
1. PROGRAM A SUNNY DAY FOR PLANTS

Project Overview

In this investigation, students use a //control.Node, a PASCO Grow Light, an EcoChamber lid and Blockly code to write a program that controls light output in order to model the sun's daily light cycle.

Time Requirements

Teacher Preparation: 5 minutes

Student Project: 45 minutes

Goals

- Write a program that uses variables and loops to control a light output that models the Sun's daily cycle.
- Explore how different kinds of loops work.

Materials and Equipment

- Data collection system
- //control.Node
- Grow Light with included cables and USB power adapter
- EcoChamber lid
- Flat #6 one-hole stopper

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).
- For ease of access, consider setting up multiple-outlet power strips within easy reach for students. Plug the USB power supply into the power strip and connect the USB cable ahead of time so students can easily connect the Grow Light to the USB cable.
- Depending on your students' experience level, you may wish to plug the USB cable and flat Power Out cable into the Greenhouse Light for students ahead of time.



- Optional: integrate a lesson on the relationship between light color and energy. Advanced biology students can make the connection to how the specific energies provided by the Grow Light colors are used by chloroplasts to power photosynthesis.
 - Have students help you collect any size threaded bottle caps from plastic drink bottles starting 1 week before you plan to perform this investigation. Rinse the caps thoroughly. You will need 1 cap per student group.
 - Students will pour a small amount of highlighter solution (preparation instructions below) into a drink bottle cap and set it under the EcoChamber lid while performing Part 2. Classroom lights can either be on or off for this activity. The light color with the highest energy will cause the solution to glow the brightest. When students compare red-only light to blue-only light, they will notice much more glowing with the blue-only light because *most* of the light released from the blue LED has the right amount of energy needed to excite the electrons in the fluorescent molecules to a higher energy state but only *some* of the light released from the red LED has the right amount of energy to excite the electrons.
 - Prepare the yellow highlighter solution the day before the investigation or sooner. To make enough solution for 10 student groups, complete the following:
 - Wear gloves to avoid staining skin.
 - Pour 75 mL of distilled water in a small cup or beaker.
 - Use pliers to pull the tip out of a new yellow water-based highlighter and add it to the water.
 - Use pliers to remove the closure at the opposite end of the highlighter tube.
 - Retrieve the fibrous inner ink reservoir from the open tube. Make one cut down the center of the reservoir then cut it into smaller pieces to fit in the cup so all pieces are completely covered by water.
 - Let the cup sit for at least 45 minutes, then pour the liquid into a separate sealed container. Avoid allowing solids into the new container. With gloved hands, squeeze excess water from the reservoir
 - The solution is re-useable and will last for years if sealed and stored at room temperature.

Safety

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

CAUTION:

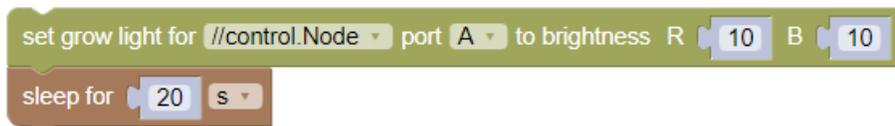
- Don't look directly at the LEDs.
- Don't touch the LEDs.

Prototype

Part 2: Program Light Intensity and Color

1. Set the brightness to the maximum value for both red and blue lights. Add a time delay to keep the light on for anywhere between 10 and 30 seconds and test your code. *Hint: Change the time units from milliseconds (ms) to seconds (s).* How long did you set the time delay for? What happened when you started running the code? How do you know if your time delay worked?

The light turned on when we started the code. We set the time delay for 20 seconds. We know the delay worked because after 20 seconds, the light turned off by itself and it stayed off. Sample code:



```
set grow light for //control.Node port A to brightness R 10 B 10
sleep for 20 s
```

2. Revise your code to turn on only red light for a few seconds, then only blue light for a few seconds. How can you turn on one color light while leaving the other color off?

Set the color you want to be on to a value above zero up to 10. Set the color you want to be off to zero. Sample code, blue-only:

```

set grow light for //control.Node port A to brightness R 10 B 0
sleep for 5 s
set grow light for //control.Node port A to brightness R 0 B 10
sleep for 5 s

```

Part 3: Set Up a Repeating Light Cycle

1. Write a program to turn on the red light at half-intensity for 7 seconds, have it turn off for 3 seconds, and use a loop to repeat the sequence until you press **Stop**. Set up the loop as shown below. The code to be repeated goes inside the loop. Share your complete, working code in the space provided.

```

repeat while true
do

```

Sample code:

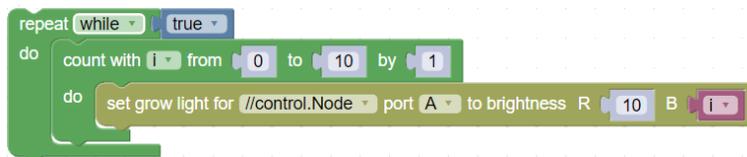
```

repeat while true
do
  set grow light for //control.Node port A to brightness R 5 B 0
  sleep for 7 s
  set grow light for //control.Node port A to brightness R 0 B 0
  sleep for 3 s

```

1. PROGRAM A SUNNY DAY FOR PLANTS | TEACHER'S GUIDE

2. Drag a counting loop block to the code area, then open **Variables**. Blockly automatically created a variable called *i* because the count block you added includes a variable called *i*. You can rename the variable if you like. Duplicate the code provided below.



