor measurements are available in the Sensor Measurement block. The Temperature measurement is shown by default when the sensor is plugged in to a //control.Node connected to your device.



value of Temperature °C

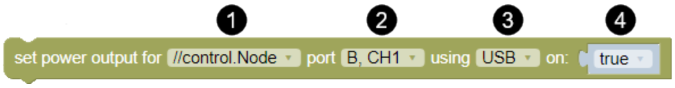
Use the drop-down menu to choose a different sensor measurement or change units such as °C to °F. Add the measurement block directly to your code if the measurement is called only once in the program. The more sensor calls you make, the more slowly your program will run. To avoid slowdown, define a variable with the sensor measurement once and then call the variable in your code instead of direct sensor measurements when possible.

Brightness: an Input and an Output

Brightness appears both as an input and as an output. A light sensor in the Greenhouse Sensor Module reports brightness as an input with an intensity range of 0 % to 100 %. The Grow Light uses brightness as a power output setting with a range of 0 (off) to 10 (maximum brightness).

The Power Output Block

The Power Output block controls the USB Fan. This block is available in the **Hardware** category when the Power Output Module is plugged in to a connected //control.Node. Four settings are available:



set power output for //control.Node port B, CH1 using USB on: true

1. **//control.Node Selector** - If you have more than one //control.Node connected to your device, you can choose which //control.Node a line of code will apply to. You will not need to adjust this setting in this investigation.
2. **Power Out Port and Channel Selector** - Use this drop-down to adjust the //control.Node Power Out port (A or B) and Power Output Module channel (CH1 or CH2). Take a moment to find Power Out ports A and B on the //control.Node and find channels 1 and 2 on the Power Output Module.
3. **Power Selector** - Each Power Output Module channel has a choice of 3 different plugs that can receive power: one USB plug and two types of terminal plugs (pin and block). You will be using only the USB plug for this investigation, however, be aware that all plugs on the same channel power up together.
4. **True/False Value** - For USB-powered outputs like the fan, when the value for on is *true* the USB plug within the selected channel will receive power. When the value for on is *false*, no power will be delivered to the USB plug. Terminal-powered outputs can be adjusted from 0% power (off) up to 100% (maximum) power.

Prototype

Part 1: Setup

1. Plug the Greenhouse Sensor into the Sensor port ❶ on the //control.Node as shown in Figure 1. Connect the Grow Light to port A ❷ and connect the Power Output Module to port B ❸.
2. Remove the two-hole stopper from the Sensor Module. Set the nut, washer, and module aside.
3. Use the craft stick to widen the cut in the two-hole stopper ❶ as shown in Figure 2. Push part of the Sensor Module cable into the edge of the widened hole as shown.
4. Remove the stick and use it to push the cable through the rest of the cut ❷ so the cable moves freely through the stopper hole.
5. Find the arrow symbol on the plug at the narrow end of the stopper. Connect the plug to the module with the arrow oriented as shown ❸.

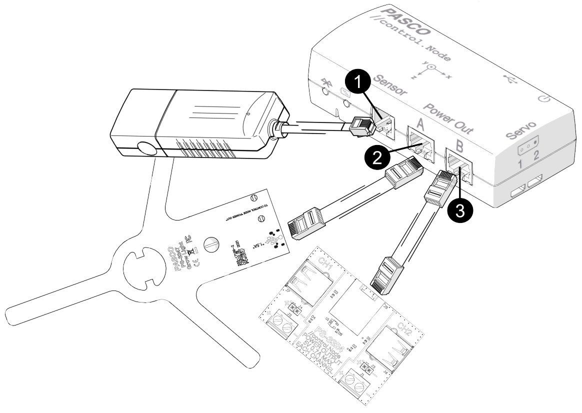


Figure 1. //control.Node Setup

6. Connect the other plug to the LIGHT HUMIDITY TEMP port on the Greenhouse Sensor ❹. Line up the single arrow on the plug to the single arrow on the back of the Greenhouse Sensor.

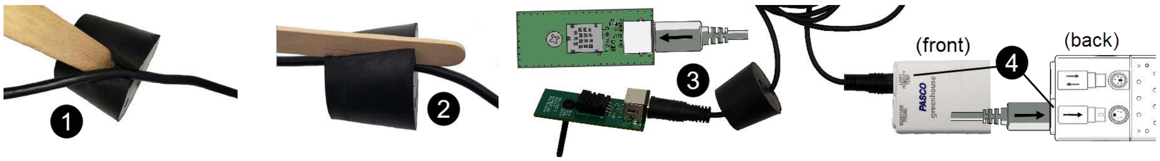


Figure 2. Insert the cable (1), push it through the cut (2); connect sensor at wide end of stopper (3); align arrows on back (4)

