SPARKvue®

Contents

Getting started

Find tutorials that help you get started with common tasks in SPARKvue.

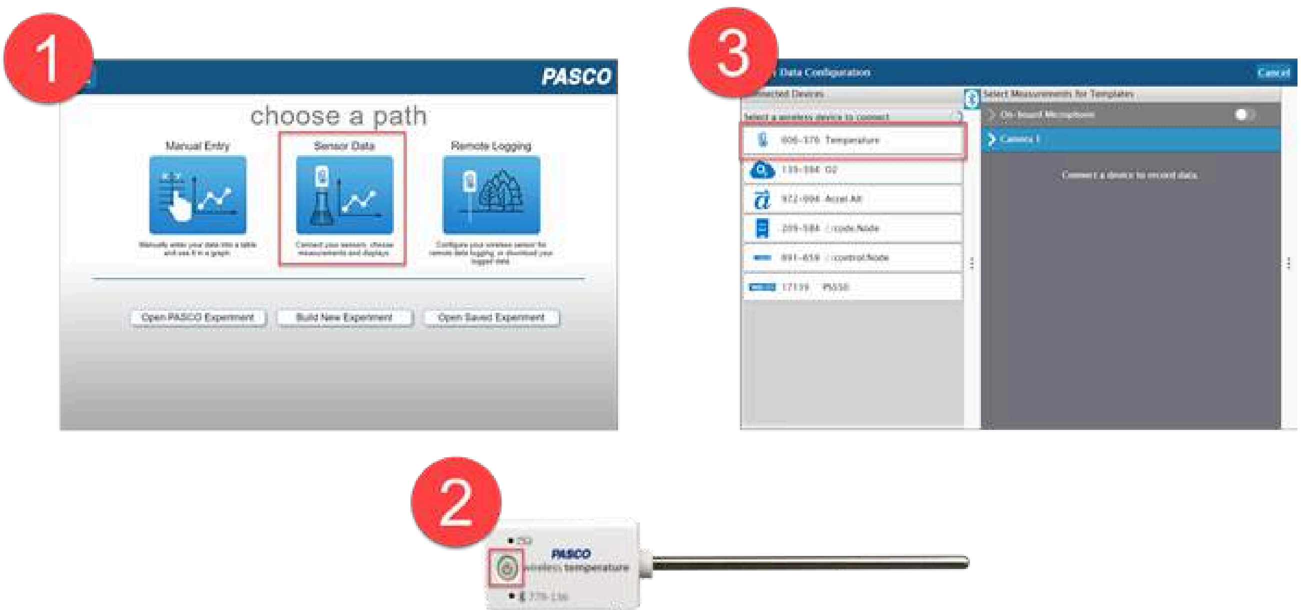
Collect, visualize, and analyze data

Learn how to set up a sensor, collect and analyze data, and save an experiment.

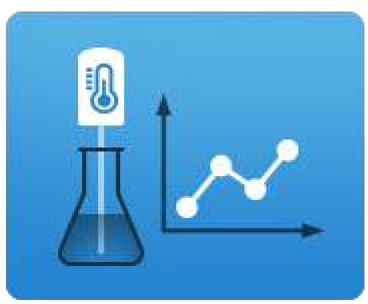
In this tutorial, we're going to use a Wireless Temperature Sensor to measure the temperature of a cup of water. If you don't have a Wireless Temperature Sensor, you can still follow the steps in this tutorial to collect data with a different Wireless sensor.

Step 1: Connect the sensor

Connect the sensor to SPARKvue and select a template for displaying the data.



1. On the Welcome Screen, click **Sensor Data**.

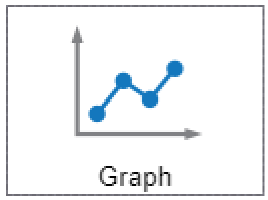


2. Turn on the sensor by pressing the power button until the lights turn on.

3. On the **Sensor Data Configuration** screen, click the sensor that matches the Bluetooth ID printed on the sensor.

> **Tip:**
> SPARKvue lists the sensors in order of proximity so your sensor is likely at the top.

4. Under **Templates**, click **Graph**.



> **Note:**
> For multi-measurement sensors, you can also select which measurement you want to display by selecting the box next to the sensor. In this case, we only have one measurement that is selected by default.

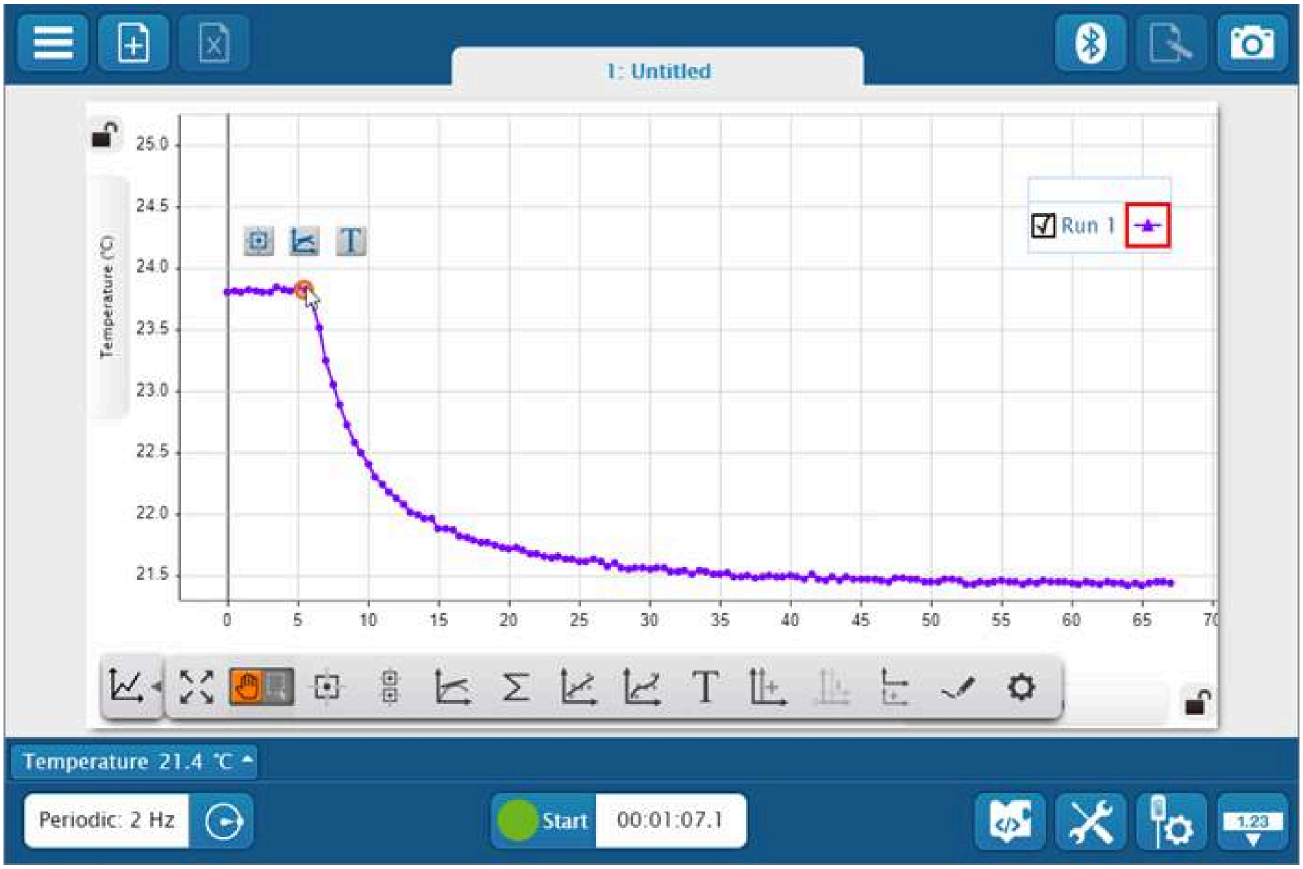
Step 2: Collect data

Record a run of data and view the results on the **Graph** display.

1. On the Experiment Screen, click **Start** to begin recording data.
2. Place the sensor probe in a cup of water.
3. Once the sensor reaches equilibrium, click **Stop** to end recording data.
4. Adjust the viewing area of the graph by dragging the axes. You can also click the scale-to-fit tool to automatically scale the graph.

Step 3: Add a note

Add a note to a data point to add context to the graph.

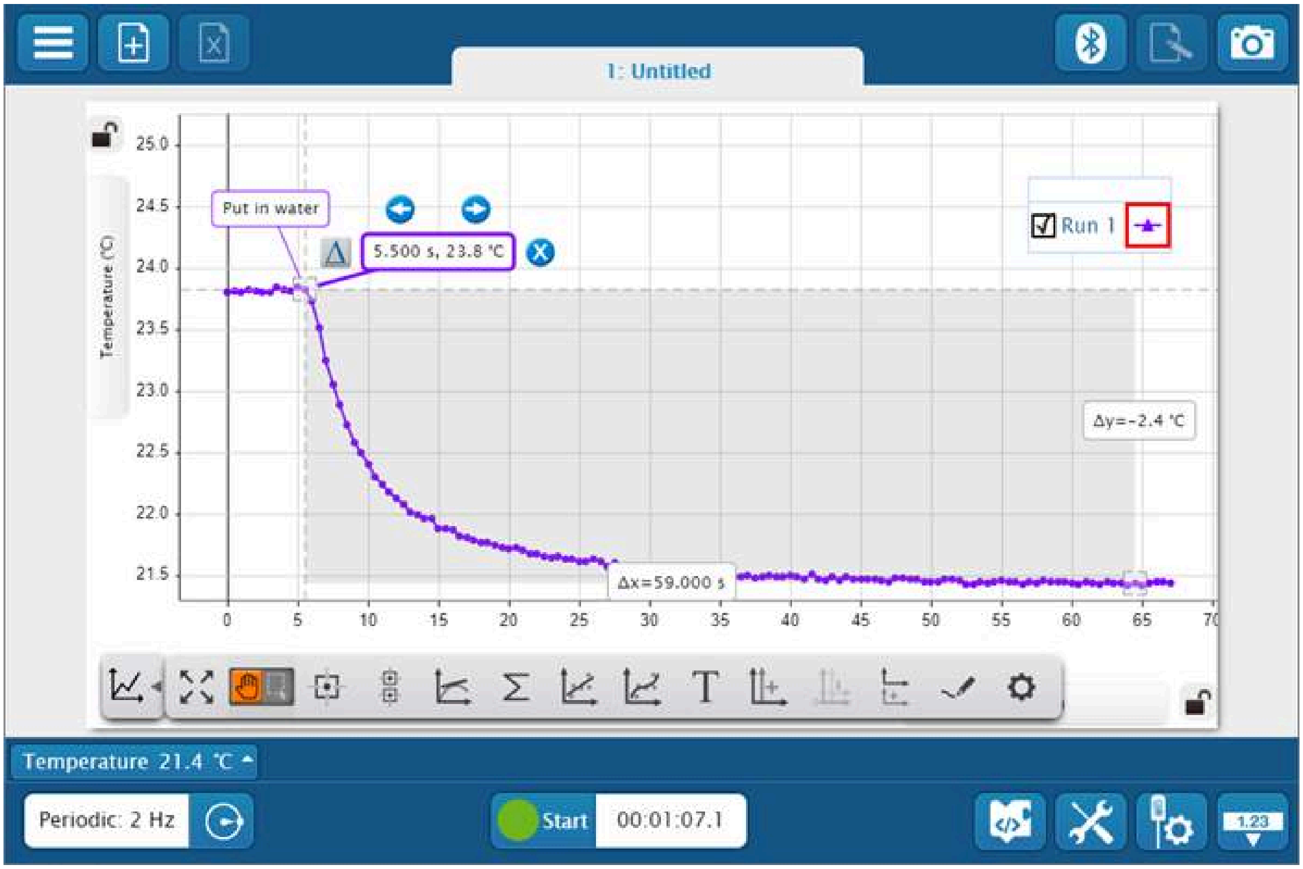


1. Click a data point to add a note. A selection of tools appears.
2. Click the annotation tool  to add an annotation to that point.
3. Select the **Enter Note** box then use your keyboard to type a note.
4. Click **OK** when finished. The note appears on the graph.

Tip:
Reposition the annotation by dragging the text box.

Step 4: Find data coordinates

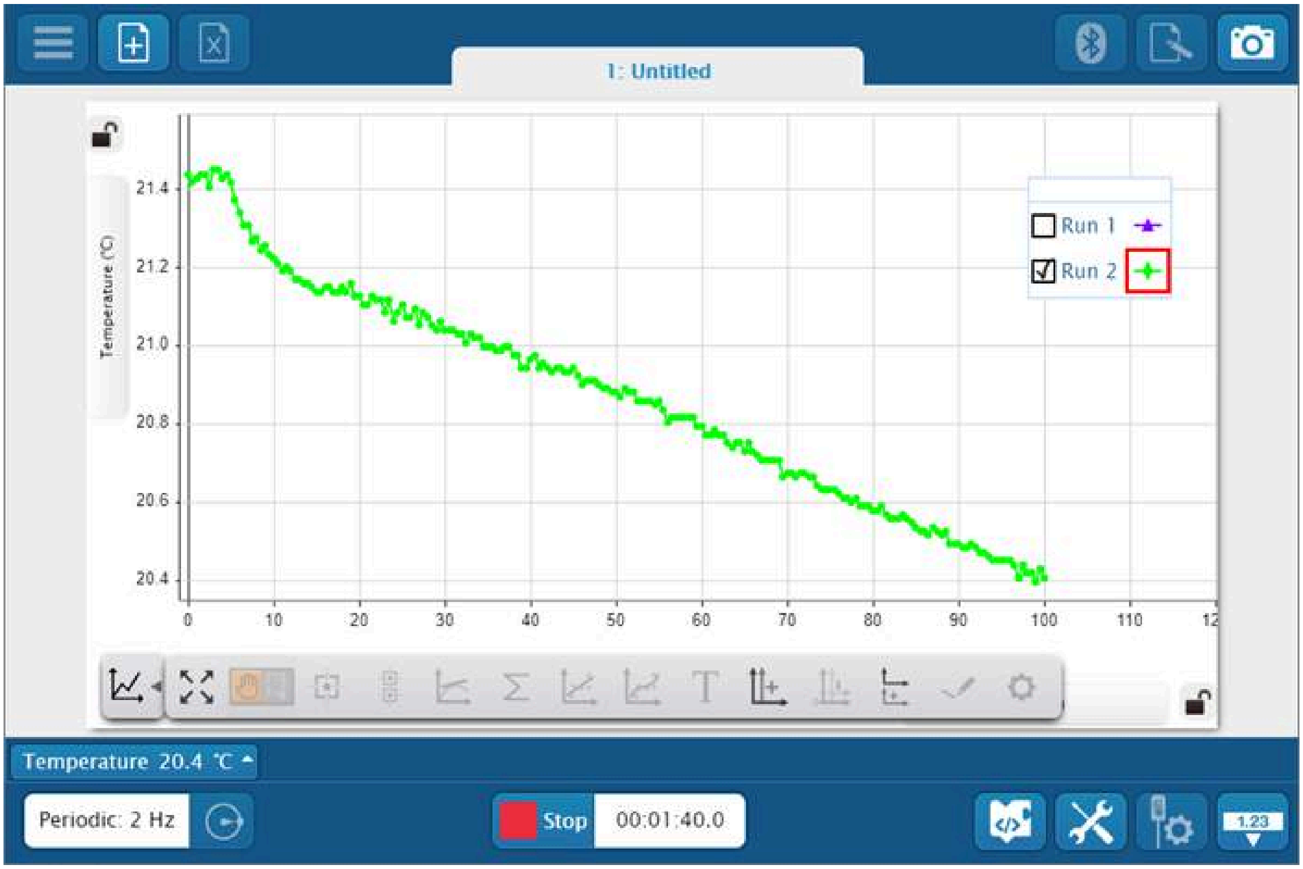
Determine the coordinates of a data point and the difference between two points on the graph.



1. Click a data point to analyze.
2. Click the coordinates tool . A box displaying the coordinates appears.
3. Click the coordinates box then click the delta tool . A second coordinates box appears.
4. Drag the second coordinates box to a new location to determine the difference between the two points. Boxes showing the difference in y and the difference x appears.