Observe how it works as given then explore different outcomes by changing code values, variables, and presence or absence of the outer loop. Respond to the following:

- a. Why is the outer *repeat while* loop used in combination with a *counting* loop?

By itself, the counting loop will run only once so the repeat while loop keeps the series of color changes going until you stop the program.

- b. How does the *i* variable work with the counting loop?

The counting loop helps you define the range of the *i* variable step as well as the step size or degree of change between each value in the range. The code shown has the blue light set to 0 or "off" then increasing in its intensity by increments of 1 from 0 to 10 until it gets to 10. You can reverse things if you move the variable *i* from the blue light to the red light and set blue to 10.

- c. Switch the repeat while loop with a repeat (10) times loop and change the value to a number lower than 10. How does this loop differ from the repeat while loop?

The repeat (10) times loop lets you define exactly how many times code will repeat instead of repeating forever until stopped manually.

- d. Open the **Math** block category to find the addition block. Replace the number in the repeat (10) times loop with a simple calculation that results in a single digit answer and observe the result. Use the drop-down menu in the math block to experiment with different math functions besides "+". Describe the most interesting thing you learned about using this particular math block.

It was interesting to see that a negative math answer worked the same as the positive number. This might not work for all situations but it's what we saw with this code.

3. Why is it important to vary light energy for plant growth instead of leaving the Grow Light on 24 hours a day?

Living things have evolved to follow cycles such as sleep cycles to respond to Earth's solar cycle. Just like people need sleep, plants need a dark cycle to complete maintenance tasks that keep them healthy.

Test

Put everything you learned together to simulate the Sun's behavior in a continuous 24-hour loop. Assume the 24-hour daily cycle begins at midnight each day and consider intensity and color changes that happen throughout the day. For simplicity you can also assume Earth makes a perfect 180° path from one horizon to the other between sunrise and sunset where you live. On a separate paper, screenshot and print or sketch a copy of your program and explain your coding approach. Save your work in SPARKvue for future reference. *Hint: Set up time blocks with seconds (s), but when you are ready to test, change time units to milliseconds (ms) so you can speed through 24 hours worth of code in about 1.5 minutes (86,400 ms or 86.4 seconds) for testing. When you are confident with your final code, return time units to seconds.*

This code is designed to be started at midnight so for 6 hours (21,600 seconds) it's dark, both lights are off. During sunrise and sunset light is more of a warm color due to blue light scattering so we chose to show only faint red light and no blue light at those times of day. We then used a counting loop with a calculation to increase both red and blue light brightness at a regular rate (2 units of brightness every hour or 3600 seconds) from 7AM to 11AM. Then we allowed maximum brightness to run for 7200 seconds between 11AM and 1PM. The second counting loop uses a calculation to decrease brightness at a regular rate (by 2 units every 3600 seconds) from 1PM to sunset. Both lights are completely off before sunrise and after sunset. We set sunrise for 6AM and sunset for 6PM lasting 1 hour each.

```

repeat while true
do
  set grow light for //control.Node port A to brightness R 0 B 0
  sleep for 21600 s
  set grow light for //control.Node port A to brightness R 1 B 0
  sleep for 3600 s
  count with i from 2 to 8 by 2
  do
    set grow light for //control.Node port A to brightness R 1 + i B i
    sleep for 3600 s
  do
  set grow light for //control.Node port A to brightness R 10 B 10
  sleep for 7200 s
  count with i from 8 to 2 by 2
  do
    set grow light for //control.Node port A to brightness R i B i - 1
    sleep for 3600 s
  do
  set grow light for //control.Node port A to brightness R 1 B 0
  sleep for 3600 s
  set grow light for //control.Node port A to brightness R 0 B 0
  sleep for 21600 s
  
```

Improve

Sample Beginner Code for Continuous Fade:

```

set speed to 0.5
repeat while true
do
  count with i from 0 to 10 by speed
  do
    set grow light for //control.Node port A to brightness R 10 B i
  do
  count with i from 10 to 0 by speed
  do
    set grow light for //control.Node port A to brightness R i B 10
  do
  count with i from 0 to 10 by speed
  do
    set grow light for //control.Node port A to brightness R i B 10
  do
  count with i from 10 to 0 by speed
  do
    set grow light for //control.Node port A to brightness R 10 B i
  do
  
```

Sample Advanced Code for Continuous Fade (uses an If Statement block):

```
set startIntensity to 10
set endIntensity to 0
repeat while true
do
  count with i from startIntensity to endIntensity by 1
  do
    set grow light for //control.Node port A to brightness R i B 10 - i
    sleep for 60 ms
  do
    if startIntensity = 0
    do
      set startIntensity to 10
      set endIntensity to 0
    else
      set startIntensity to 0
      set endIntensity to 10
```

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](https://www.pasco.com/support)**