

3. PROGRAM PERFECTLY TIMED RAIN

You already know that too little water can harm plants, but did you know too much water can also cause problems? Greenhouses tend to have limited air flow, so mildew may grow on leaves when there's too much water. Even worse, overwatering can lead to rotten roots or microbe overgrowth and the plant could die. Plant parents need to make sure their greenhouse plants get just enough "rain" to thrive. In this investigation, you will write code to automatically monitor soil moisture conditions, design a water pump system that uses soil moisture data to deliver exactly the right amount of water when the soil needs it, and alert you when it's time to refill the system's water supply. Each of these automated tasks can be addressed with a *function*, or a reusable task that is broken down into a series of steps.

Goals

- Use data to inform the design of an automated water delivery system.
- Write functions to organize tasks in a program.

Materials and Equipment

- Data collection system
- //control.Node
- Greenhouse Sensor
- Soil moisture probe
- USB Pump
- Power Output Module with cable
- Greenhouse Accessory Kit*
- EcoChamber with included stoppers
- Small cup or beaker
- Plant pot, ~4" diameter x ~4" height or smaller
- Potting soil
- Graduated cylinder, 10-mL
- Water reservoir
- Several strong rubber bands or zip-ties
- Binder clips
- Scissors

*The kit includes tubing, connectors, drip irrigation heads, hook-and-loop fasteners, and a #5 one-hole stopper.

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.

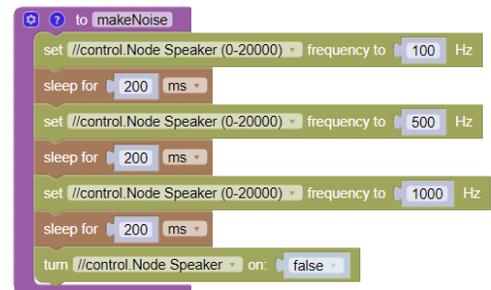
Research

Function Blocks

Connect the //control.Node to SPARKvue and select any data display from the Templates menu, then:

1. Open the **Blockly Code Tool** .
2. Go to the **Functions** category and create the *makeNoise* function code as shown.
3. Start code execution, and...

And... nothing happens! Function code by itself is not a program. Functions are meant to be inserted into programs. Open the Functions category to add the *makeNoise* function block. Start code execution to hear the //control.Node produce sound according to the directions in your code. Set the function block inside a *repeat while true* loop, change the function's frequency and sleep values, and test the results.

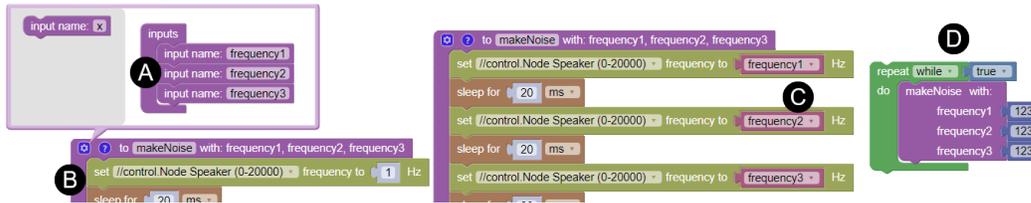


Functions with Inputs

Open the function settings by selecting the gear icon , then complete the following:

1. Drag 3 input name blocks into the input area and name each frequency1, frequency2, and frequency3 as shown below (A). The fourth frequency must remain at zero to turn off the speaker at the end of the function.
2. Right click or long-press an empty area on the left side of the function block as indicated by (B).
3. Choose **Create get frequency** for all 3 frequency inputs and place them in the appropriate //control.Node block (C). The makeNoise function inside the loop automatically updates to include the frequency inputs.
4. Add a number block to each input as shown (D).

Thanks to function inputs, you can modify frequency values for the makeNoise function directly in the loop - try it!



Comments on Functions

Select the question icon  to open the comment bubble. Write a short description of what the *makeNoise* function does. Comments are especially helpful when you share functions with others. Programmers use a *function library* to share functions they have created or find functions someone else created to use in their own code. SPARKvue has a function library for select //control.Node kits. Select the folder icon  in the upper-right corner. Change the category to **Greenhouse** and explore the available functions - be sure to read the comment for each function. Right-click or long press any block with a jagged edge, then choose **Expand Block** to see the code.

Prototype

Part 1: Setup

Greenhouse Sensor Setup

1. Complete the activities in the **Research** section, then turn off the //control.Node. Plug the Greenhouse Sensor into the Sensor port on the //control.Node  as shown in Figure 1.

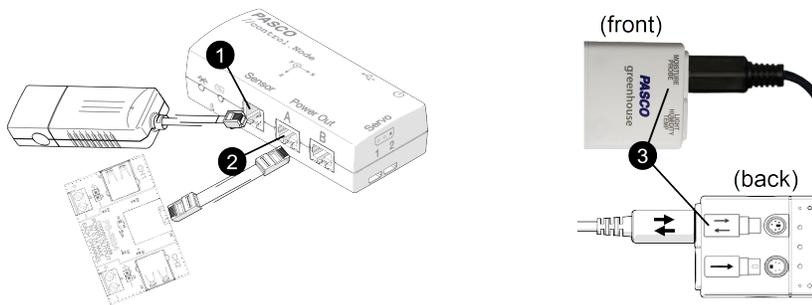


Figure 1. //control.Node and Greenhouse Sensor setup

2. Connect the Power Output Module to port A on the //control.Node .
3. Connect the soil moisture probe to the **Moisture Probe** port on the Greenhouse Sensor; make sure the double-arrows on the plug are aligned with the double arrows on the back of the sensor .