Step 5: Collect a second run

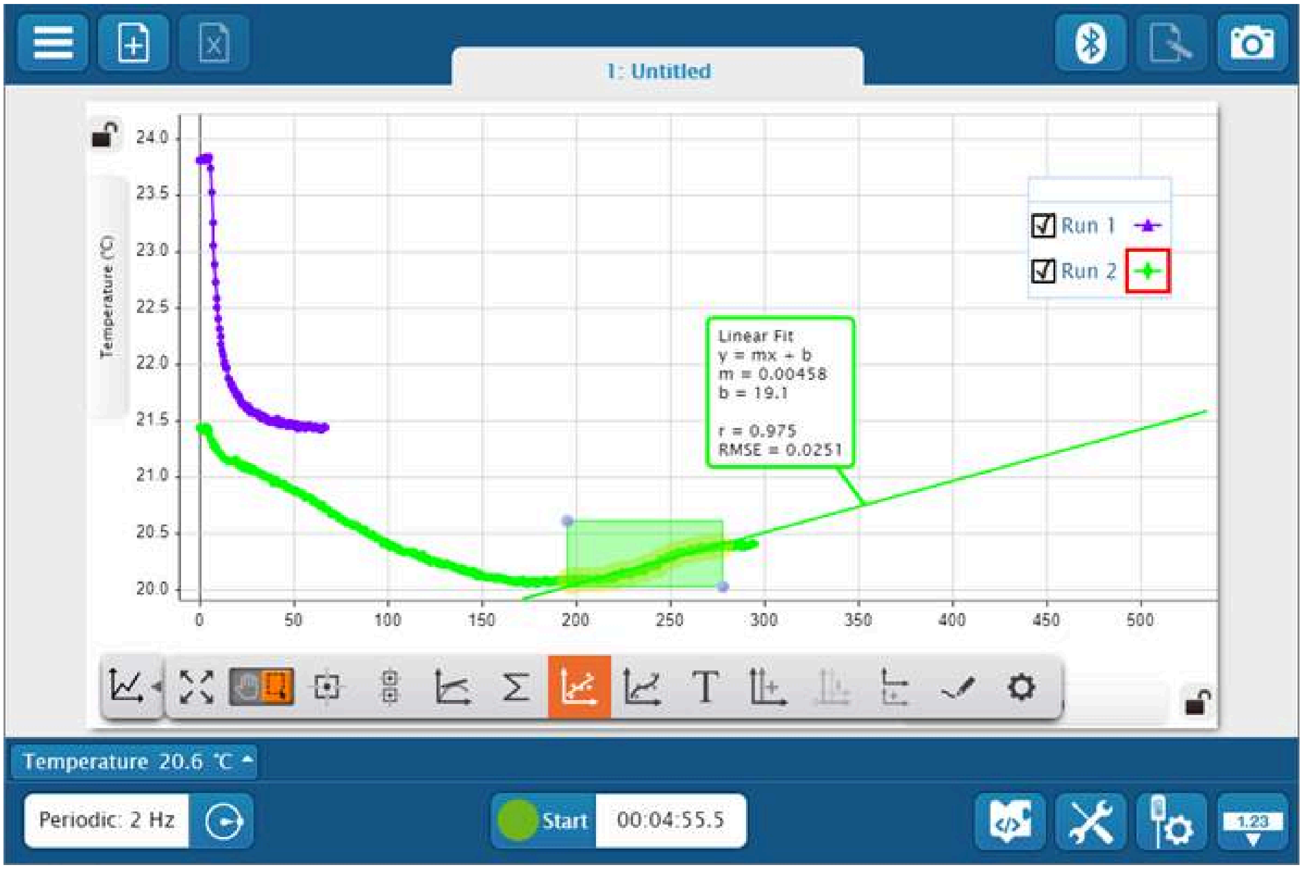
Record a second data run and compare it to the first run.



1. Click **Start** to begin recording a second data run. The first data run is hidden.
2. Remove the Temperature Sensor probe from the water and set it on the table.
3. After several minutes, click **Stop** to end recording data.
4. In the graph legend, select the **Run 1** box ☐ to make the first run visible.

Step 6: Fit the data to a line

Add a linear fit to a selection of the data to find the slope and y-intercept.

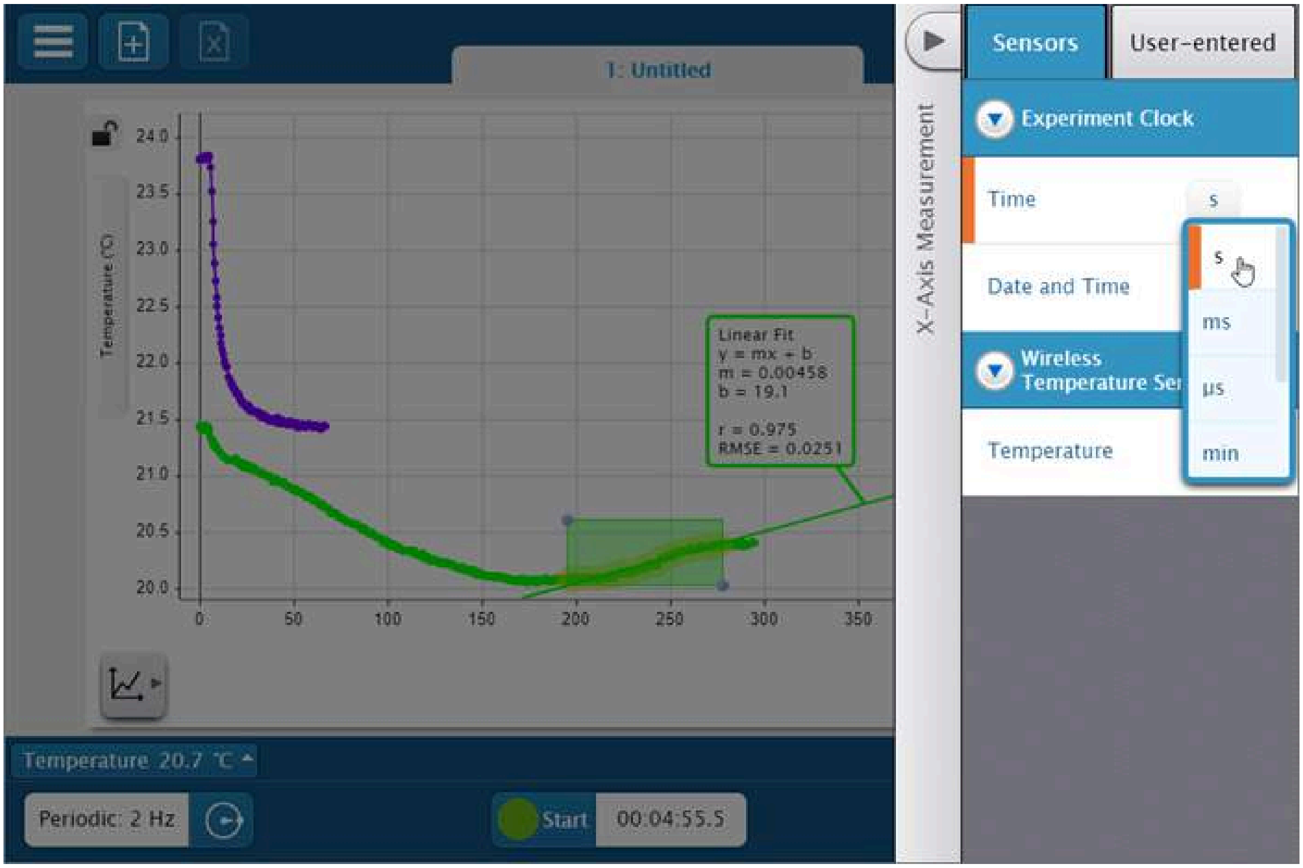


1. Toggle the cursor tool to selection mode .
2. Drag the cursor across the graph to select a portion of the graph to analyze.
3. Select the linear fit tool to fit a line to the selection. A line appears with a box displaying the slope, y-intercept, and error values.

> **Tip:**
> Drag the axes to see a projection of the linear fit.

Step 7: Change the units of the Time axis

Change the units of the Time measurement from seconds to minutes to make the data easier to read.



1. Click the toolbar drawer to collapse the toolbar.
2. Click **Time (s)** on the x-axis.
3. In the **X-Axis Measurement** panel, click **s** then select **min** from the list.

Step 8: Save the data

Save your experiment file to share with others.

1. Click the **Main Menu** then select **Save As**.
2. Create a name for your file then save it on your device.

> **Tip:**
>
> On mobile devices, directly email your file by tapping the share tool then tap **Share SPARKlab**.

To open your file at a later time, click **Open Saved Experiment** on the Welcome Screen.

Next steps

Now that you've gone over the basics of collecting data in SPARKvue, here are some suggestions to learn more about SPARKvue:

- Explore the graph display tools.
- Explore the Experiment Screen tools.

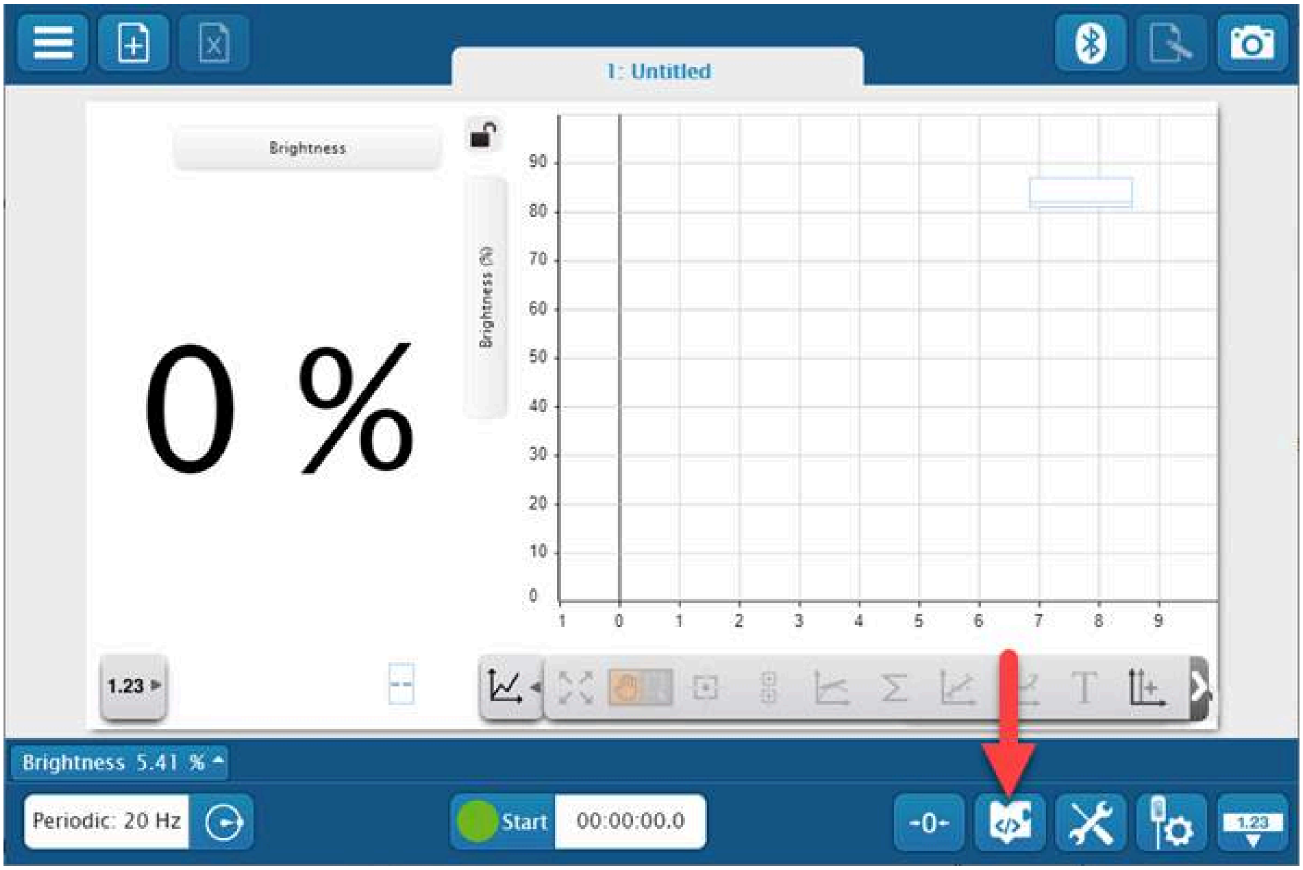
- Build a new experiment page.
- Use the search feature at the top of this page to find a specific topic.

Create a program using sensor data

Learn how to code a simple program and use sensor data to control outputs.

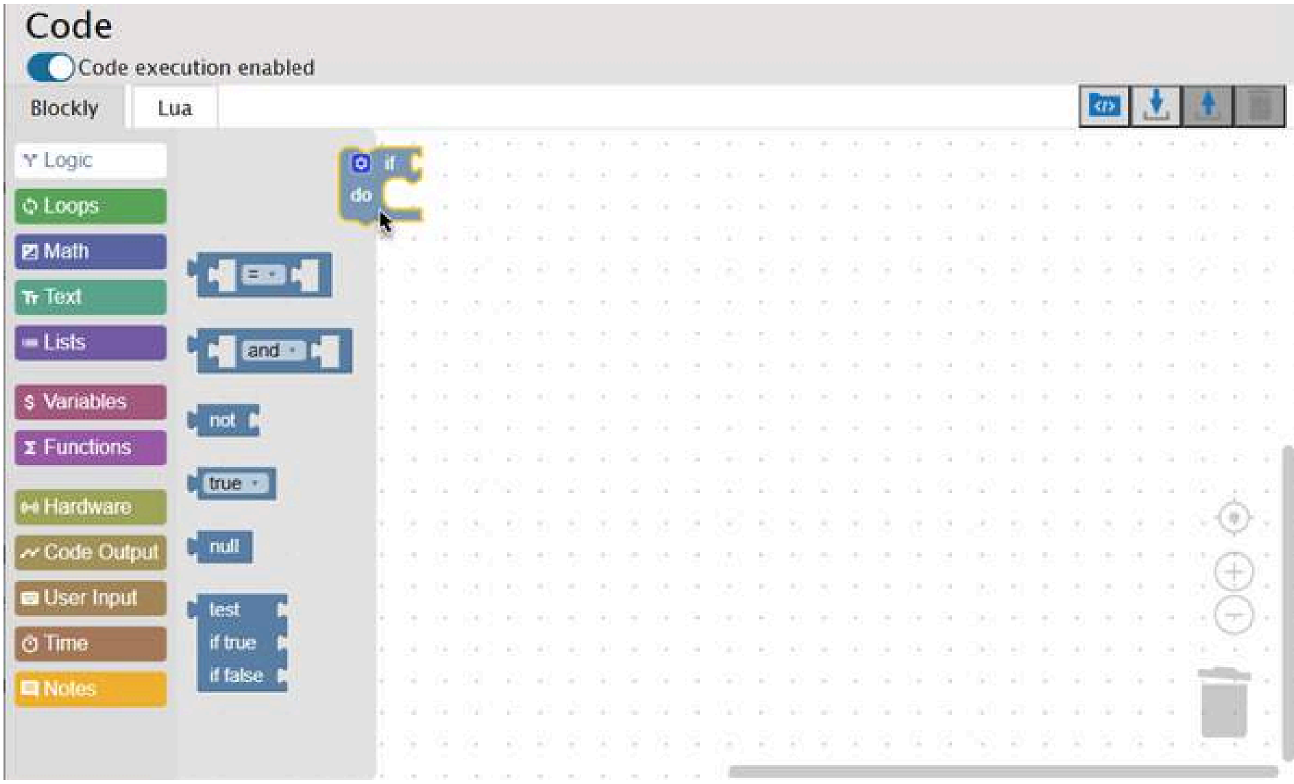
In this tutorial, we're going to build a program to output a message when a light sensor detects darkness. Though this tutorial is written specifically for a light sensor, you can use alternative sensors by making some minor modifications to the code.

Step 1: Set up the sensor



1. Connect the sensor to SPARKvue using the **Sensor Data** path.
2. Select the digits and graph template.
3. Click the Code button  to open the Code tool.

Step 2: Add a block to the workspace



1. Select the **Logic** category in the toolbox.
2. Drag an **if** block from the toolbox to the workspace.

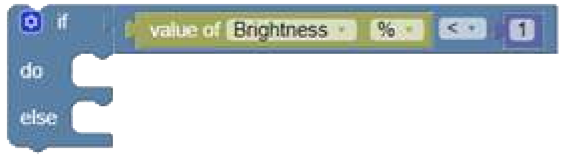
Step 3: Modify the block

1. Click the gear  in the block.
2. Drag the **else** block and connect it to the **if** block.
3. Click the gear  to close it.

Step 4: Connect a second block

1. Select the **Logic** category.
2. Drag the **equality** block and connect it to the **if** block.

Step 5: Create a condition using a sensor data



1. Select the **Hardware** category. Drag the **value of** block and insert it into the **equality** block's first space.
2. Select the **Math** category. Drag a **number** block and insert it into the **equality** block's last space.
3. Click inside the **number** block then enter the number `1`.
4. Click the **equal** sign `=` then select the **less than** sign `<`.

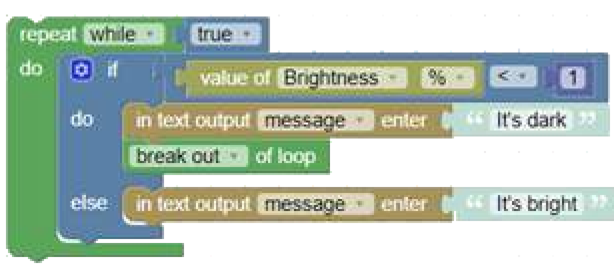
Step 6: Create an output

1. Select the **Code Output** category then click **Create text output**.
2. In the window, enter `message` then click **OK**. This creates a **text output** block.
3. Drag the **text output** block and connect to **do** in the if block.
4. Click inside the **text** block then enter `It's dark`.
5. Drag another **text output** block and connect it to else in the if block.

> **Tip:**
> Alternatively, right-click the first **text output** block and select **Duplicate**.

