# 2. CODE A COOLING BREEZE FOR A GREENHOUSE

## Project Overview

Students use Blockly code, a //control.Node, and Greenhouse Sensor measurements to program a fan that controls the air temperature inside an EcoChamber greenhouse illuminated by a Grow Light.

## Time Requirements

**Teacher Preparation:** 15 minutes

**Student Project:** 1 or more 45-minute class periods

## Goals

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

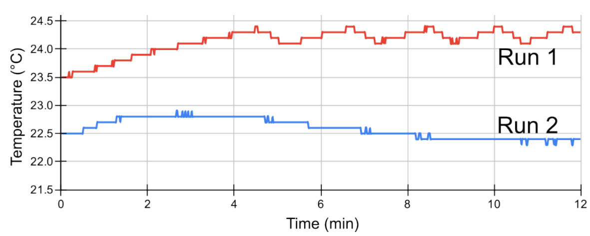
## Materials and Equipment

- 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

## Teacher Tips

- For best results, fully charge wireless devices before starting the investigation.
- For long-term investigations, it is recommended to connect the //control.Node to continuous USB power.
- For Blockly help, enter code-related search terms in PASCO's online Blockly reference guide at **help.pasco.com/sparkvue**. The guide is also accessible from the **Help** option in SPARKvue's main menu while data collection is stopped.
- For helpful Greenhouse videos, visit PASCO's Greenhouse Sense and Control Kit video library (**click here** or scan the QR code).
- The Grow Light circuit operates at a maximum of about 100 cycles or flashes per second. These flashes are not visible to either the eye or the light sensor. The 0 to 10 intensity scale determines what portion of the computing cycle to keep the LED on. A zero means the light is off. A number like 3 means the light is on 30% of each computing cycle. The light appears to change brightness to the eye or sensor which determines an average intensity.
- Some data collection or code testing may take 10 minutes of wait time or more (for example, in Part 2 step 4). Students can use this time to plan the next coding challenge in the activity.
- This investigation requires background knowledge of Blockly variables, loops, and Grow Light programming. To establish prior knowledge, consider having students complete the coding investigation titled *Program a Sunny Day for Plants* before performing this investigation.
- The *wooden craft stick* listed in the materials is also known as a Popsicle® stick.
- If Petri dishes are unavailable, plastic weigh boats, souffle cups, or condiment cups work well. To avoid contact with the Sensor Module, containers should be between 2" - 3" in diameter and no taller than 1.5".
- Consider completing steps 1 through 10 in Part 1 for students to maximize student coding time.

- Students enjoy using emoji in their text output. Review **emojiguide.org** to find emoji that can be copied and pasted into text output code on any device. Share the link with students if appropriate.
- Optional: If the fan is not able to achieve the desired temperature, try removing the inner filter. Disconnect the fan from the //control.Node. Use a size T10 Torx® screwdriver to remove the screws from the square side of the fan. Remove the outer shroud, leaving the fan in place. Find the side with the removable plastic grate. Gently pop out the grate and filter and store in a zip-seal bag. Securely screw the shroud back on the fan.
- Optional: Have students design an experiment to compare seedlings grown in the presence and absence of wind. Choose a plant that creates structurally demanding produce quickly, such as green beans. If using a fan, keep it near the seedlings; if using a natural breeze, make sure nothing blocks air flow around the seedlings.
- Optional: Explore how increased atmospheric carbon dioxide emissions impact temperature by having students use the stoppered hole to deliver excess carbon dioxide into the sealed EcoChamber. Pressurized carbon dioxide canisters for bicycle tire inflation can be found in most stores with a sporting goods department. You may see an initial temperature drop since an expanding gas loses heat, but over time the chamber will reach much higher temperatures when excess carbon dioxide is present since carbon dioxide is a greenhouse gas that absorbs re-radiated infrared energy. The polycarbonate chamber walls also trap infrared energy but can transfer some heat out by conduction through the plastic. You can also experiment with the effect of different colored papers set beneath the chamber on greenhouse temperature
- Optional: Help students improve science literacy by asking them to explain patterns in data and have them defend programming decisions based on data interpretation. Sample question and answer:
  *The graph below shows temperature data taken from two greenhouses set up identically to yours. One is temperature-controlled with a program similar to yours, and the other has no program running. Which run (1 or 2) shows data from the greenhouse with program-based temperature control? Explain your thinking.*




Both runs show an initial temperature increase and then a temperature drop, but Run 1 shows a repeating pattern: 1 minute of cooling immediately followed by warming.