- Insert the probe cable into a flat #6 one-hole stopper so the probe comes through the bottom of the stopper. Guide the probe through any greenhouse lid hole then secure the stopper in the hole as shown in Figure 2.
- Line up the lid and chamber side holes to close the greenhouse.
- Adjust the probe cable length through the stopper hole until the probe sits upright inside the chamber, just touching the bottom as shown.

Software Setup

- Start a new experiment in SPARKvue. Connect the //control.Node to your device.
- Find **Greenhouse Sensor** in the Sensor Data Configuration screen. Select **only** the %VWC (Loam) measurement and clear the *Temperature* measurement.

NOTE: VWC stands for volumetric water content, a measure of soil moisture percentage. Loam is a type of soil; if you have soil that is known to be either clay or sand, choose the proper soil type instead of loam.

- Disable the //control.Node On-board Sensor measurements by changing the slider from on to off .
- Choose the **Digits and Graph** template. The data display screen will open.

Watering System Setup

- Connect the USB Pump to Channel 1 on the Power Output Module.
- Attach the wide, clear tubing to the USB pump nozzle.
- Fill the reservoir half-full with tap water.
- Use the suction cups to affix the USB pump to the bottom of the reservoir, then fill the reservoir with water as shown in Figure 3.
- Guide the USB cable and tubing outside the container as shown. If the container does not have a lid, create one with cardboard or aluminum foil.

NOTE: Always keep the pump completely below water and keep the container filled with plenty of water. Wrap clear containers with aluminum foil to discourage algae growth, but leave a portion unwrapped to monitor water level.

- Refer to Figure 4 as you plan your watering system. Will you need only one drip head, or should you add more? Move on to the next step *after* your plan is complete.
- Cut a section of blue tubing at least 2 inches or longer. Use the connector shown **1** to join the clear tubing from the pump to the cut tubing section.
- Connect the wide end of the stopper to the other end of the blue tubing section **2**.
- Add the remaining blue tubing to the narrow end of the stopper **3**.
- Cut tubing as needed and add drip heads **4**, however, leave blue tubing as long as possible.
- Fold open tubing ends (if any) and secure with either a rubber band or zip tie **5** for future modifications.



Figure 2. Probe setup



Figure 3. Reservoir

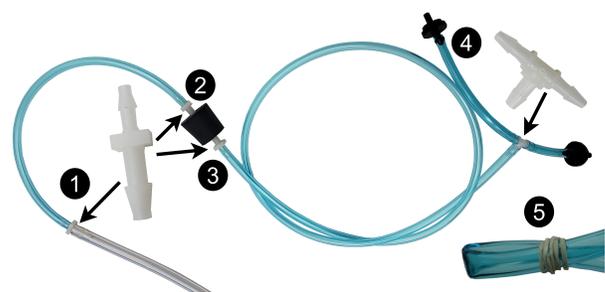


Figure 4. Tubing, connector, and drip head setup

3. PROGRAM PERFECTLY TIMED RAIN | STUDENT HANDOUT

12. The full length of your tubing should now be sealed except where there are drip heads. Leave one drip head open and use a rubber band to temporarily close additional drip heads (if any).

Part 2: Prepare for Water Delivery

Collect Water Volume Data

1. Set the open drip head inside the empty cup or beaker.
2. Delete or disable any existing blocks, then write code as shown to let the pump run until you stop the program.

```
repeat (while) true
do
  set power output for //control Node port A, CH1 using USB on: true
```

3. Allow the pump to run until all air bubbles have been removed from the tubing. Seal the drip head with your finger, and stop the program. Do not remove your finger from the drip head until directed.

NOTE: Do not allow air in the tubing from this point on. Keep drip heads at a higher level than the rest of the tubing to avoid air pockets. Run the pump continuously to remove trapped air.

4. Revise your code to run the pump for exactly 10 seconds as shown.

```
set power output for //control Node port A, CH1 using USB on: true
sleep for 10 s
```

5. Position the drip head to let it drip into the graduated cylinder. Remove your finger and run the program. Cover the drip head with your finger when the program stops.
6. Read the graduated cylinder volume at eye level to get the volume of water that flowed out of the single drip head over 10 seconds (in mL). Record the result for Run 1 in Table 1.
7. Pour the water from the graduated cylinder into the cup. Gently shake excess water from the cylinder.
8. Repeat steps 5-7 two more times; record results in Table 1.
9. Calculate the average volume of water delivered by the single drip head in 10 seconds; record the result in Table 1.
10. Multiply the average by the number of drip heads used in your system to get the average total volume of water delivered in 10 seconds. Record the result in Table 1.
11. Divide the average total volume of water by 10 seconds to get the water delivery rate in mL/s. Record the result in Table 1.
12. Pour the water in the cup back into the reservoir.