6. Click inside the **text** block then enter `It's bright`.

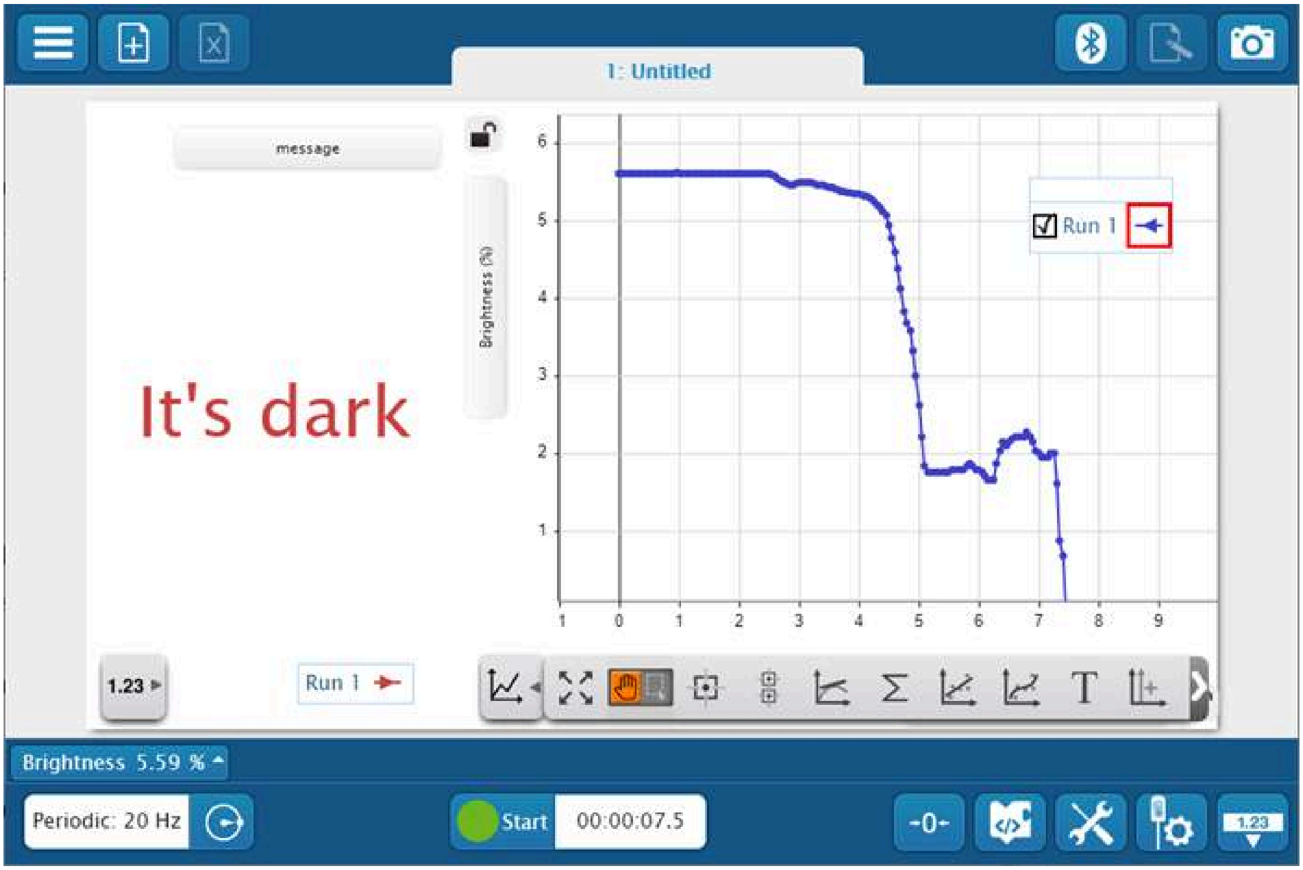
Step 7: Create a loop



1. Select the **Loops** category. Drag the **repeat while** block and place it over the **if** block so it surrounds it.
2. Select the **Logic** category. Drag the **true** block and connect it to the **repeat while** block.

3. Select the **Loops** category. Drag the **break out of loop** block and connect it to the bottom of the **do** statement in the **if** block.
4. Click the Code button to exit the Code tool.

Step 8: Run the program



1. In the Digits display, click the **Brightness** measurement.
2. Select the **User-entered** tab then select **message**.
3. Click **Start** .
4. Cover the sensor to see the message change. This also stops the program.

> **Tip:**
>
> You can also click **Stop** to stop the program manually.

Next steps

Now that you've created your first program, here are some further topics you can explore to take your coding to the next step.

- Explore the Code tool features by looking at the Code tool overview.
- Learn more about how the blocks in this lesson are used or learn about some new blocks in the Blockly toolbox.

- If you have a //code.Node or //control.Node, import programs provided in the PASCO Code Library.
- Download an Blockly Extension experiment from the PASCO Experiment Library.

Build a custom experiment page

Create a custom experiment layout to display data and control hardware.

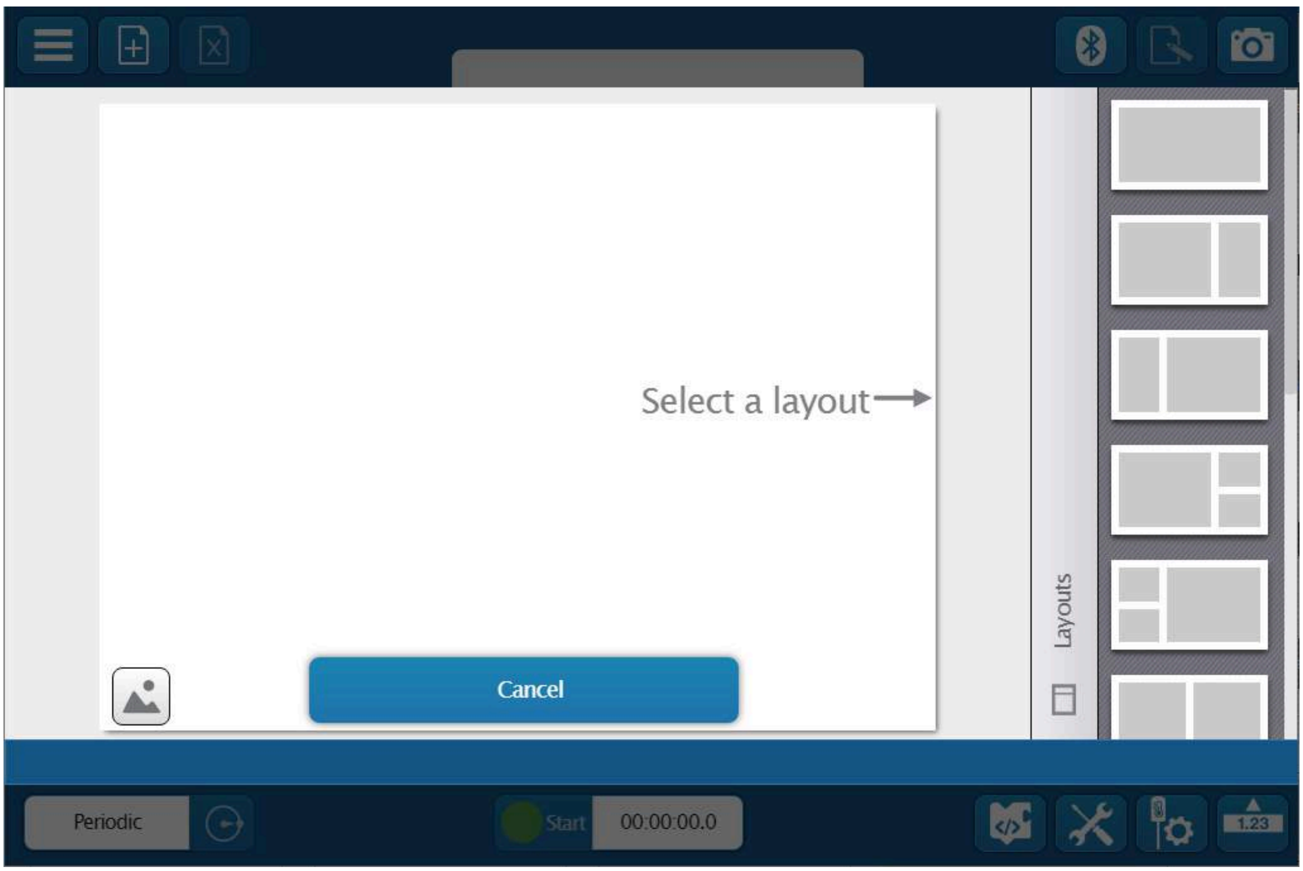
In this tutorial, we're going to build an experiment page from scratch. We will select a page layout, choose displays we want to use to visualize data, and select measurements we want to display.

Step 1: Create a new page

Let's start by creating a new experiment page. This can be done from the Welcome Screen or Experiment Screen:

- If you're on the Welcome Screen, click **Build New Experiment**.
- If you're on the Experiment Screen, click New Page .

Step 2: Select a layout



1. You can add a custom background to your page, but it isn't necessary. If desired, click Add Background to add a background image to your page.
2. Select a layout on the right side of the screen containing one or more display elements. Scroll for additional layout choices.

> **Note:**
> Some displays require a minimum size. See Display types for more information.

Step 3: Connect sensors

Before you can select your displays, you need to connect each sensor you want to use in your experiment. Follow the steps below to connect a Wireless Sensor. If you have a PASPORT or Science Workshop sensor, see the Connect a sensor to SPARKvue page.

1. Turn on the sensor.
2. Click to open the **Wireless Devices** window.
3. Select the sensor that matches its device ID.
4. Click **Done**.

Step 4: Select displays and measurements

1. Select a display for each area on the experiment page. If you want to leave a display element blank, click the ×.

2. For each display, click **Select Measurement**, then select the measurement you want to display from the menu.

> **Tip:**
> If you made a mistake when creating your experiment page (such as selecting the wrong layout) click **New Page** to start a new page. You can delete the old page by clicking **Delete Page**.

Step 5: Collect data

After you have set up your displays, you are ready to collect data. Click **Start** to begin collecting data.

For more information...

- See the Experiment Screen page for information on how to use the tools on the screen.
- See Display and analyze data for information on how to analyze data in each display.

Quick reference

Browse quick reference guides to get an overview SPARKvue's most used features.

Welcome Screen

Choose a path on the Welcome Screen to quickly start an experiment.

1 **Main Menu**

Start or join a Shared Session, set preferences, and check for updates.

2 **Manual Entry**

Enter non-sensor data into a table and display in a graph.

3 **Sensor Data**

Connect sensors and select measurements to display in common layouts.

4 **Remote Logging**

Collect data directly on a Wireless Sensor instead of a computer or mobile device. Download remotely logged data to SPARKvue for analysis.

5 **Open PASCO Experiment**

Collect data using SPARKvue experiment files designed by the PASCO curriculum team.

6 **Build New Experiment**

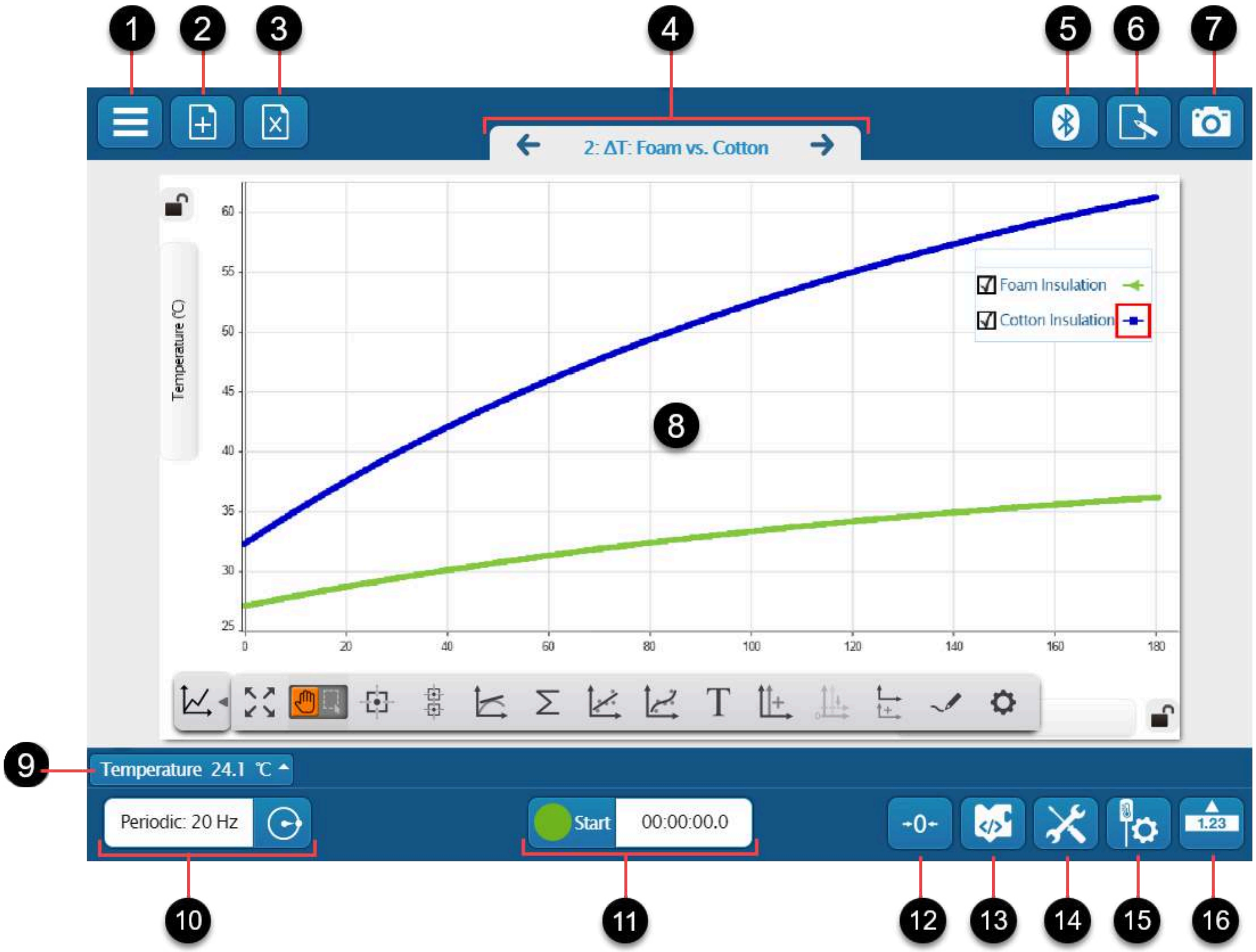
Create a custom experiment layout to display data and control hardware.

7 **Open Saved Experiment**

Access previously saved files on your computing device.

Experiment Screen

Use the Experiment Screen to collect, display, and analyze data.



❶ **Main Menu**

- **Start New Experiment**: Return to the Welcome Screen.
- **Experiments**: Open a PASCO Experiment.
- **Open**: Open a previously saved experiment.
- **Save** or **Save As**: Save the current file to your computing device.
- **Export Data**: Export data as a CSV file.
- **Preferences**: Toggle auto-hide for new runs on all experiment pages.
- **Shared Session**: Start or join a Shared Session.

❷ **New Page**

Build a new experiment page after the one currently displayed on the screen.

❸ **Delete Page**

Delete the currently displayed page.

4 **Page Name and Number**

Click to change the name of the current page or to select a different page to display. Click the arrows to navigate between pages.

5 **Wireless Devices**

Connect a wireless sensor and set up remote data logging.

6 **Journal**

Create and edit the snapshot Journal. Print or export the Journal after making your edits.

7 **Journal Snapshot**

Take a snapshot of the current screen and add it to the Journal.

8 **Experiment Page**

Display and analyze experiment data in multiple ways such as a graph or table. Control various hardware such as signal generators and Smart Cart accessories.

9 **Live Data Bar**

See the live measurement reading for measurements currently selected in the experiment page. Click to zero, change the sign, calibrate, and configure the measurement properties.

10 **Sampling Options**

Displays the current sample rate or sampling mode. Click to configure the following:

- Sampling mode (Periodic or Manual)
- Sampling rate
- Stop Condition

11 **Data Recording Controls**

- Click to **Start** or **Stop** data collection. The clock displays the elapsed data collection time.
- When using the Code tool without sensors connected, the **Start** button changes to **Execute**. Click to run the program in the Code tool.

12 **Zero Measurements**

Click to zero all measurements of connected sensors.

Note:
Not all sensors support zeroing measurements.

Tip:
Use the Live Data Bar to zero measurements individually.

13 **Code**

Click to open the Code tool to create programs using Blockly.

14 **Experiment Tools**

- Manage Runs: Delete or rename a run.
- Manage Images: Delete and rename images taken with the Camera display.
- Calculated Data: Create calculations to transform data.
- Data Properties: Change the number format (such as the number of decimal places), change the default units, and apply one color to all runs.

15 **Hardware Setup**

Configure sensors and their measurements:

- Sensors: Turn on or off, zero, and change the sign.
- Measurement: Calibrate the measurement and access the Data Properties.

