3. PROGRAM PERFECTLY TIMED RAIN

Project Overview

Students use Blockly code, a //control.Node, and Greenhouse Sensor measurements to program an automated plant watering system of their own design.

Time Requirements

Teacher Preparation: 30 minutes

Student Project: 1 or more 45 minute class periods

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.

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).
- This investigation requires background knowledge of Blockly variables, loops, accessing Greenhouse Sensor measurements, operating the Power Output Module, creating numeric and text outputs, and creating data displays. To establish prior knowledge, consider having students complete the coding investigations titled *Program a Sunny Day for Plants* and *Code a Cooling Breeze for a Greenhouse* before performing this investigation.
- Prepare soil for students ahead of time. Use a large zip seal bag to gradually add water to potting soil until moist, but below an ideal moisture level for potted plants. Start with 1 cup of water to 5 cups of dry soil; use less water for soil that is not completely dry. Students may need multiple soil samples; plan on enough soil to fill the plant pot up to 3 times per student group.

- Plant pots are not included in the Greenhouse Sense and Control Kit. You will need pots about 4" or less in diameter and height. If plant pots are unavailable, use similar-sized plastic cups with several drainage holes cut through the bottom.
- The water reservoir is not included in the Greenhouse Sense and Control Kit. To prevent accidental water overflow, the ideal water reservoir should have less water volume than empty space in lower half of the EcoChamber. Look for a container with a lid that can accommodate the tubing and USB cord as shown in Figure 2 on the student handout. Students will need to know the approximate volume of the water reservoir. Write the volume (in mL) on the reservoir in permanent marker.
- To maximize student coding time, complete Part 1 ***Greenhouse Sensor Setup*** steps 1-6 and Part 1 ***Watering System Setup*** steps 1-5 for students ahead of time.
- It takes a fair amount of grip strength to connect tubing; younger students may need assistance. Add a small amount of glycerin to connector wedges to ease insertion and removal from tubing.
- Tubing can be cut shorter if desired. A replacement tubing accessory kit (tubing, connectors, and drip irrigation heads) can be purchased from **pasco.com** (part #PS-3348) or you can purchase replacement tubing and drip irrigation heads separately.
    - Replacement polyurethane tubing can be purchased from your local hardware store or from a science supply company. Tubing sizes are as follows:
        - Clear tubing connected to USB pump: Outer diameter = ⅜"; inner diameter = ¼"
        - Blue tubing: Outer diameter = ¼"; inner diameter = $^{3}/_{16}$"
    - Replacement pressure-compensating drip irrigation heads to fit ¼" tubing can be purchased from your local hardware store. The flow rate for original drip heads is 2 GPH; use heads with different flow rates if desired.
- If you plan on completing further greenhouse investigations, keep the plant pot and watering system setup intact to save time and material costs.
- Optional: Have students design an investigation to see how different soil types respond to the same amount of added water in terms of soil moisture and water available to plants.

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.

Prototype

Part 2: Prepare for Water Delivery

Collect Water Volume Data *- see Table 1 (next page) for sample data. Each run shows the water volume delivered by one drip head over 10 seconds.*

Manage Water Delivery with Soil Moisture Data Input

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.

    %VWC (Loam) = 14%

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

    Sample loop with function:

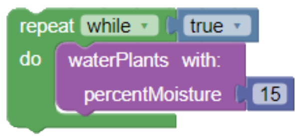


Table 1. Sample Water Delivery Data

Run #	Water Volume (mL)
1	5.2
2	5.2
3	5.3
Average	5.23
Average Total	10.46
Water Delivery Rate (mL/s)	1.05

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

    The sleep delays let the water slowly trickle down into the soil so the moisture probe has a chance to get an accurate reading before more water is added. If the sleep delays weren't there, the pot could quickly become overwatered, because water can't move fast enough through the soil for the sensor to immediately "see" added water.

Keep Track of Water Delivered

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?

    This variable keeps track of how many times the pump has turned on. When the program starts, count starts at zero, and increases by 1 each time the pump turns on.