The fixed cooling time suggests a fan turns on for 1 minute then shuts off until temperature approaches ~24.5 °C. Then the fan turns on again to stay below that temperature. Run 2 shows a temperature decrease that goes below the initial temperature so the room itself must be cooling.

## 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.*

## Prototype

### *Part 1: Setup*

12. Start collecting data. Record room temperature and brightness in the space below, then stop collecting data.

    Room temperature is 21.6 °C and brightness is 9.45 %.

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

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.

   At maximum red and blue intensity, temperature is 24.2 ° C and brightness is 47.38 %.

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.

      The blue light adds a lot more heat to the chamber per unit intensity compared to red light. For example the temperature goes up faster and reaches a higher level when only the blue light is on at intensity = 3 compared to only red light at the same intensity.

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

      Either light color at any intensity will increase the chamber temperature somewhat, so you could use the light to warm up the chamber if you wanted to but you could not use the light to make the chamber cooler. If you need light but don't want the temperature to increase as much, you could reduce the light intensity as a compromise.

   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!*

      The sensor starts collecting data before the program starts to run, so the Grow Light is off for the first data point at time = 0. The first data point measures the background light level before the light turns on. By the second data point, the Grow Light has turned on so the sensor reports its brightness combined with background light.

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?

       The "set" blocks are outside the loop because they are references for the first logic statement. These blocks are not inside the loop because room temperature and light intensity are both fixed, unchanging values so the code does not need to keep checking for new values.

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

       If the current chamber temperature is greater than or equal to 2.5 °C above room temperature, show "It's too warm!" in the chamberStatus text output and turn the fan on for 30 seconds. But if the chamber temperature is lower than room temperature, show "At or below room temperature." in the text output and leave the fan off. Repeat these instructions until the program is stopped.

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.

    The background brightness has to be captured early in the program outside of the loop with a variable set to equal the first brightness measurement - this initial measurement is the background brightness. Then a numeric output that takes the current Brightness measurement and subtracts the initial measurement can be added to the loop.

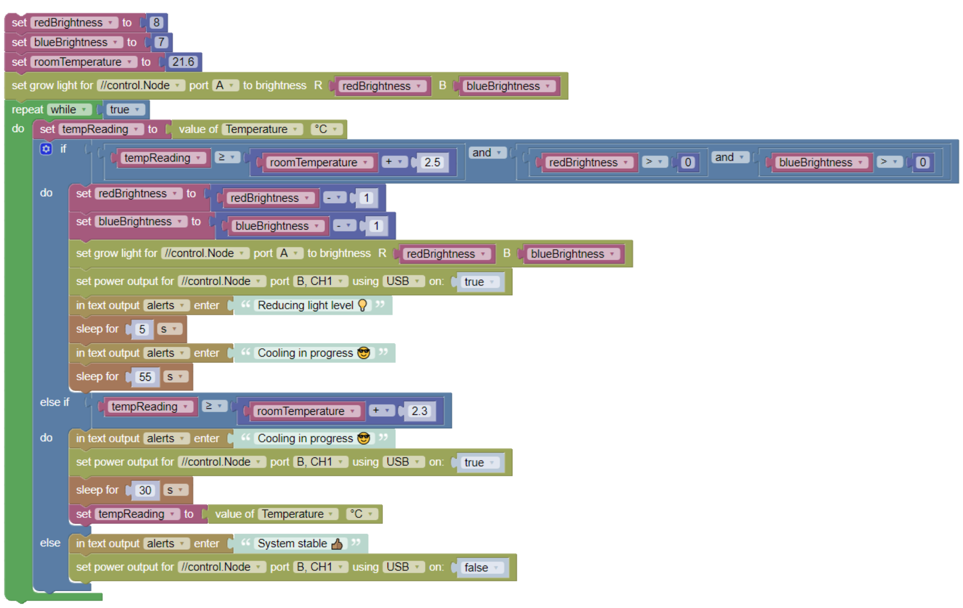

The initialBrightness variable must appear in any order outside and before the loop, and the numeric output must appear in any order inside the main level of the loop.

### *Part 3: Compound Conditional*

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.