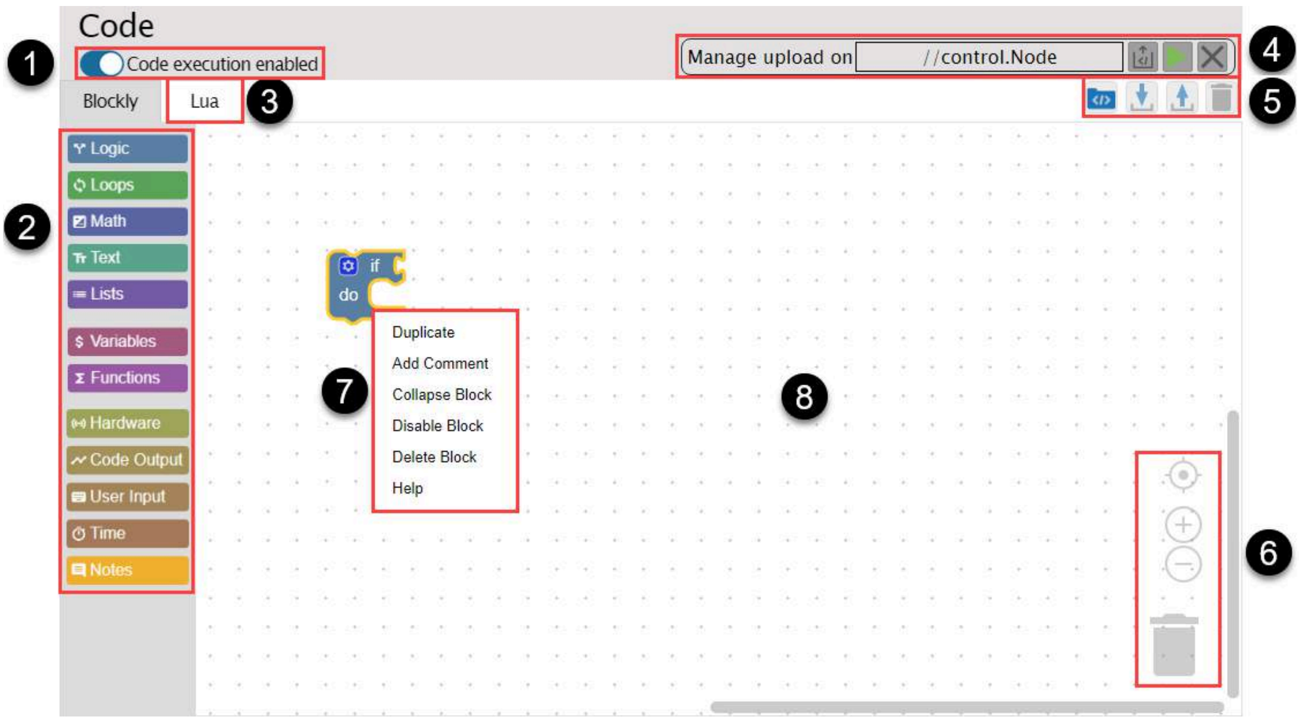
16 **Live Data Bar Visibility**

Click to show or hide the Live Data Bar.

Code tool

The Code Tool interface consists of a workspace to build a program using blocks from the Blockly Toolbox.

To access the Code tool, click  on the bottom of the Experiment Screen.



1 **Code execution switch** 

Toggle to disable code from running a program during data collection. This can be useful when you want to test sensor measurements without having the code interrupt data collection.

2 **Blockly Toolbox**

Click a category to add a block to the workspace to build a program.

3 **Lua code viewer**

Click the Lua tab to view, but not edit, what the program looks like in text-based code to help with debugging.

4 **//control.Node upload tools**

Upload code to the //control.Node so that it can run independently from a computing device. This toolbar appears when a //control.Node is connected.

5 Code Tools

- Import code from the PASCO Code Library. This icon appears when a //code.Node or //control.Node is connected.
- Import code from a .pcbx file.
- Export all code as a .pcbx file for importing into other SPARKvue or Capstone files.
- Delete all code on the workspace.

6 Workspace Tools

- Center the code on the workspace.
- Zoom in on the workspace.
- Zoom out of the workspace.
- Delete blocks by dragging them to the trash can. Click the trash can to retrieve previously deleted blocks.

❼ Block tools

Right-click the block to access the block tools menu. If using a mobile device, tap and hold the block. What appears in the menu depends on the block. The following tools are common to most blocks:

- **Duplicate**

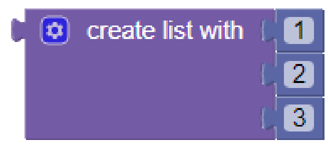
  This creates a copy of the block.

- **Add Comment**

  This adds a comment icon  to the block. Click the icon to add a note to the block to help others understand what your code does.

- **External and Inline Inputs**

  This option appears for blocks that have more than one input. Selecting **External Inputs** will make input blocks appear as a vertical list outside of the block. Selecting **Inline Inputs** will make input blocks appear inside the block has a horizontal list.



- **Collapse and Expand Block**

  This collapse all the code inside the block into a single line. This can be useful for de-cluttering the workspace to make the code easier to read. Select **Expand Block** to display the code again.

- **Disable Block**

  This disables the block and the code inside the block. This can be useful for debugging a program.

- **Delete Block**

  This deletes the block and all code inside the block.

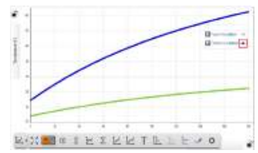
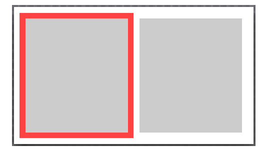
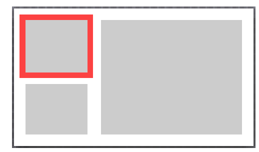
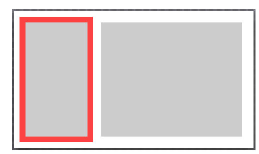
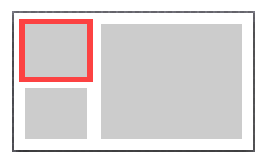
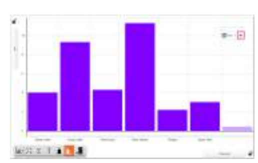
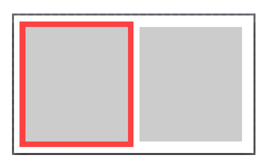
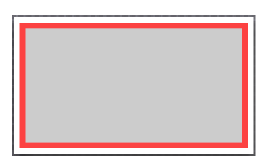
❽ Workspace

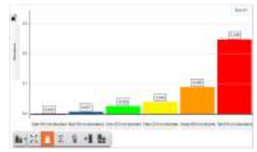
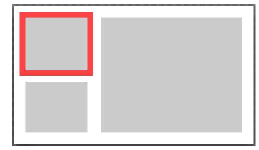
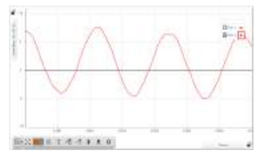
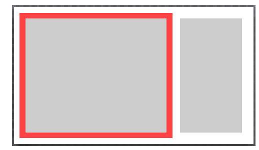
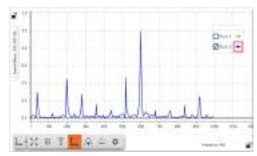
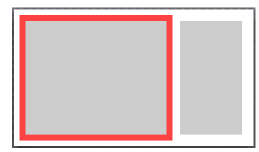
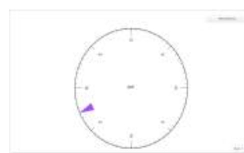
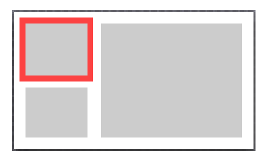
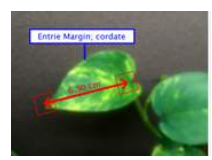
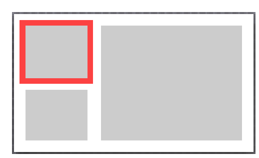
Drag blocks from the toolbox to the workspace to create a program.

Display types

Use a display to visualize data, control hardware, and add content.

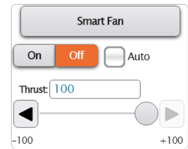
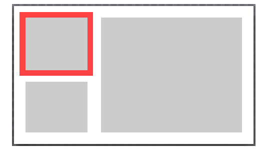
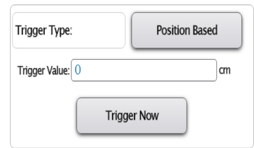
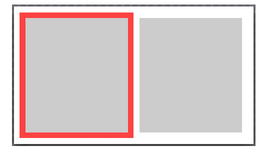
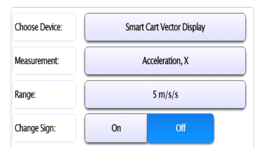
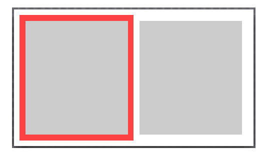
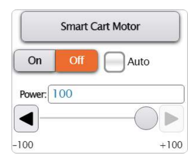
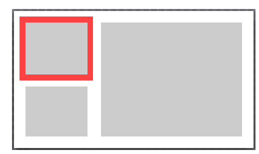
Visualize data

Type	Icon	Application	Preview	Minimum Size
Line Graph		Display data in a XY line graph, showing one or more measurements versus time or another measurement.		
Digits	1.23	Show the most recently recorded data sample as digits, statistics of a single data run, or text output from a Blockly program.		
Table		Display data using two or more columns and show data statistics.		
Meter		Show the most recently recorded data sample or a statistical value of a single run on an analog display.		
Bar Graph		Compare manually-sampled measurements across categories.		
Map		Tag sensor measurements with GPS data and display the measurement locations on a map.		

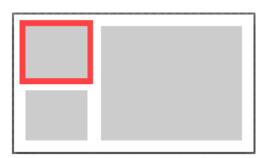
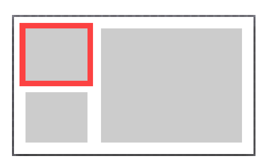
Type	Icon	Application	Preview	Minimum Size
Bar Meter		Visually compare the most recently recorded data values across similar measurements.		
Scope		View data that oscillates rapidly in time such as electrical signals and sound waves.		
FFT		Display data in a spectral analysis display to measure the intensity as a function of frequency.		
Compass		View directional measurements from the Wireless Weather Sensor such as wind direction and heading.		
Camera		Capture and analyze still images from a webcam or digital microscope.		

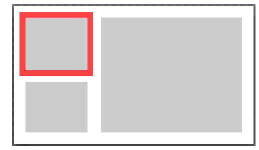
Control hardware

Type	Icon	Application	Preview	Minimum Size
Signal Generator		Control the output waveform type, frequency, and amplitude on the Wireless AC/DC Module or 550 Universal Interface.		

<table>
<tr><th>Type</th><th>Icon</th><th>Application</th><th>Preview</th><th>Minimum Size</th></tr>
<tr><td>Smart Fan</td><td></td><td>Control the thrust and direction of a Smart Fan Accessory attached to a Smart Cart.</td><td>
</td><td></td></tr>
<tr><td>Ballistic Cart</td><td></td><td>Launch a projectile from the Smart Ballistic Cart Accessory by setting a position or time based trigger.</td><td>
</td><td></td></tr>
<tr><td>Vector Display</td><td></td><td>Change the measurement displayed on a Smart Cart Vector Display and configure the range and sign of the measurement.</td><td>
</td><td></td></tr>
<tr><td>Smart Motor</td><td></td><td>Control the speed and direction of a Smart Cart Motor attached to a Smart Cart.</td><td>
</td><td></td></tr>
</table>

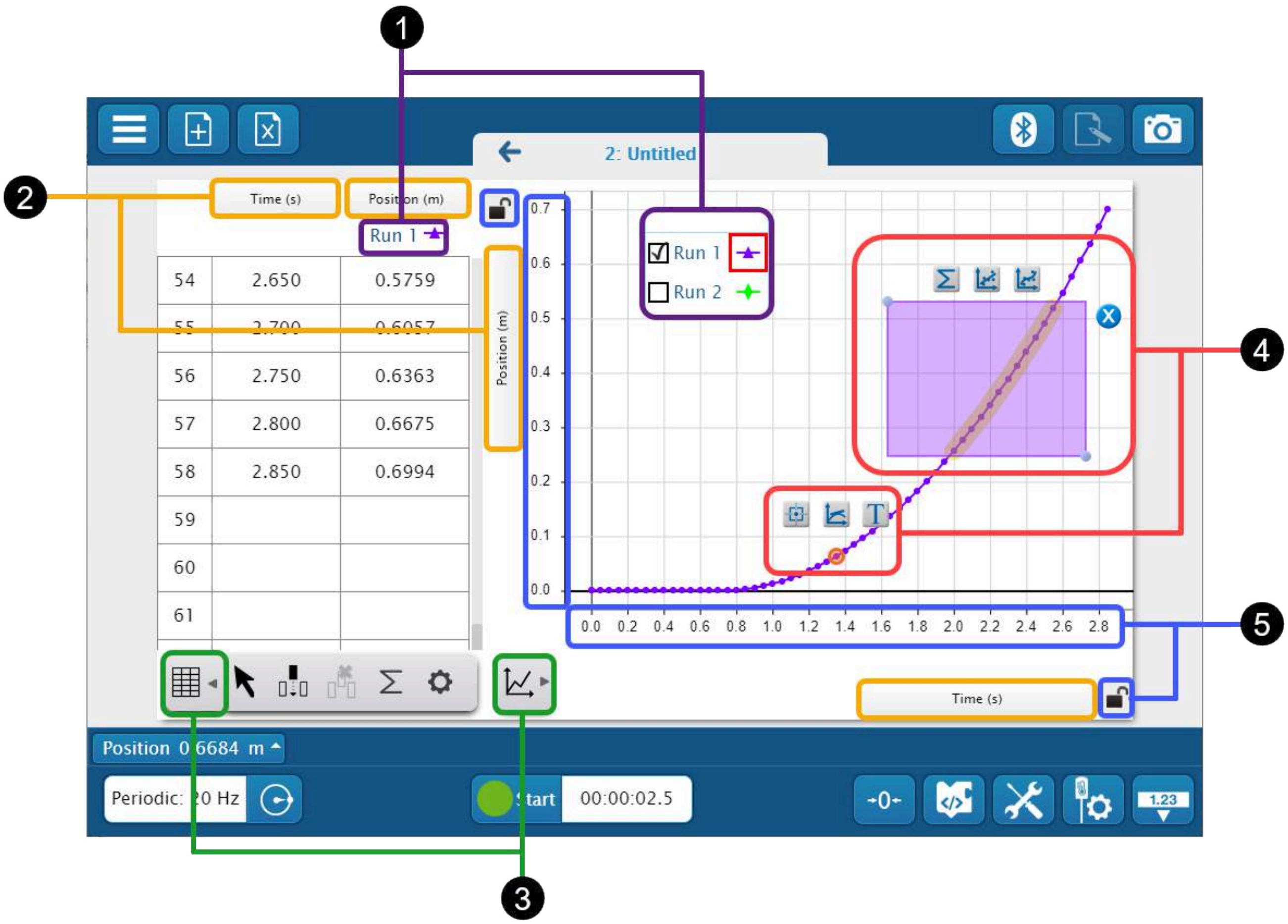
Add content

<table>
<tr><th>Type</th><th>Icon</th><th>Application</th><th>Preview</th><th>Minimum Size</th></tr>
<tr><td>Media</td><td></td><td>Insert an existing still image.</td><td></td><td></td></tr>
<tr><td>Text Box</td><td>T</td><td>Insert text for experiment notes or instructions.</td><td>
</td><td></td></tr>
</table>

Type	Icon	Application	Preview	Minimum Size
Assess-ment		Create multiple-choice, drop-down, or short-answer questions.		

Data display tools

Each data display includes common tools that help with analyzing data and modifying the appearance. Graph displays (Line Graph, Bar Graph, Scope, and FFT) include additional tools that are specifically useful for graphs.



1 **Run selector**

Click to select a run to show in the display.

For graph displays, click the check box to show ☑ or hide ☐ a run. When analyzing a run, first select the run color to make the run active. The active run is indicated by a red box.

2 **Measurement selector**

Click any measurement in a display to change the measurement or its units. In the measurement selection pane, select a measurement to display or select the unit to change the units. If you want to display user-entered or calculated data, select the **User-entered** tab at the top of the pane.

3 **Toolbar drawer**

Click to show or hide the display's toolbar. Each display includes analysis tools specific for that display. Select a tool in the drawer to activate it in the display.

❹ In-context tools

Graph displays include analysis tools in the plot area for easy access. These are the same tools found in the toolbar. The tools that appear depends on whether you select a single data point or drag over a range of data.

- **Single data point**

  Select a single data point then select either the coordinates , slope , or annotation tool. After selecting the tool, you can select a different data point by clicking the left and right arrows. Click to remove the tool.

- **Range of data**

  Drag over a range of data then select the statistics , linear fit , or curve fit tool. Click to remove the tool.

❺ Axes scale

Drag across an axis to change the scale of the axis. To prevent the axis from scaling, click the lock . You can also drag the plot area to move the plot.

How do I?

Get quick answers to specific questions with step-by-step instructions.

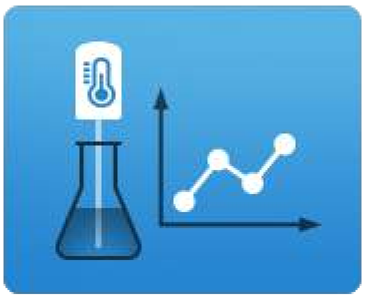
Set up data collection

Connect a sensor, manually enter data, and open an experiment.