Table 1. Water Delivery Data

Run #	Water Volume (mL)
1	
2	
3	
Average	
Average Total	
Water Delivery Rate (mL/s)	

Manage Water Delivery with Soil Moisture Data Input

1. Fill the plant pot with soil and firmly compact it. Add more soil until the pot is filled with compacted soil up to 1 cm below the top.
2. Remove temporary rubber band closures from drip heads.

3. Use binder clips and rubber bands **1** to secure tubing and drip heads to the plant pot inside the chamber as shown in Figure 5. Point drip heads slightly upward with tubing set in a hook shape as shown **2** to avoid air bubbles and excess drainage.
 4. Run the pump until all air is removed throughout the tubing, then immediately stop the pump.
 5. Set the probe upright in the soil as deep as possible within 1 cm of a drip head as shown. Compact the soil against both sides of the probe to maximize soil contact.
 6. Close the chamber lid and adjust the probe cord length if necessary. Arrange the tubing stopper in a side hole.
 7. Delete or disable existing code. Toggle the code tool icon to return to the data display.
 8. Create a new page . Choose a 3-panel layout .
 9. Set the smaller panels to **Digits 1.23** displays and set the large panel to a **Line Graph** , display.
 10. Choose **Select Measurement** to set the top **Digits** display to **%VWC (Loam)** (or the same soil type used previously). Leave the bottom display unassigned for now.
 11. On the **Line Graph**, set the y-axis to the same measurement as the top **Digits** display.
 12. Start collecting data. When the soil moisture level (%VWC loam) stabilizes, stop collecting data and record the initial soil moisture value in the space below.
13. Return to the Code Tool. Convert the code below into a function set inside a repeating loop, but instead of creating a variable for *percentMoisture*, set it as an input in the function block. Set the *percentMoisture* loop-based input 1% higher than the initial moisture reading from above. Emojis are optional.



Figure 5. Drip setup

```

repeat until round value of %VWC (Loam) % >= percentMoisture
do
  set power output for //control.Node port A, CH1 using USB on: true
  in text output alerts enter "Watering 💧"
  sleep for 1 s
  set power output for //control.Node port A, CH1 using USB on: false
  sleep for 5 s
set power output for //control.Node port A, CH1 using USB on: false
in text output alerts enter "Soil moisture is OK ✅"

```

NOTE: The soil moisture probe can provide decimal readings, so the **round** block is used to make sure readings are whole numbers that directly compare to the data display.

14. Return to the data display. 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 *alerts* text output.
15. Test and revise the code as needed until your loop works as expected.

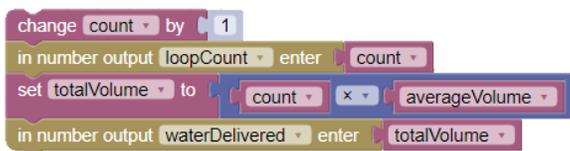
NOTE: The soil may become totally saturated and excess water may build up inside the chamber. Dispose of excess water from the chamber (do not return this water to the reservoir) if it begins to pool and refill the plant pot with fresh soil as needed.

16. Sketch your working loop in the space provided (no need to reproduce the contents of the function itself).

17. What is the purpose of each sleep delay in the function? What might happen if those sleep delays are removed?

Keep Track of Water Delivered

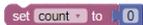
1. Add the following blocks to your function in-between the 1 second sleep delay and the power output block that turns the pump off; set *averageVolume* as a function input. Create the *count* and *totalVolume* blocks from the **Variables** menu.



```
change count by 1
in number output loopCount enter count
set totalVolume to count x averageVolume
in number output waterDelivered enter totalVolume
```

2. Use the water delivery rate from Table 1 as the value for *averageVolume* in the function block.

3. Add the following as the first block in your program, outside the repeating loop, to start *count* at zero:



```
set count to 0
```

4. Modify the **Digits** displays to monitor values for *loopCount* and *waterDelivered* numeric outputs as you test the code. Change values as needed to see how the program behaves when the soil *does* and *does not* need water, then answer the following:

a. What does *count* do?

b. What information does the *totalVolume* calculation provide?

Test

Create a program with a relevant data display that uses functions with inputs inside a repeating loop to achieve the following:

- Slowly add water to soil until a target soil moisture level is reached;
- Use the `//control.Node` speaker to set off an audible alarm and display a message in the `alerts` text output when the water reservoir is about 1/3 full;
- Adds a new alert (either visual or audible, or both) when soil moisture is much higher than the target level.
- Uses a variable in place of a direct sensor measurement when possible to keep the program running as fast as possible.

Save your work in SPARKvue for future reference. On a separate paper, screenshot and print or sketch your code including function contents, explain how it works, and explain your data display choices.

Improve

- Modify your program to disable the pump when the reservoir is only 1/4 full.
- Add a second soil moisture sensor to your water reservoir to use direct data input (instead of calculations) to monitor reservoir water level.
- Incorporate a Grow Light function to add another visual alert when the reservoir water level is low.