7. Refer to Figure 3 as you complete the following:
   a. Feed the Sensor Module and a few inches of cable through a hole on top of the EcoChamber lid. Seal the hole with the two-hole stopper.
   b. Push the Sensor Module screw through the bottom of the open stopper hole so the stopper contacts the module Ⓐ. Slide the washer over the screw emerging at the top of the stopper then hand-tighten the nut over the washer. The module should now be secured underneath the lid.
   c. Gently pull excess Sensor Module cable through the stopper.
8. Push a flat one-hole stopper through an open hole in the lid. Push the Grow Light screw through the stopper hole to set the light on the lid as shown. Rotate the Grow Light and Sensor Module stopper until the light sensor is centered with the Grow Light (see Ⓑ, arrow points out the tiny, flat, square light sensor).
9. Connect the USB Fan to Channel 1 on the Power Output Module.
10. Line up the two side holes between the chamber and lid to close the EcoChamber (greenhouse). Seal remaining holes with solid stoppers.

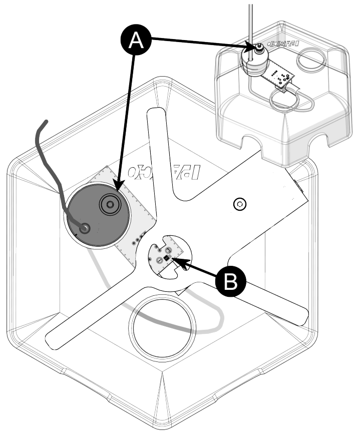


Figure 3. Light sensor centered on lid

11. Turn on the //control.Node and connect it to your device. Complete the following in the Sensor Data Configuration screen:
    a. Under Greenhouse Sensor, select **only** the *Temperature* and *Brightness* measurements ☑.
    b. Disable the //control.Node On-board Sensor measurements by changing the slider from on ● to off ○.
    c. Choose the **Digits and Graph** template.
12. Start collecting data. Record room temperature and brightness in the space below, then stop collecting data.

13. Insert the USB cable from the Grow Light into its power adapter. Plug the power adapter into a power outlet.

Part 2: Sensor-Based and Code-Generated Data Displays

1. Open the **Code Tool** by clicking the icon . Write code to turn on the Grow Light with both colors set to maximum intensity.
2. Toggle the **Code Tool** icon to return to the data display screen.
3. Select the **Time (s)** measurement below the x-axis. Find **Time** in the menu that opens. Click the **s** unit and choose **min** to change from seconds to minutes.
4. Start collecting data. Collect data until temperature stays the same for at least 30 seconds, then stop collecting data. While you wait, review step 7 and start working out your code. Once temperature is stable, record the final chamber temperature and brightness level in the space below.

5. Remove side stoppers. Carefully remove the lid without moving the Grow Light or Greenhouse Sensor Module position. Set the lid on the table so it can cool down.

> ***NOTE:*** *The light sensor and Grow Light must stay centered in the same position throughout the investigation (see Figure 3). If either component moves, carefully adjust positions until the brightness reading at full intensity equals the brightness recorded in step 4.*

6. Build a cooling system to use between runs as shown in Figure 4. Add ice cubes to a shallow dish until it is nearly full Ⓐ.
7. Set the dish inside the zip seal bag, remove excess air from the bag, and seal it.

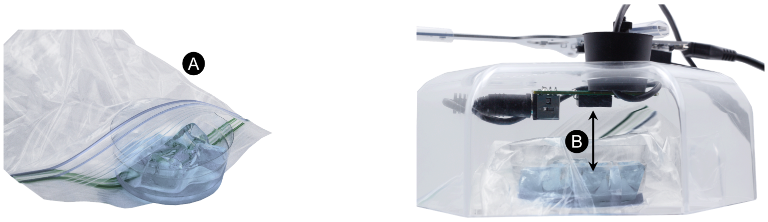


Figure 4. Cooling system placed beneath lid

8. Set the chamber lid on the table with the cooling system positioned directly beneath the stopper where the Sensor Module temperature sensor sits (Ⓑ, arrow). Do not allow the cooling system to contact any part of the Sensor Module.

9. Open the **Code Tool**. Write code to test how different Grow Light color and intensity combinations affect temperature inside the closed EcoChamber (see questions (a) and (b) below). Use the cooling system to bring the module to room temperature between tests; each test must begin at the same temperature. Replace the lid, run your code, and modify if necessary; then answer the following:

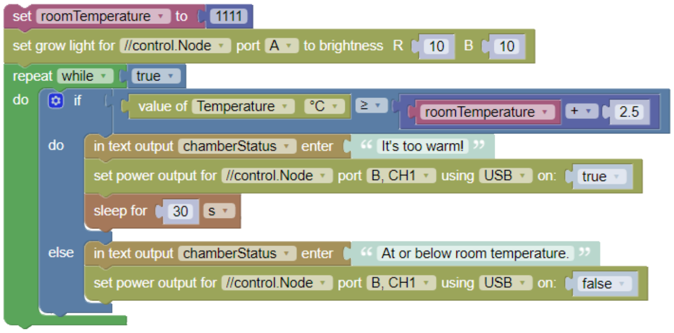
    a. Is the chamber temperature affected more by red light, blue light, or do both colors have the same impact? Explain your answer; assume both lights are set to the same intensity.

    b. Can light intensity and color be used to control the temperature inside the chamber? Why or why not?

    c. Check the boxes in the graph legend to show all runs. What do you notice about the first data point for brightness at time = 0 seconds for every run? What could explain this result? *Hint: Look at Part 1 data!*

10. Return the module to room temperature then place the lid on the chamber. Set the USB Fan in a side hole with blades outside the chamber. Seal the other side hole with a stopper.