Set up and display sensor data

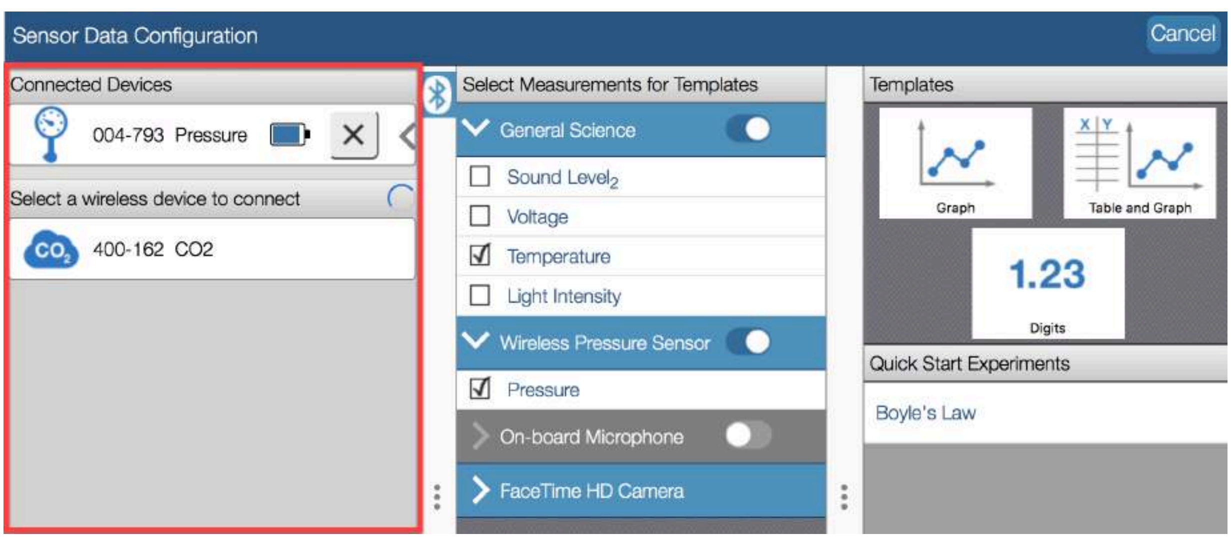
Connect sensors and select measurements to display in common layouts.

1. Open SPARKvue or click  then select **Start New Experiment**.
2. Click **Sensor Data**:



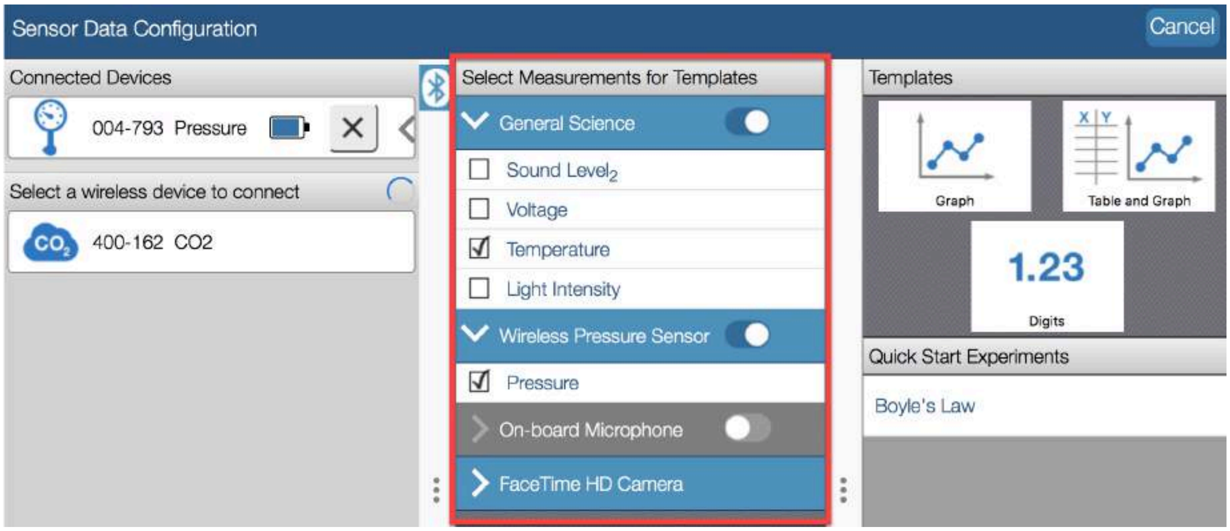
3. Connect a Wireless Sensor:
   a. Turn on the sensor.
   b. Select the sensor that matches its device ID.

> **Note:**
> For PASPORT and Science Workshop sensors, see Connect a sensor to SPARKvue.

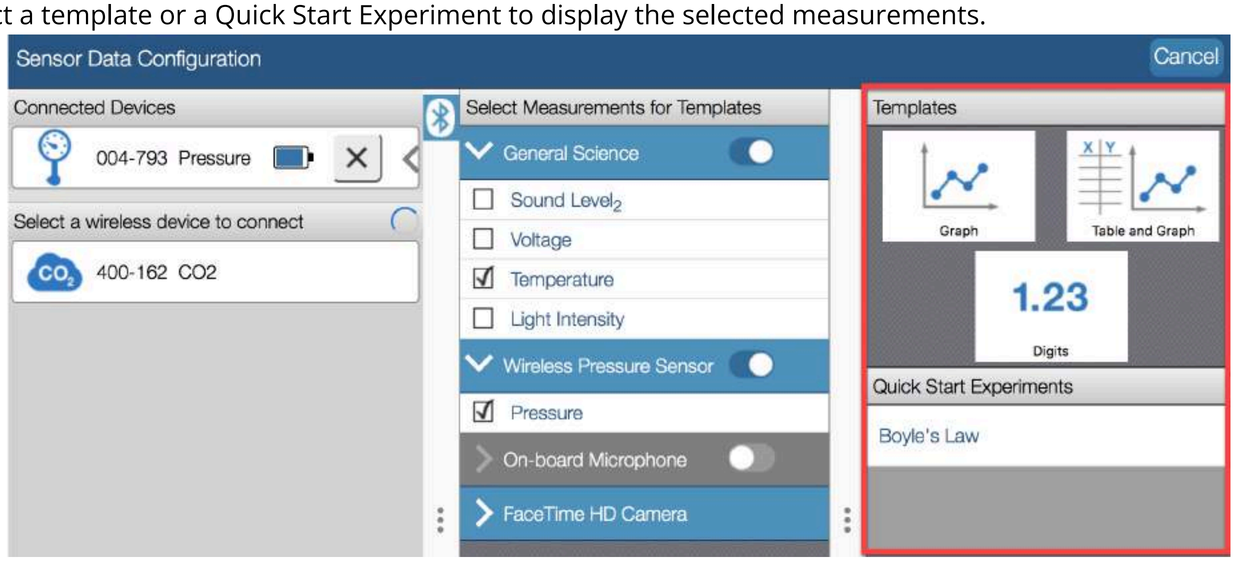


4. Select up to three measurements to display.

> **Tip:**
>
> - Toggle the switch to disabled to turn off sensors you don't need in your experiment.
> - Toggle the arrow to show or hide sensor measurements.



5. Select a template or a Quick Start Experiment to display the selected measurements.



6. On the Experiment Screen, click **Start** to begin collecting data.

See Display and analyze data for information on how to analyze data in each display.

Remotely log data

Collect data directly on a Wireless Sensor instead of a computer or mobile device.

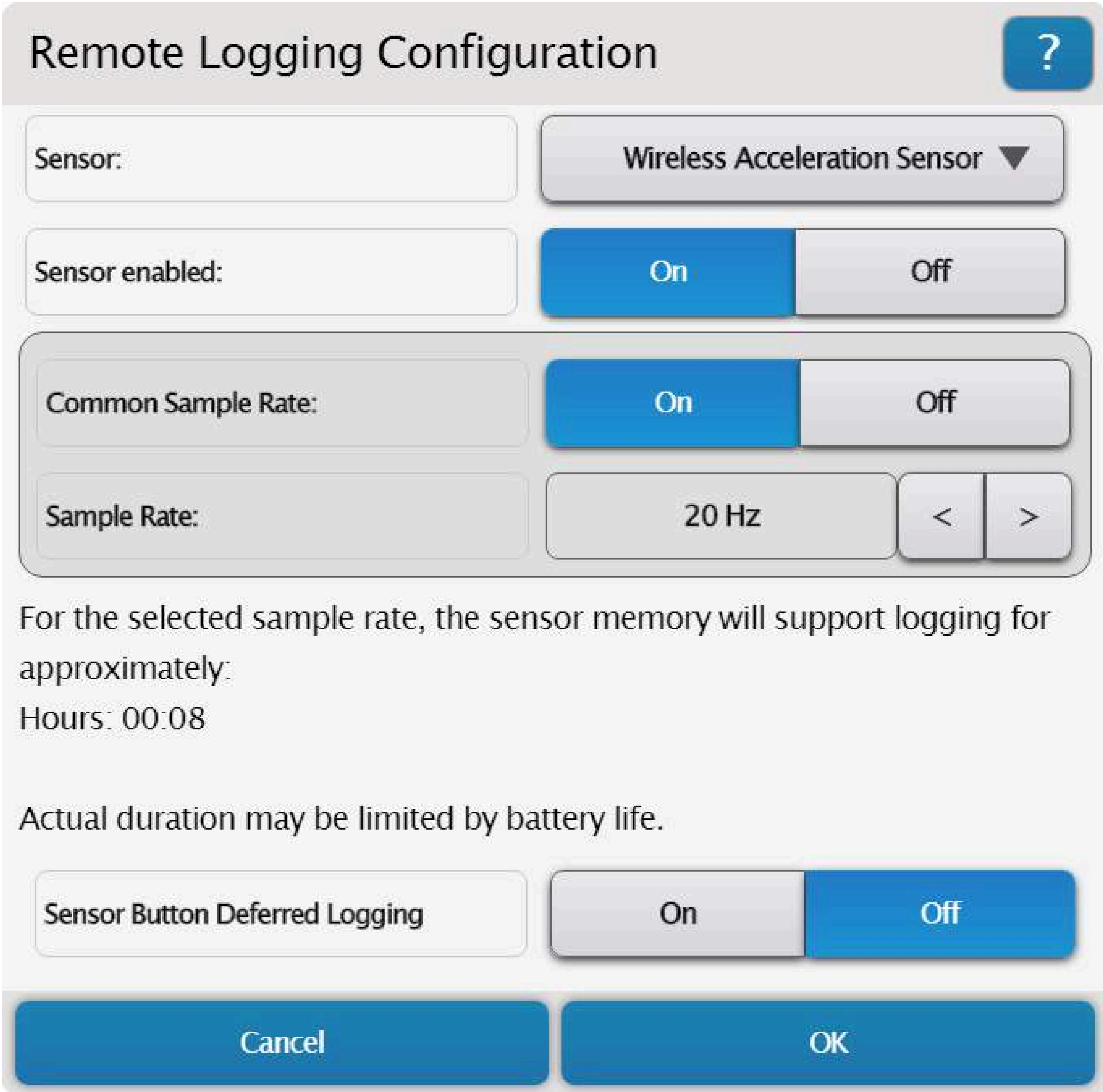
> **Note:**
> Remote Logging is only available for PASCO Wireless Sensors.

Set up remote logging

1. Open SPARKvue or click then select **Start New Experiment**.
2. Click **Remote Logging**:

3. Turn on the sensor then click the sensor which matches the device ID.

4. Configure remote logging for each sensor:



a. Select a sensor to configure from the **Sensor** menu.
b. Toggle **Sensor Enabled** to **Off** if you don't want to log data with this sensor.
c. Set the **Sample Rate** using the left and right arrows. Toggle **Common Sample Rate** to **Off** to set different sample rates for each sensor.

Tip:
The configuration window indicates the amount of time that the sensor can log data below the sample rate. To increase the logging time:

- Decrease the sample rate.
- Disable unused sensors.

5. Optional: Toggle **Sensor Button Deferred Logging** to **On** to start data logging by pressing the power button on the sensor.

6. Click **OK**.

Data logging begins immediately after you click **OK** or press the power button on the sensor (if you selected **Sensor Button Deferred Logging**). The Bluetooth status light blinks yellow and green until data logging begins. When the sensor starts logging data, the Bluetooth status light blinks yellow.

Click **OK** and close SPARKvue. To stop data logging, turn off the sensor or connect it to SPARKvue to download the data.

Download remotely logged data

Download data remotely logged on a Wireless Sensor for data analysis. You can download the data to multiple devices as long as data isn't deleted from the sensor after downloading it.

1. Open SPARKvue or click then select **Start New Experiment**.
2. Click **Remote Logging** .
3. Turn on the sensor or press the power button if the sensor is currently logging.

> **Note:**
> The sensor doesn't appear in **Wireless Devices** when the Bluetooth status light blinks yellow. Press the power button to make the sensor appear.

> **Tip:**
> Connect the sensor using USB, if available, to download data at a faster rate.

4. Select the sensor under **Sensors with data**.

5. In the **Logged Data** window, select **Download Data**.
6. Select a method to download the data:

- **Templates**

  Use this method to download the data into a new file.

  a. In the **Select Measurements for Templates** panel, select ☑ up to three measurements to display.
  b. In the **Templates** panel, select a template or a Quick Start Experiment to display the selected measurements.

- **Quick Start Experiments**

  Use this method to download the data to a new Quick Start Experiment file. Names of Quick Start Experiments appear if available for the connected sensor.

  Select a Quick Start Experiment from the list, if available.

- **Add to existing experiment**

  Use this method to download the data to an existing experiment file.

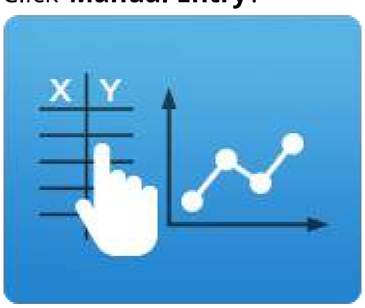
  a. Click Open PASCO Experiment or Open Saved Experiment.
  b. Select a file to open.

> **Note:**
> It may take a few minutes to download the data depending on the amount of data logged.

Manually enter data in a graph

Enter non-sensor data into a table and display in a graph.

1. Open SPARKvue or click ☰ then select **Start New Experiment**.
2. Click **Manual Entry**:



3. Enter a data point for x and y in row 1 of the table. A data point appears on the graph.

4. Repeat for additional data points.

> **Tip:**
> Use the **Tab** key on your keyboard to quickly move between cells when entering data.

5. Use the graph and table tools to change the appearance or analyze the data.

Open a PASCO experiment file

Collect data using SPARKvue experiment files designed by the PASCO curriculum team.

1. Open the PASCO Experiments file browser:
    - On the Welcome Screen, select **Open PASCO Experiment**.
    - On the Experiment Screen, click then select **Experiments**.
2. Select an experiment file then click **OK**.
    - **Essential Chemistry**

      This directory contains files designed for the Essential Chemistry lab manual. You can download worksheets for each lab for free from the PASCO Experiment Library.
    - **Essential Physics**

      This directory contains files designed for the Essential Physics lab manual. You can download worksheets for each lab for free from the PASCO Experiment Library.
    - **Quick Start Labs**

      This directory contains files configured for specific PASCO sensors, but not necessarily associated with any written lab activities. Connect a sensor using the Sensor Data path from the Welcome Screen to select files designed for that sensor.
    - **SPARKlabs**

      Files contained in the main directory are complete labs containing data collection steps and analysis questions. These files don't have a corresponding print worksheet.
3. Connect a sensor to SPARKvue.
4. Click **Start** to begin collecting data.

See the Experiment Screen page for additional information on how to use the tools on the screen.

Open a saved experiment

Access previously saved files on your computing device.

1. Open your computing device's file explorer:
    - On the Welcome Screen, select **Open Saved Experiment**.
    - On the Experiment Screen, click then select **Open**.
2. Use your computing device's file explorer to find and open the file.
3. Connect a sensor to SPARKvue.
4. Click Start

   to begin collecting data.

See the Experiment Screen page for additional information on how to use the tools on the screen.

Connect a sensor to SPARKvue

Connect PASCO Wireless, PASPORT, and Science Workshop sensors to SPARKvue from the Experiment Screen.

> **Tip:**
> Use the Sensor Data path to connect sensors from the Welcome Screen.

Wireless Sensors

Connect Wireless Sensors to SPARKvue using Bluetooth or USB:

- USB: Connect the sensor directly to a computing device using a USB cable.
- Bluetooth:
    1. Turn on the sensor.
    2. Click  to open the **Wireless Devices** window.
    3. Select the sensor that matches its device ID.
    4. Click **Done**.

PASPORT Sensors

Connecting PASPORT sensors to SPARKvue requires a compatible USB or Bluetooth interface.

1. Power on the interface.
2. Connect the interface to your computing device using USB or Bluetooth:
    - USB: Connect the interface directly to a computing device using a USB cable.
    - Bluetooth:
        a. Pair the interface with your computing device using your device's Bluetooth settings.

        > **Note:**
        > Skip this step if using an AirLink Interface (PS-3200).

        b. Click  to open the **Wireless Devices** window.
        c. Select the interface that matches its device ID.
        d. Click **Done**.
3. Connect the sensor to a PASPORT port on the interface.

> **Tip:**
> Click  to open **Hardware Setup** to confirm that the sensor connected successfully.

