NAME PERIOD DATE

1. PROGRAM A SUNNY DAY FOR PLANTS

If you have solved math problems with *x* or have done science experiments, you have already worked with *variables*. A variable is any kind of information that can change its value. In math class *x* is a variable. If you are studying how quickly a plant grows in science class, plant height and time are variables. Computer programs use variables to store values the program can refer back to while running. You probably have experience with *loops*, too. Programmers use loops when they want something to happen more than once. For example if you enter the wrong password on your smartphone, the phone uses a loop to give you more chances to enter the correct password instead of locking you out of your phone. In this investigation, your goal is to use variables and loops to write code that makes a Grow Light act like the Sun.

You might be surprised to see that the Grow Light does not produce white light like a regular light bulb. The Grow Light only has red and blue LED lights. From a plant's point-of-view, the Grow Light is similar to sunlight. Even though all the colors of the rainbow are available in sunlight, plants use mostly red and blue light. Green chlorophyll molecules in plant leaves absorb energy for photosynthesis most efficiently from red and blue light.

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

Safety

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

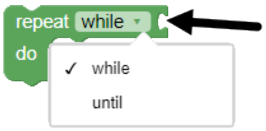
CAUTION:

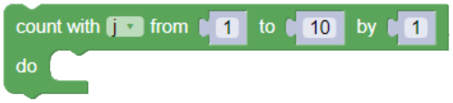
- *Don't look directly at the LEDs.*
- *Don't touch the LEDs.*

Research

Create and Explore Loop Blocks

Use loop blocks to control the number of times a program runs lines of code. The **Loops** block category has several blocks to choose from, including the following:





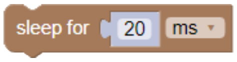
The loop block on the left repeats a task *while* a condition is valid or *until* a condition is met. You can set up the condition by adding blocks to the notch pointed out by the arrow. The loop block on the right is a counting loop. The example shown is set up to count up from 1 to 10 by 1 integer at a time. You might choose to have the block count down, use a different range, or count by a different integer.

Create and Explore Variable Blocks

Variable blocks are used to store a value that the code can refer to later. You can create variables in the **Variables** block category. You will need to choose a name for your variable. If you want to write programs like a professional, go online to learn about the different rules programmers use to name variables. Once you have created a variable, several new blocks with the variable name will appear in the **Variables** category.

Time Delays

Use the time blocks to pause code or get a timestamp from when the program was started. The **Time** category contains a sleep block to pause the code for a designated amount of time. When a program is sleeping it is not performing any other task. The sleep block below will cause a program to pause for 20 milliseconds (20 thousandths of a second or 0.02 second). You can change the time units from milliseconds (ms) to seconds (s) when needed.



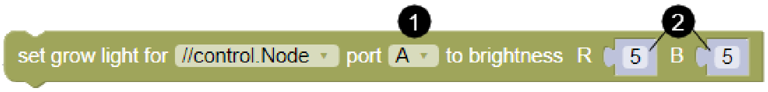
sleep for 20 ms ▾

Where is the on/off switch for the Grow Light?

The "on/off switch" on the Grow Light is actually code designed by you! Once the Grow Light is plugged in to the //control.Node and the //control.Node is connected to SPARKvue, the light stays off until you run a program that turns it on. Write code to vary the light intensity for each color from off to full brightness on a scale of 0 to 10.

The Grow Light Block

Use the Grow Light block to turn the Grow Light on or off and control the red-blue color mix. The block is found in the **Hardware** block category.



set grow light for //control.Node ▾ port A ▾ to brightness R 5 B 5

1. **Port** — Select the port where the Grow Light is connected, **A** or **B**.
2. **Brightness** — Set the brightness for **(R)**ed and **(B)**lue from 0 through 10. A zero setting turns the light off. The light turns on when values are set between 1 and 10 where 10 is the maximum brightness. Brightness can be set in whole number increments.

Prototype

Part 1: Setup

1. Set up the EcoChamber lid with the Grow Light positioned in the one-hole stopper ❶ as shown in Figure 1.
2. Plug the flat Power Out cable ❷ from the Grow Light into Power Out port A on the //control.Node.
3. Connect the //control.Node to your device, then select the **Digits** template.
4. Open the **Blockly Code Tool**. If the **Code Tool** is already open, delete any existing blocks.
5. Insert the USB cable ❸ from the Grow Light into its power adapter. Plug the power adapter into a power outlet.

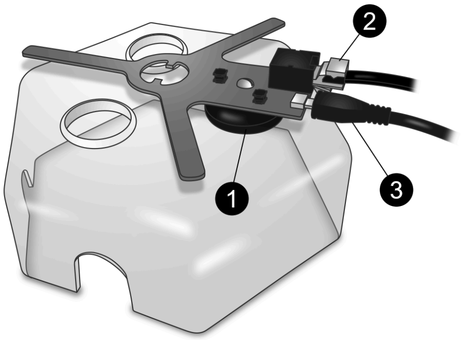


Figure 1. Grow Light setup on lid

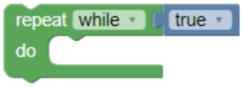
Part 2: Program Light Intensity and Color

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?

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?

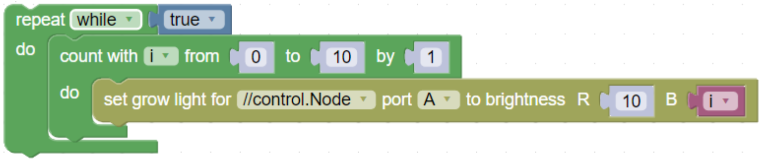
Part 3: Set Up a Repeating Light Cycle

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
```

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.



```
repeat while true
do  count with i from 0 to 10 by 1
    do  set grow light for //control.Node port A to brightness R 10 B i
```

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?

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

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?

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.

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

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.*

Improve

Modify your code to:

- Run a start-of-class to start-of-class 24-hour cycle instead of midnight-to-midnight.
- Add automatic corrections to your daily light loop to simulate seasonal changes in sunlight intensity and duration.
- Seamlessly fade light color from red to purple to blue over the course of one hour for smoother light transitions throughout the 24-hour cycle.
- Fade from red to purple to blue to purple back to red in a continuous cycle that lasts a few seconds instead of 24 hours.

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