    This time the program evaluates a single temperature value more than once, so using a variable to call that value will keep the program running as fast as possible. Before adding the else if statement, there was only one condition the code could evaluate. The else if statement lets you evaluate more than one condition. You can add any number of else if statements, each with a different output if desired, to respond to a number of different changing conditions. In this case, the program can now provide text output advice for changing the Grow Light intensity when the fan can't keep the temperature low enough to meet the desired conditions.

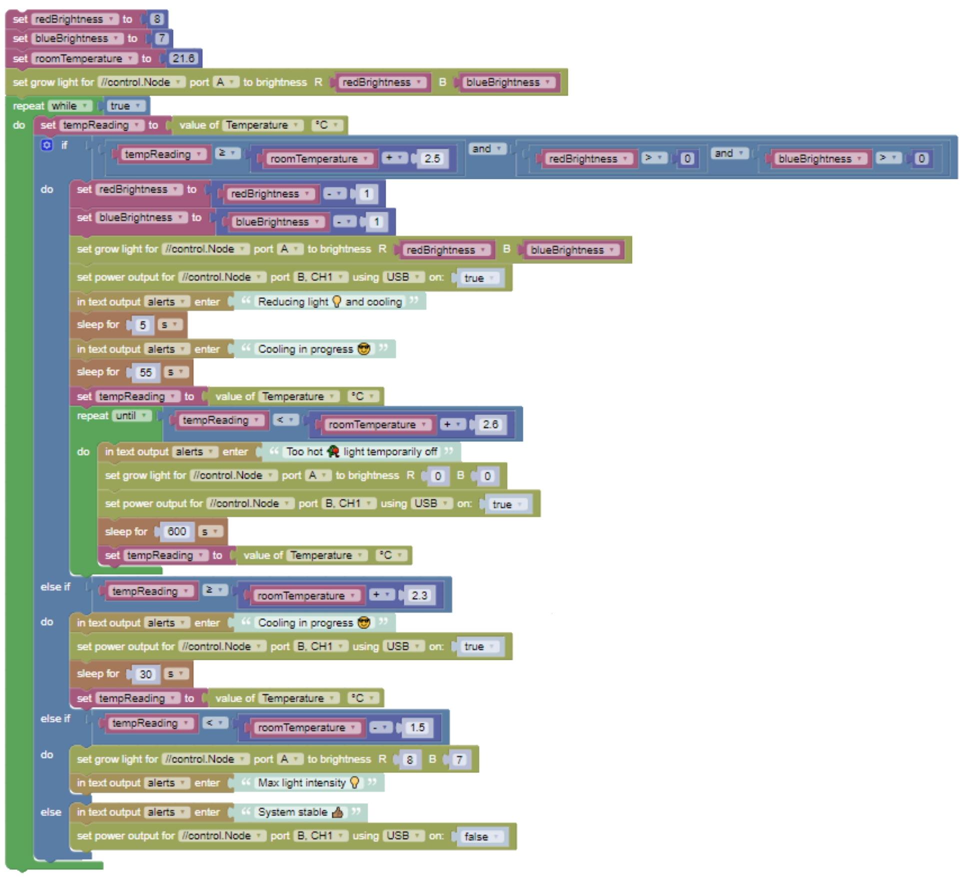
## Test



*The initial light intensity setting that could maintain no more than 2.5 °C above room temperature without constantly running the fan was red at level 8 and blue at level 7. Blue was set lower because it adds more heat to the chamber. Both stoppers were removed to maximize airflow and reduce temperature. The code has 2 thresholds to keep temperature in check. The first checkpoint runs the fan for at least 30 seconds when temperature reaches 2.3 °C above room temperature to give the chamber a chance to cool down before it goes 2.5 °C above room temperature. When the 2.5 °C threshold is reached, Grow Light intensity is reduced by 1 for each light color and the fan runs for 60 seconds. There is a new text output that alerts the observer when the fan turns on for cooling and when the light is being reduced. The data display includes a Digits and Graph display of temperature and a Digits display of the text output.*

## Improve

*Sample Code for Temporary Light Shutdown and Increased Light Intensity Below Room Temperature:*



## Technical Support

Need more help? Our knowledgeable and friendly Technical Support staff is ready to provide assistance with this or any other PASCO product.

| | |
|---|---|
| Phone (USA) | 1-800-772-8700 (Option 4) |
| Phone (International) | +1 916 462 8384 |
| Online | **pasco.com/support** |