Science Workshop Sensors

Connecting Science Workshop sensors to SPARKvue requires a 550 Universal Interface or a PASPORT interface with a connected Digital or Analog adapter.

Note:
The 850 Universal Interface and legacy Science Workshop interfaces aren't compatible with SPARKvue.

1. Power on the interface.
2. Connect the interface to your computing device using USB or Bluetooth:
   - USB: Connect the interface directly to a computing device using a USB cable.
   - Bluetooth:
     a. Pair the interface with your computing device using your device's Bluetooth settings.

     Note:
     Skip this step if using an AirLink Interface (PS-3200).

     b. Click [Bluetooth icon] to open the **Wireless Devices** window.
     c. Select the interface that matches its device ID.
     d. Click **Done**.
3. Connect the sensor to a Science Workshop port on the interface then:
   - **Analog Sensors**
     a. Click [Hardware Setup icon] to open **Hardware Setup**.
     b. Click [gear icon] for the port that you connected the sensor to.
     c. In the **Assign Analog Adapter** window, select the connected sensor then click **OK**.

     Tip:
     For analog sensors, the default gain setting of Low (1x) works for most applications.

     d. Close **Hardware Setup**.
   - **Digital Sensors**
     a. Select a menu item to expand the list.
     b. Select the device or timer from the list.
     c. Enter the properties for the device or timer, if prompted.
     d. Click **OK**.

Display and analyze data

Display data multiple ways such as in a line graph, a table, or a digital meter. Learn how to use each display and its tools to analyze data.

Graph

Display data in a XY line graph, showing one or more measurements versus time or another measurement. Modify how the data appears and analyze the data using multiple tools.

Graph display overview

Get an overview of the Graph display features including the analysis tools and how to modify the display.

1 **Axis Lock**

Click to lock or unlock axis scaling.

2 **Measurement Selector**

Click to change the displayed measurement or units.

3 **Axis**

Click and drag to scale each axis individually.

4 **Plot Area**

Click and drag to move the plot area.

5 **Legend**

Select a run to analyze by clicking the run in the legend. Right-click the run symbol to change the run color and other data set properties.

6 **Tools Drawer**

Click to show or hide the Graph tools.

7 **Scale to Fit**

Change the scale of a line graph to zoom in on a range of data or to view an entire run of data.

8 **Cursor Mode**

Click to toggle the function of the cursor.

- Switch to Move Mode to use the cursor to move the plot area.
- Switch to Select Mode to highlight an area on the plot for data analysis.

9 **Coordinates Tool**

Use to determine the coordinates of a single data point. Use the delta tool to determine the difference between two points of data.

Multi-Coordinates Tool

Use to compare coordinates across multiple runs or measurements.

10 **Slope Tool**

The slope tool provides the slope of a line tangent to a curve at a single point.

11 **Statistics**

Use the statistics tool to find the minimum, maximum, mean, and standard deviation of a range of data.

12 **Linear Fit**

Use the linear fit tool to fit a line to data and determine the slope and y-intercept of the trend line.

Curve Fit Tool

Use the curve fit tool to fit a curve to data and obtain values about the trend line parameters.

13 **Annotation Tool**

Use the annotation tool to label a data point in a graph with a brief note.

14 **Add y-axis**

Display multiple measurements on the same plot area by adding additional y-axes to a line graph. Align the axes so that the origin is the same for each axis.

Align Origins

When displaying more than one y-axis, select to align the origins and scales of each axis.

Add Plot Area

Display an additional measurement in a line graph on separate plot area located below the current plot area.

15 **Prediction Tool**

Before collecting data, draw a prediction of the results directly on the graph.

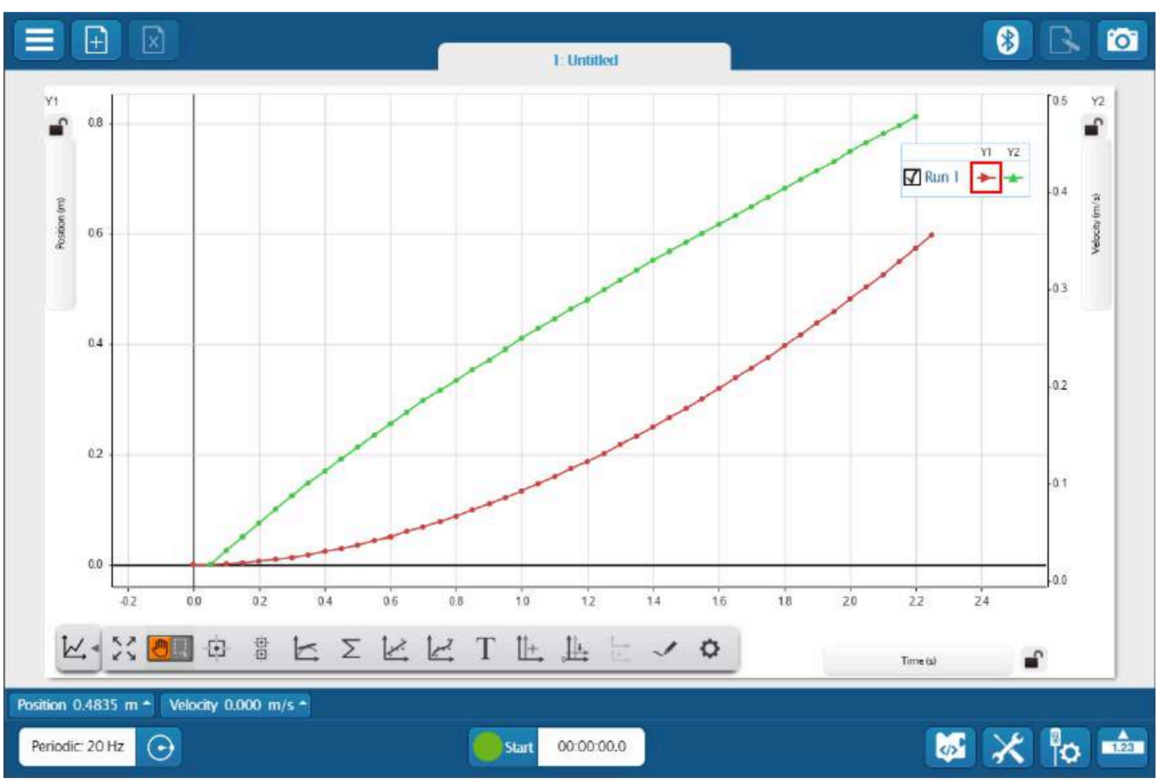
16 **Properties**

Change the visibility of data points, symbols, and lines connecting data points in a line graph.

Add or remove an additional y-axis

Display multiple measurements on the same plot area by adding additional y-axes to a line graph. Align the axes so that the origin is the same for each axis.

> **Tip:**
> If you want to add a another plot area below the current plot area, see Add a Plot Area.



Add an axis

1. Add a y-axis using one of the following methods:
   - Go to then click .
   - Click the y-axis of the current graph then click at the top.
2. On the new axis, click Select Measurement then select a measurement from the list.

Align the axes

1. Go to then select . This aligns the origins and the scales of each axis.
2. Optional: If you want to have different scales for each axis, click to lock the axis then scale the other axis.

Remove an axis

Click the axis you want to remove then click at the top.

> **Note:**
> It's not possible to remove the original graph axis, Y1.

Add or remove a plot area to a line graph

Display an additional measurement in a line graph on separate plot area located below the current plot area.

> **Tip:**
> If you want to add a second y-axis to graph on the same plot area, see Add a y-axis.

1. Add a plot area using one of the following methods:
   - Go to then click .
   - Click the graph then click at the top right corner.
2. In the new plot area, click Select Measurement then select a measurement from the list.

To remove a plot area, click the plot area you want to remove and then click at the top right corner.

Change the scale and range of a line graph

Change the scale of a line graph to zoom in on a range of data or to view an entire run of data.

Automatically scale to view all data

Go to then click .

If multiple runs are visible on the graph, the graph scales to show all data runs. If you want to scale to a single run, hide the other data runs then click .

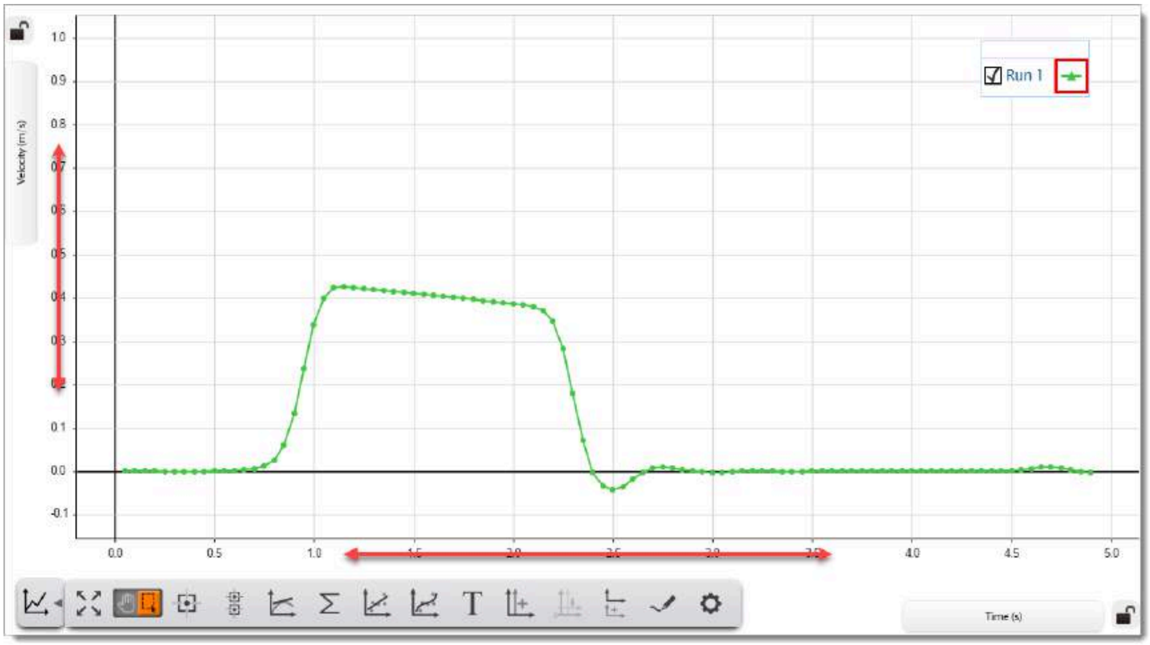
Scale each axis individually

Click and drag on the axis to change the scale of a single axis:

- Drag away from the origin to zoom in.
- Drag toward the origin to zoom out.

To scale each axis individually for graphs with multiple y-axes:

1. Click to lock the scale of one axis.
2. Scale the other axis by clicking and dragging the axis.



Change the range of the graph but not the scale

You can change the range of data displayed in a graph without changing the scale by moving the graph. To move the graph:

1. Go to then toggle the cursor tool to .
2. Click and drag the graph to the desired range.

Zoom in on a range of data

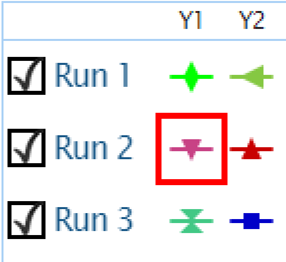
1. Go to then toggle the cursor tool to .
2. Click and drag over the area you want to zoom in.
3. Click .
4. Optional: Remove the selection highlighter by clicking it then clicking .

Show or hide runs in a line graph

Change the visibility of individual data runs without deleting the data.

> **Note:**
> This article explains how to *hide* data runs from display. To *delete* data runs, see Delete data runs.

Use the graph legend to select which runs are visible.



Click to select the runs you want to make visible.

Click to deselect runs you want to hide.

Hide data points and connecting lines

Change the visibility of data points, symbols, and lines connecting data points in a line graph.

1. In the graph toolbar , click Properties .
2. Toggle the Show Connected Lines switch to turn connected lines on or off.
3. Click the Data Point Marker menu then select one of the following:
   - **Hidden**: Don't show data points.
   - **Dot**: Display data points as dots for all runs.
   - **Symbols**: Display data points as symbols. Each run has a unique symbol.

Determine the coordinates of points on a graph

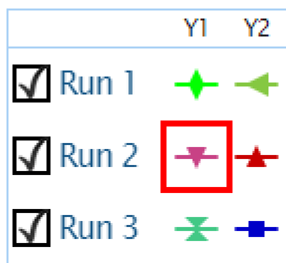
Determine the coordinates of a single data point or compare data point coordinates from multiple measurements and runs.

Coordinates of a Single Point

Use the coordinates tool to determine the coordinates of a single data point.

Figure 1. Using the coordinates tool in a line graph.

1. Select the run you want to analyze in the graph legend. A red border indicates the selected run.



2. Click the point you want to analyze then click .
3. Place the tool on a different data point by click and dragging the tool or clicking  and .

To remove the tool from the screen, click the coordinates tool then click .

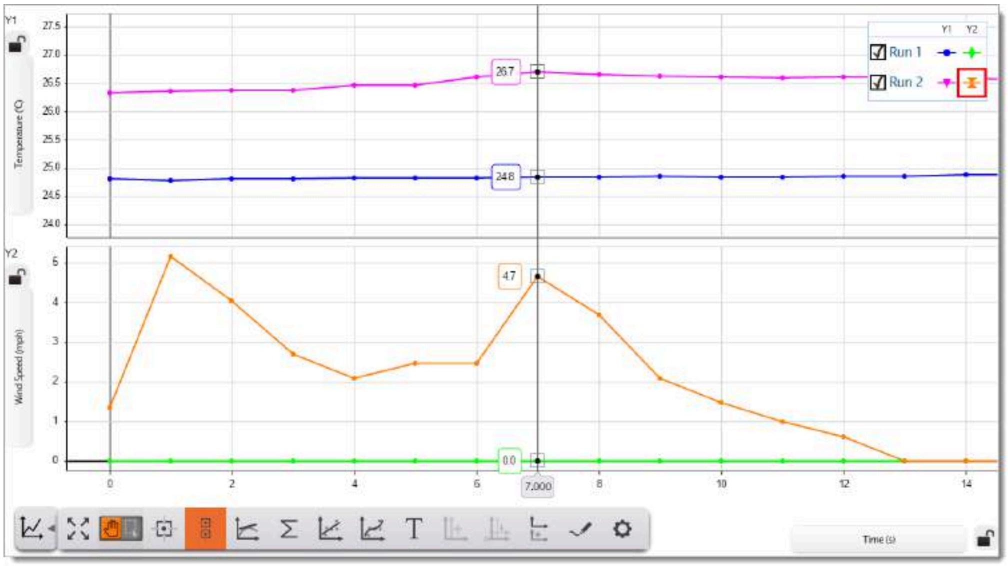
Tip:

Click  to determine the difference between two data points.

Compare Coordinates of Multiple Points

Use the multi-coordinates tool to compare coordinates across multiple runs or measurements.

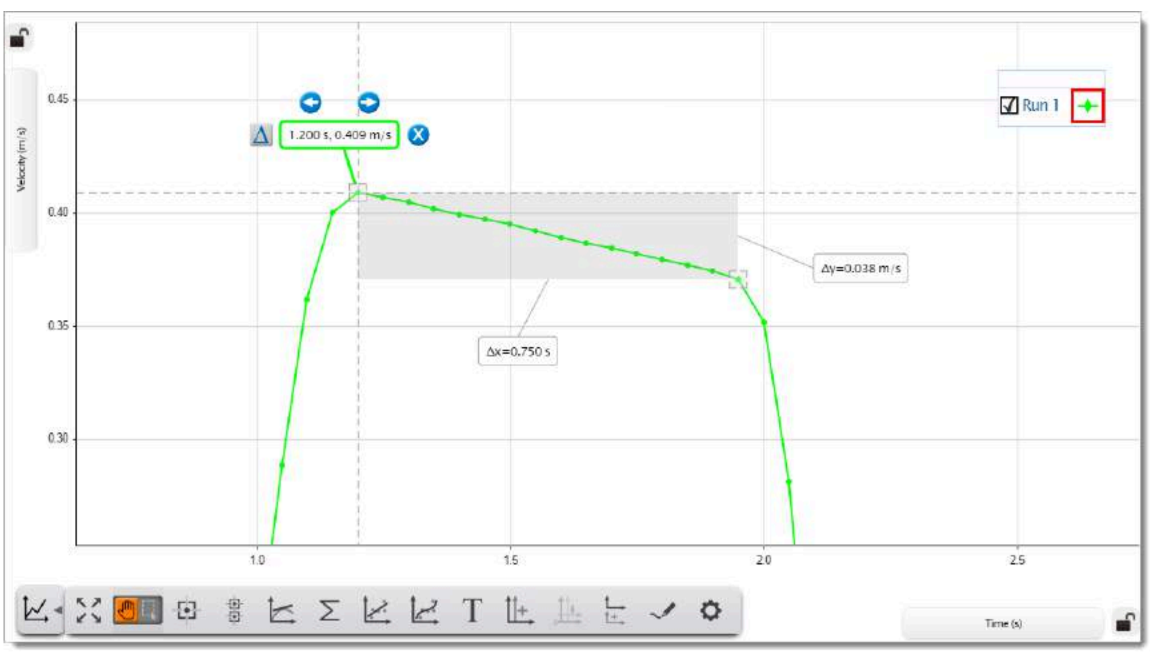
Figure 2. Using the multi-coordinates tool in a line graph.