11. Recreate the following code, but replace the value '1111' with the Part 1 room temperature reading. Change the units for Time to seconds, s. Feel free to modify time or temperature values to speed up testing.



```
set roomTemperature to 1111
set grow light for //control.Node port A to brightness R 10 B 10
repeat while true
do  if value of Temperature °C ≥ roomTemperature + 2.5
    do   in text output chamberStatus enter "It's too warm!"
         set power output for //control.Node port B, CH1 using USB on: true
         sleep for 30 s
    else in text output chamberStatus enter "At or below room temperature."
         set power output for //control.Node port B, CH1 using USB on: false
```

12. Click **Select Measurement** to set the top **Digits** display to Temperature. For the bottom display, click **Select Measurement** then switch the measurement list from the **Sensors** tab to the **User-entered** tab. Set the bottom display to the chamberStatus text output. On the **Line Graph**, set the y-axis to Temperature.

13. Start collecting data. Make observations of how the code works based on the measurements and text outputs in the data display. Allow data collection to continue until the fan turns itself on and temperature begins to decrease. Stop collecting data and answer the following questions:

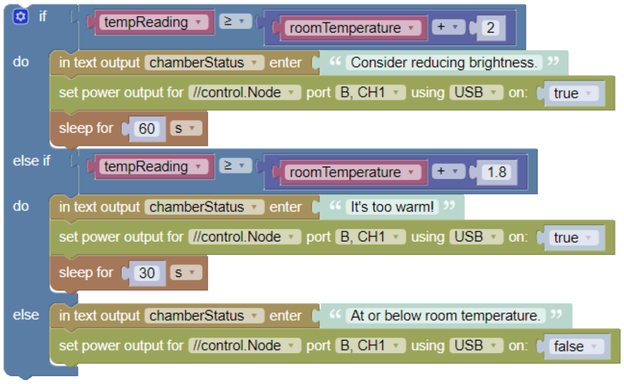
    a. Why aren't the first few code blocks included inside the loop?

    b. Translate the code inside the logic block to complete sentences as if you were giving directions to a person.

14. Create a numeric output in your code that displays corrected Grow Light brightness with the background brightness recorded in Part 1 subtracted out of the reading. Replace the Temperature **Digits** display with the numeric output. Modify code if necessary until the numeric output works as desired. Sketch new blocks added to your code below and summarize the modifications you made.

Part 3: Compound Conditional

1. Create a variable called tempReading and modify your logic block according to the following code. Test the code and monitor the data display until you have seen at least two different text output messages.



```
if      tempReading ≥ roomTemperature + 2
do        in text output chamberStatus enter "Consider reducing brightness."
          set power output for //control.Node port B, CH1 using USB on: true
          sleep for 60 s
else if tempReading ≥ roomTemperature + 1.8
do        in text output chamberStatus enter "It's too warm!"
          set power output for //control.Node port B, CH1 using USB on: true
          sleep for 30 s
else      in text output chamberStatus enter "At or below room temperature."
          set power output for //control.Node port B, CH1 using USB on: false
```

2. Explain why a variable is used instead of a direct sensor measurement, and describe how adding one or more *else if* statements changes how a program evaluates information.

Test

Write a program that adjusts light level to the highest possible brightness under all of the following conditions:

- the fan must keep the greenhouse (EcoChamber with lid) temperature no more than 2.5 °C above room temperature without running constantly;
- if the chamber reaches the maximum temperature, Grow Light intensity must gradually decrease until the fan can maintain temperature (red and blue intensity can have different values if necessary);
- incorporate one or more new text or numeric outputs that provides useful information to a greenhouse observer (you do not need to use any of the text or numeric outputs from Part 2);
- create a data display that shows the new text or numeric output to the observer and include at least one sensor measurement display.

Keep the lid components in place as shown in Figure 2 but feel free to experiment with airflow by adding or removing solid stoppers. On a separate paper, screenshot and print or sketch your code, explain how it works, and explain your data display choices. Save your work in SPARKvue for future reference.

Improve

The conditions outside your greenhouse will not always be the same as they are now. Modify your code to:

- Respond to variations in room temperature and natural light level throughout the day.
- Add a temporary light shutdown when the chamber exceeds the temperature limit.
- Allow the Grow Light to jump to higher intensity when the chamber temperature becomes significantly lower than room temperature.

Remember to unplug the Grow Light from the power adapter when you are finished.