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

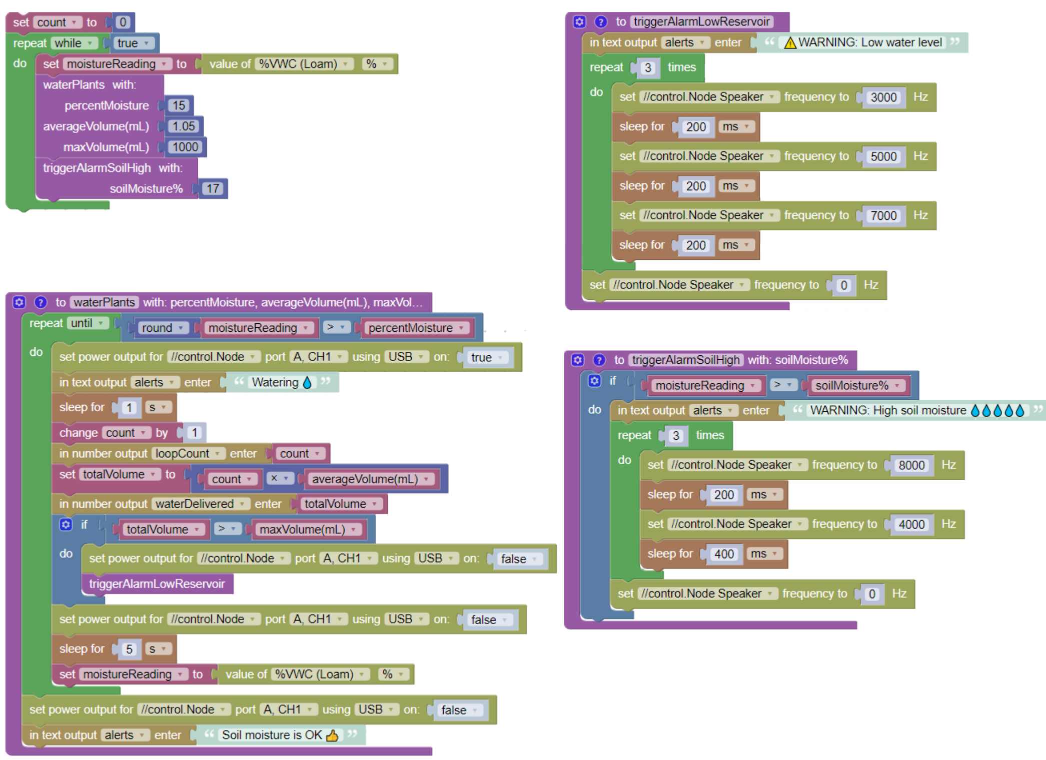
    This calculation gives the real-time, cumulative amount of water dispensed since code execution began. It takes the number of times the pump has turned on and multiplies it by the average amount of water delivered each time the pump turns on for 1 second.

Test

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

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.

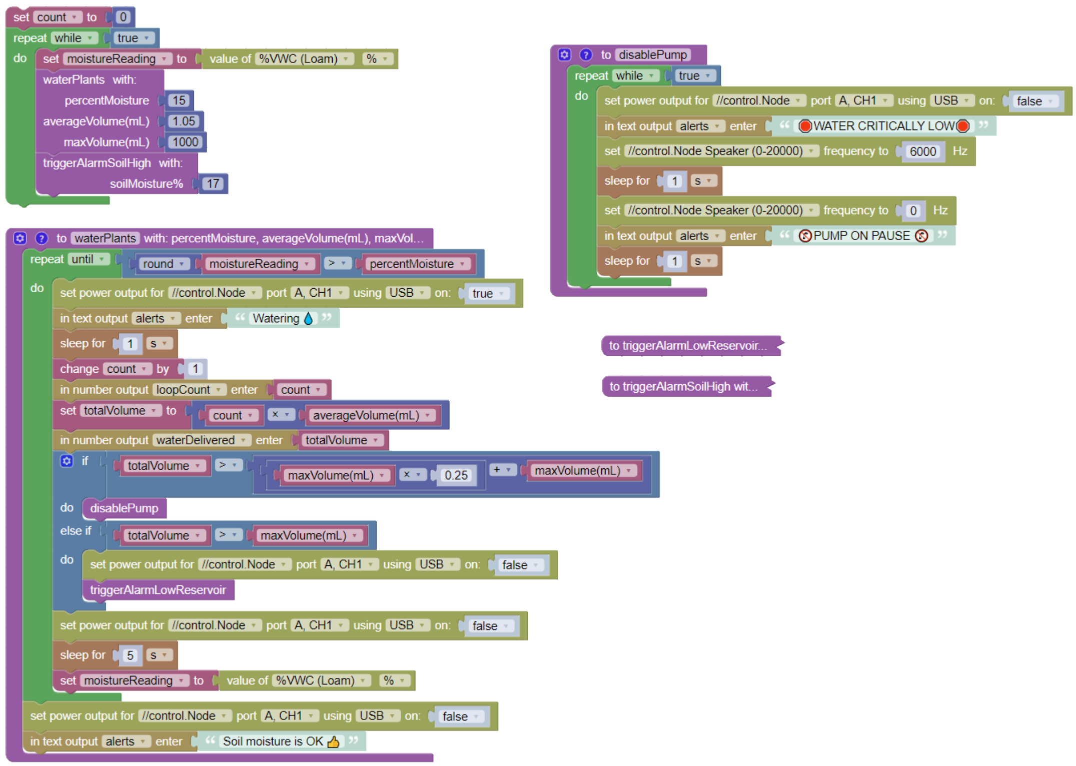
I ended up with 3 functions in my program, with one nested into the main waterPlants function. I modified waterPlants to set a new input that lets me decide the water volume that will trigger the triggerAlarmLowReservoir function (a speaker and text alert that goes off when the low water reservoir threshold "maxVolume(mL)" is reached). Since the reservoir has a volume of about 1.5 L, the alarm goes off once an estimated 1,000 mL of water has been delivered. The triggerAlarmSoilHigh function alerts the user when soil moisture is too high. The function has an input where the user decides what soil moisture level will trigger the alert. I noticed the program was running a little slow, then I found places to call a variable I created for soil moisture instead of calling the measurement directly from the sensor. That helped the program run better. For the display, I wanted to see the amount of water delivered and also the text alerts, plus a graph of the soil moisture. I wanted to get the most information on the screen as possible with out being overwhelming so I could quickly see the data and figure out what might have happened if the alarm is goes off.



Improve

- Modify your program to disable the pump when the reservoir is only ¼ 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.

Sample code for first bullet, for ~1.5 L reservoir; setting a calculation-defined trigger based on the initial low volume trigger instead of a direct value provides the most flexibility:



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**