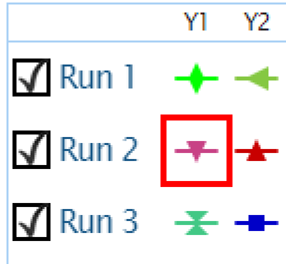
1. Go to [icon] then click [icon] .
2. Place the tool on a different data point by dragging the tool or clicking [icon] and [icon].

Find the difference between two data points

The coordinates tool in a graph display includes a delta tool to determine the difference between two data points in a run of data. Activate the delta tool to display the difference between the vertical, **Δy**, and horizontal, **Δx**, coordinates.



1. Select the run you want to analyze in the graph legend. A red border indicates the selected run.



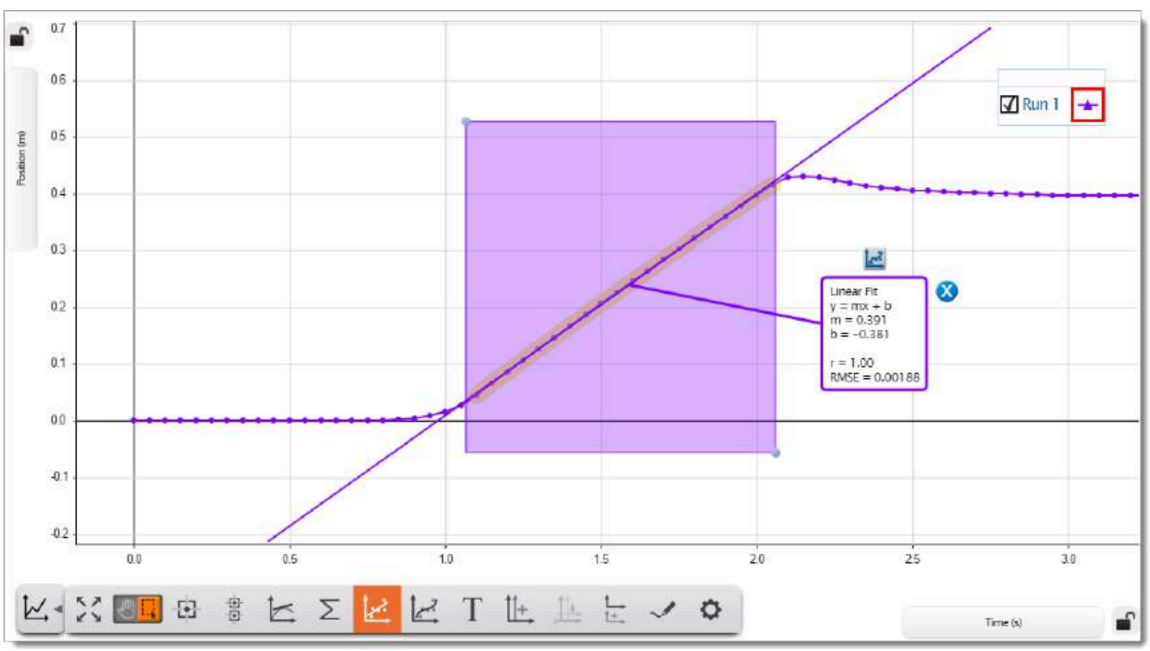
2. Click the point you want to analyze then click .
3. Click to make the delta tool visible.
4. Place the tool on a different data point by dragging the tool or clicking and .

Remove the tool by clicking the tool then click .

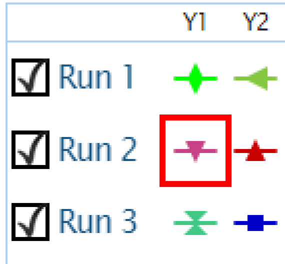
Fit a line to data in a line graph

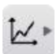
Use the linear fit tool to fit a line to data and determine the slope and y-intercept of the trend line.

> **Tip:**
> Other curve fits are available using the curve fit tool.



1. Select the run you want to analyze in the graph legend. A red border indicates the selected run.



2. Go to  then toggle the cursor tool to .
3. Click and drag over the data you want to analyze.

> **Tip:**
> You can adjust your selection by click and dragging the handles on the top-left and bottom-right corner of the highlighter box. Click and drag the highlighter box to move it.

4. Click .

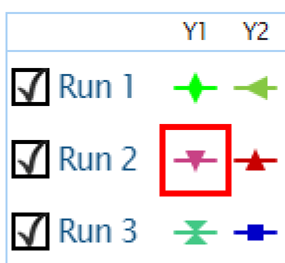
A line appears over the data with an information box. The information box includes the value of the slope, **m**, and y-intercept, **b**, of the best fit line. Also included are the coefficient of determination, **r**, and the root mean square error, **RMSE**, to show how well the line fits the data.

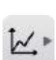
If you wish to try a different fit to your data, click the information box then click .

Fit a curve to data in a line graph

Use the curve fit tool to fit a curve to data and obtain values about the trend line parameters.

1. Select the run you want to analyze in the graph legend. A red border indicates the selected run.



2. Go to  then toggle the cursor tool to .

3. Click and drag over the data you want to analyze.

> **Tip:**
> You can adjust your selection by click and dragging the handles on the top-left and bottom-right corner of the highlighter box. Click and drag the highlighter box to move it.

4. Click .
5. Select the desired fit from the list then click OK.

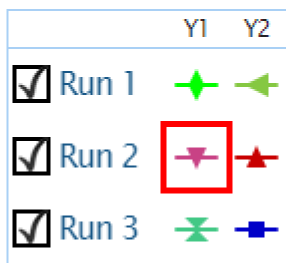
A curve appears over the data with an information box. The information box includes values about the trend line parameters. Also included are the mean square error, **MSE**, and the root mean square error, **RMSE**. which provide information on well the curve fits the data.

If you wish to try a different fit to your data, click the information box then click .

Find the statistics of a range of data in a line graph

Use the statistics tool to find the minimum, maximum, mean, and standard deviation of a range of data.

1. Select the run you want to analyze in the graph legend. A red border indicates the selected run.



2. Go to then toggle the cursor tool to .
3. Click and drag over the data you want to analyze.

> **Tip:**
> You can adjust your selection by click and dragging the handles on the top-left and bottom-right corner of the highlighter box. Click and drag the highlighter box to move it.

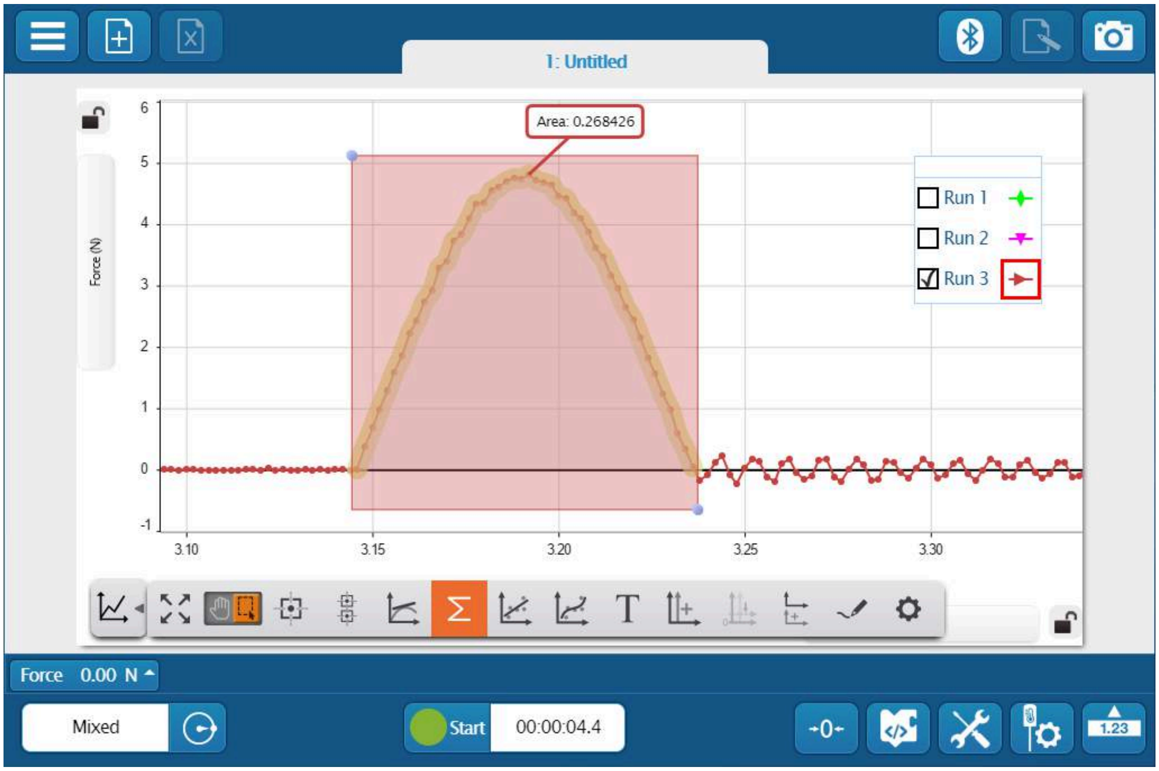
4. Click .
5. Select one or more statistics that you want to display then click OK.

An information box appears containing the statistics you selected. If you wish to display other statistics, click the information box then click .

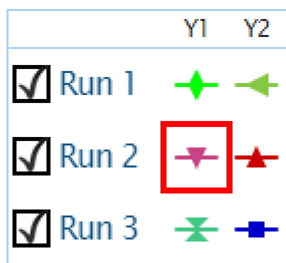
Find the area under a curve

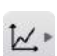
Use the area tool to find the area under a curve.

Figure 3. Using the area tool to find the area under a curve.



1. Select the run you want to analyze in the graph legend. A red border indicates the selected run.



2. Go to  then toggle the cursor tool to .
3. Click and drag over the data you want to analyze.

> **Tip:**
> You can adjust your selection by click and dragging the handles on the top-left and bottom-right corner of the highlighter box. Click and drag the highlighter box to move it.

4. Click .
5. Select Area then click OK.

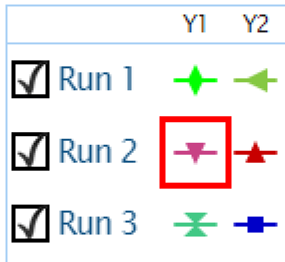
An information box appears containing the area under the curve. If you wish to display other statistics, click the information box then click the statistics button .

Find the slope of the tangent to a curve

The slope tool provides the slope of a line tangent to a curve at a single point.

Tip:
See Linear Fit to find the slope of a range of data.

1. Select the run you want to analyze in the graph legend. A red border indicates the selected run.



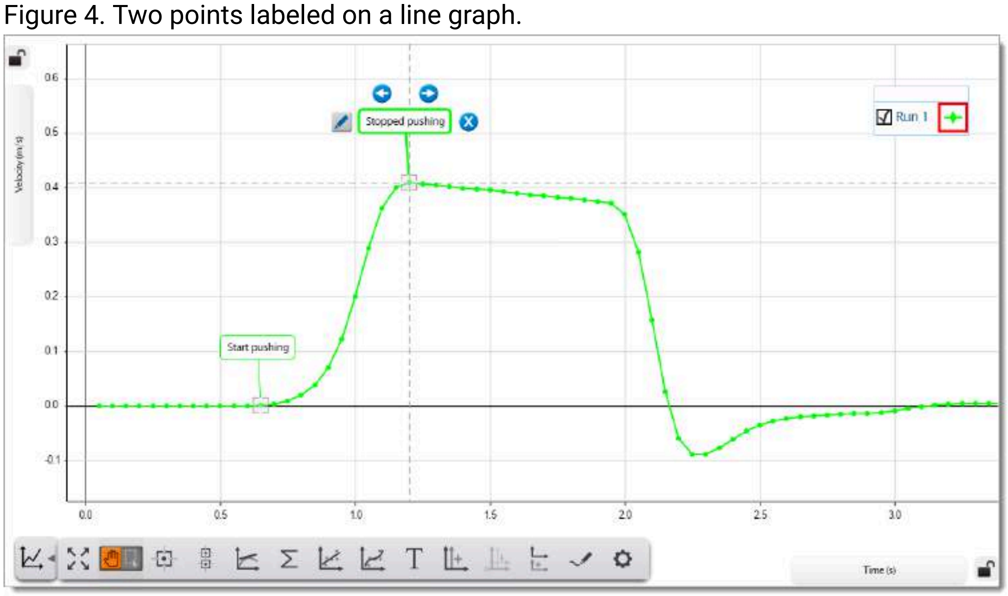
2. Click the point you want to analyze then click .
3. Place the tool on a different data point by clicking and dragging the tool or clicking  or .

To remove the tool, click the tool then click .

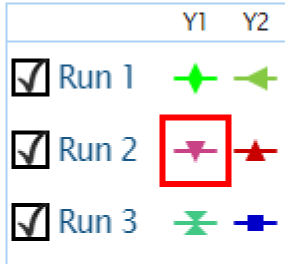
Label a data point in a line graph

Use the annotation tool to label a data point in a graph with a brief note.

Figure 4. Two points labeled on a line graph.



1. Select the run you want to analyze in the graph legend. A red border indicates the selected run.



2. Click the point you want to add a note to. You can move the tool by dragging the tool or clicking or .
3. Click to add a note.
4. Click the Enter Note field and type your note.

> **Tip:**
>
> Click Symbols or Greek Letters to enter special characters including subscripts and superscripts.

5. Click OK.

If needed, click to edit the note. To remove the note, click .

Draw a prediction on a line graph

Before collecting data, draw a prediction of the results directly on the graph.

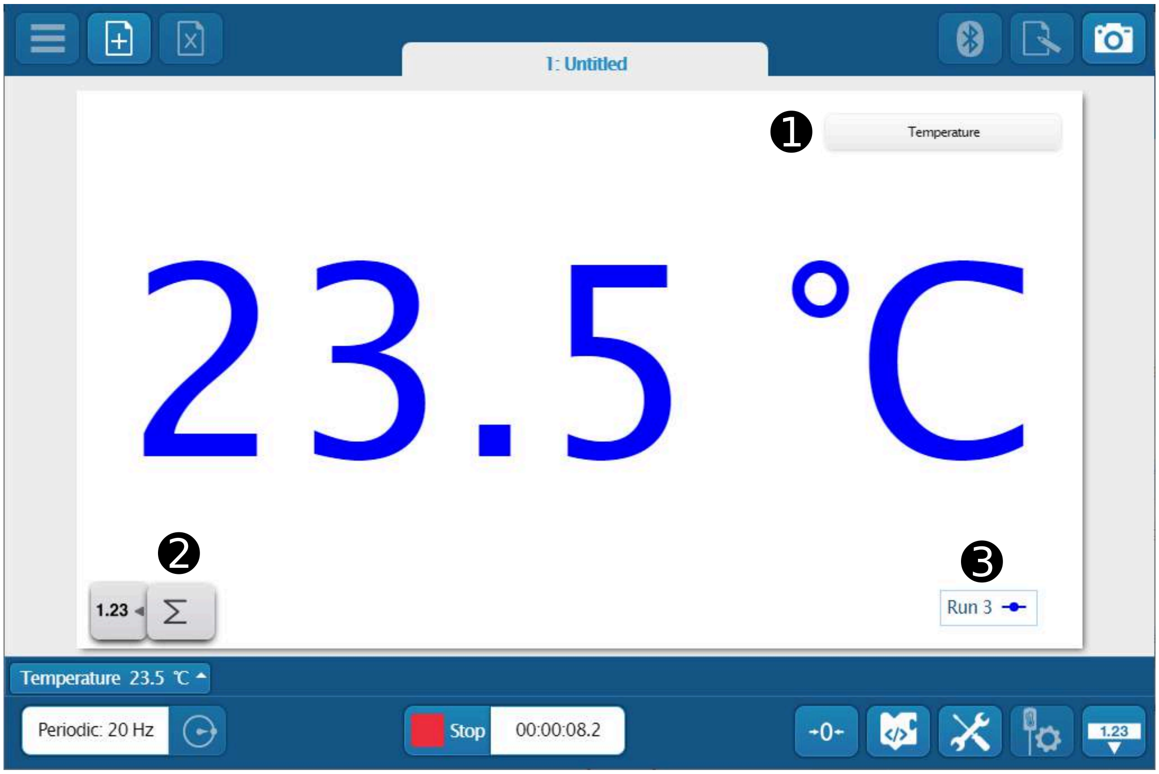
1. Go to then select .
2. Click and drag on the plot area to draw your prediction.
3. Click done when finished.

To remove the prediction from the graph, click to deselect the tool.

Digits Display

Show the most recently recorded data sample as digits, statistics of a single data run, or text output from a Blockly program.

Display Overview



1 **Measurement Selector**

Click to change the displayed measurement or units.

2 **Statistics Tool**

Click to display the minimum, maximum, mean, or standard deviation for the selected run.

3 **Run Selector**

Indicates the currently selected run. Click to select a new run to display.

Change the Displayed Measurement or Units

1. Click the currently selected measurement.
2. Change the measurement or units in the Measurement panel:
   - Click the measurement you want to display.

> **Tip:**
> Click the User-entered tab to access user-entered data, calculated data, or text-output from a Blockly program..

   - Click the unit of the current measurement then select the unit you want to display from the list.

Display Statistics for the Current Run

1. Click the tool drawer to open the tools then click Σ.
2. Select a statistic to display.
3. Click OK.

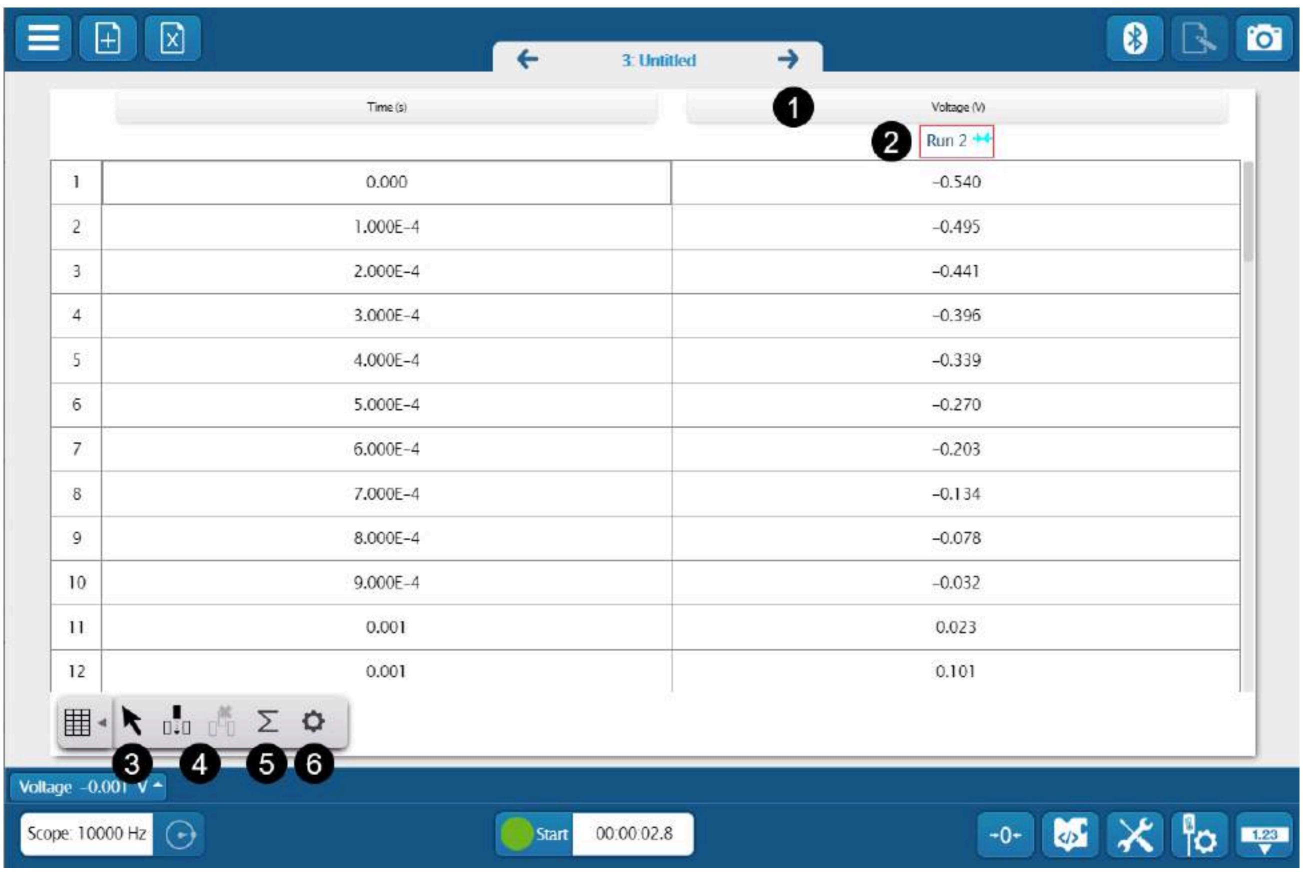
Display Text Output from a Blockly Program

1. Create a program using the Code tool that includes text output.
2. Click **Select Measurement**, select the **User-entered** tab, then select the name of the text output.

Table display

Display data using two or more columns and show data statistics.

Overview



1. **Measurement Selector**

   Click to change the displayed measurement or units.

2. **Run Selector**

   Click to change the displayed run.

3. **Selection Tool**

   Use this tool to select table cells when using the Statistics and Remove Column tools.

4. **Add or Remove Columns**

   Click to add a column to the table. To remove a column, use the select tool to select a column then click .

5. **Statistics**

   Use the statistics tool to display the minimum, maximum, mean, and standard deviation of the column of data.

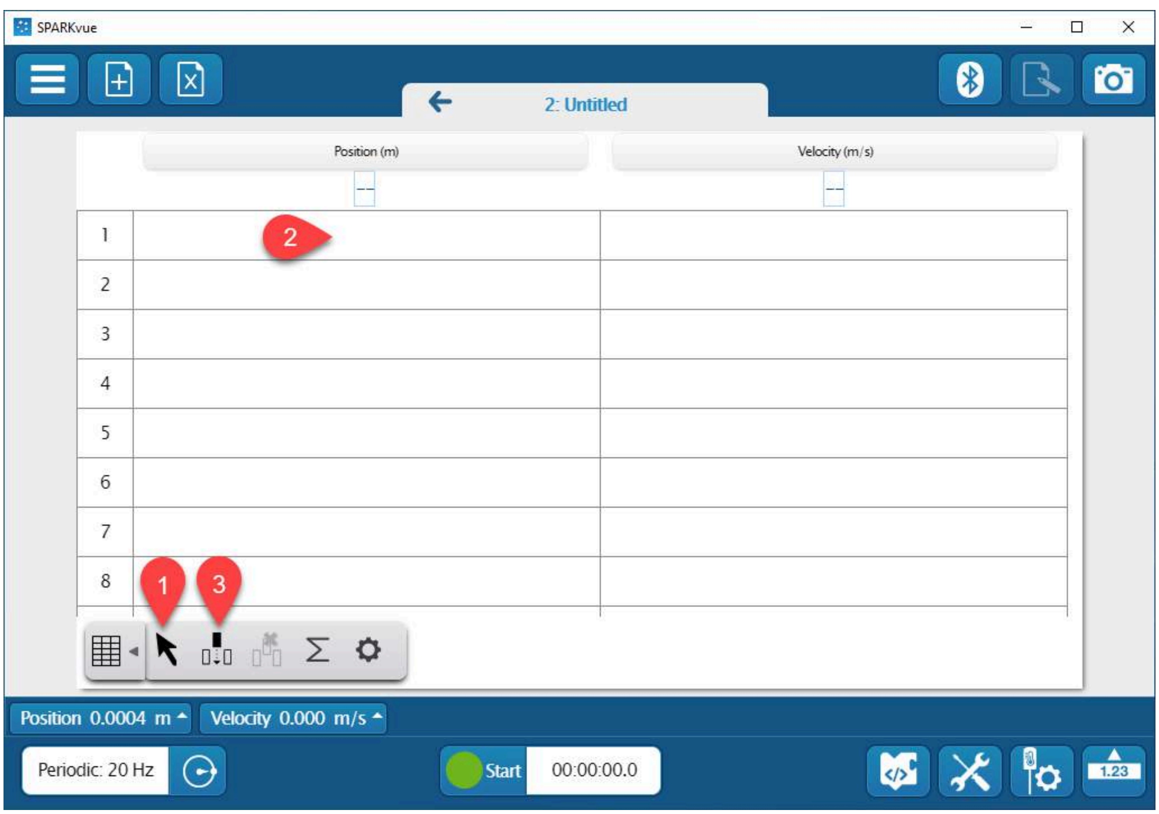
6. **Properties**

   Open the properties to toggle the visibility of the row numbers.

Add a column

Go to then click .

This adds a new column to the end of the table. To add a column to a different location in the table:

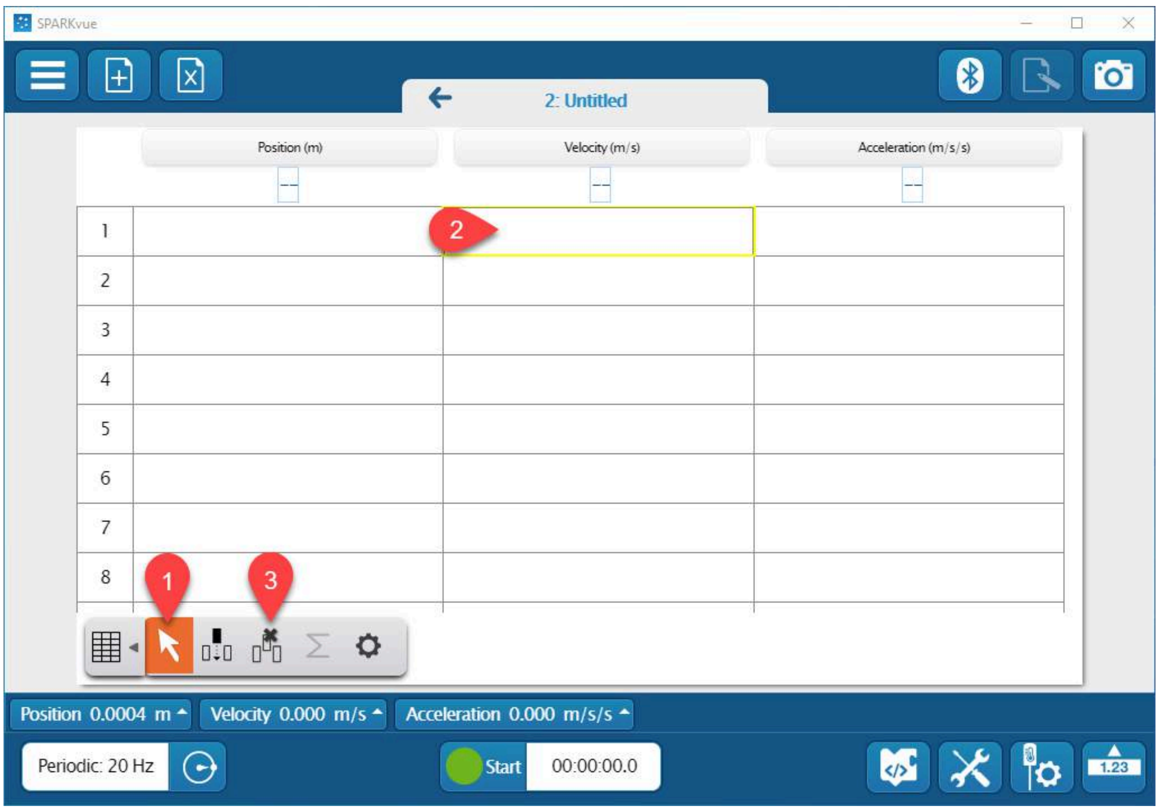


1. Go to then click .
2. Select a cell in the column to the right of where you want to insert a new column.
3. Click .

After you insert a new column, click Select Measurement to designate a measurement for that column.

Remove a column

> **Note:**
> Removing a column of data does *not* delete the data. If you want to delete data, see Delete Data Runs.



1. Go to then click .
2. Select a cell the column you want to remove.
3. Click .
4. Click OK to confirm.

Change the name and units of user-entered data

> **Note:**
> Changes made to the measurement name and units in the table carries over to other displays.

1. Click the pencil next to the name of the measurement you want to edit.
2. Enter a new name in the Measurement field.

> **Tip:**
>
> Click Symbols .?$ or Greek Letters ∝βγ to enter special characters including subscripts and superscripts.

3. Optional: Enter the units for the measurement in the Units field.
4. Click OK.

Show statistics of a measurement

Use the statistics tool to display the minimum, maximum, mean, and standard deviation of data and display it at the bottom of the column.

1. Go to then click Σ .
2. Select one or more statistics you want to display then click OK. The table displays the statistics at the bottom of each column.
3. Optional: To display the statistics for a range of measurements, click then click and drag over the table cells you want to analyze.

Meter Display

Show the most recently recorded data sample or a statistical value of a single run on an analog display.

1 **Measurement Selector**

Click to change the displayed measurement or units.

2 **Meter Display Tools**

- Click Σ to display the minimum, maximum, mean, or standard deviation for the selected run.
- Click to scale the meter to the best fit.
- Click **1.23** to display the measurement as digits below the meter.
- Click to change the visual appearance of the meter and manually set the minimum and maximum values on the scale.

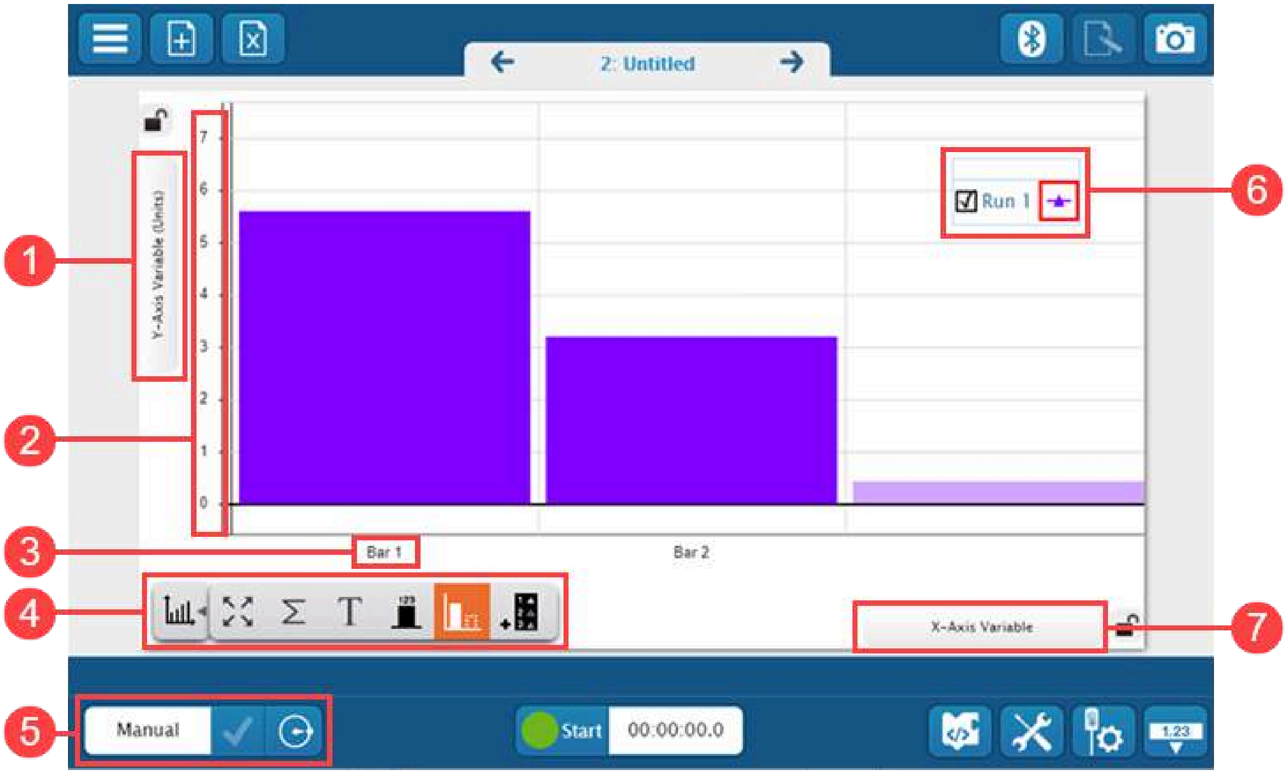
3 **Run Selector**

Indicates the currently selected run. Click to select a different run to display.

Bar Graph display

Compare manually-sampled measurements across categories.

Displaying data in a bar graph is useful when you want to compare a single sensor measurement between different categories, such as measuring the pH of several different chemicals. If you want to compare values across multiple sensors of the same type of measurement, a bar meter display would be more appropriate.



❶ **Y-Axis Variable**

Click to select a measurement to display on the y-axis. This is typically a sensor measurement.

❷ **Y-axis**

Drag to manually scale the y-axis.

❸ **Bar name**

Click the bar above the bar name to change the name.

4 **Bar graph tools**

Scale to fit

Click to scale the graph to fit all of the data.

Statistics Σ

Click to display statistics for the current run such as maximum, minimum, mean, and standard deviation.

Annotation T

Add a note to a bar. Click to turn on the tool then click a bar to add a note to.

Show digits

Click to show the digits of the measurement above each bar.

Show next bar

Click to show the next bar. Click this bar to add a new data point when displaying user-entered data on the Y-axis.

Add a run

Click to add a new run of data. This is only active when the Y-axis is displaying user-entered data.

5 **Manual entry**

Manual sampling is automatically selected when using a bar graph. After clicking **Start**, click the check to record a measurement.

6 **Legend**

Use the legend to identify, display, and select runs to analyze. Select the checkbox to show or hide runs. For analysis, the selected run is indicated by a red box.

7 **X-Axis Variable**

Click to select a measurement to display on the x-axis. This is typically user-entered data.

Collect data

1. Click **Y-Axis Variable** then select a measurement to display on the y-axis.
2. Click **X-Axis Variable**. The X-Axis variable is typically user-entered data, which means you need to create a User-entered measurement:

   a. Click the **User-entered** tab.
   b. Click **Create Data Set**.
   c. Enter a **Measurement Name** for the Data Set then click **OK**.

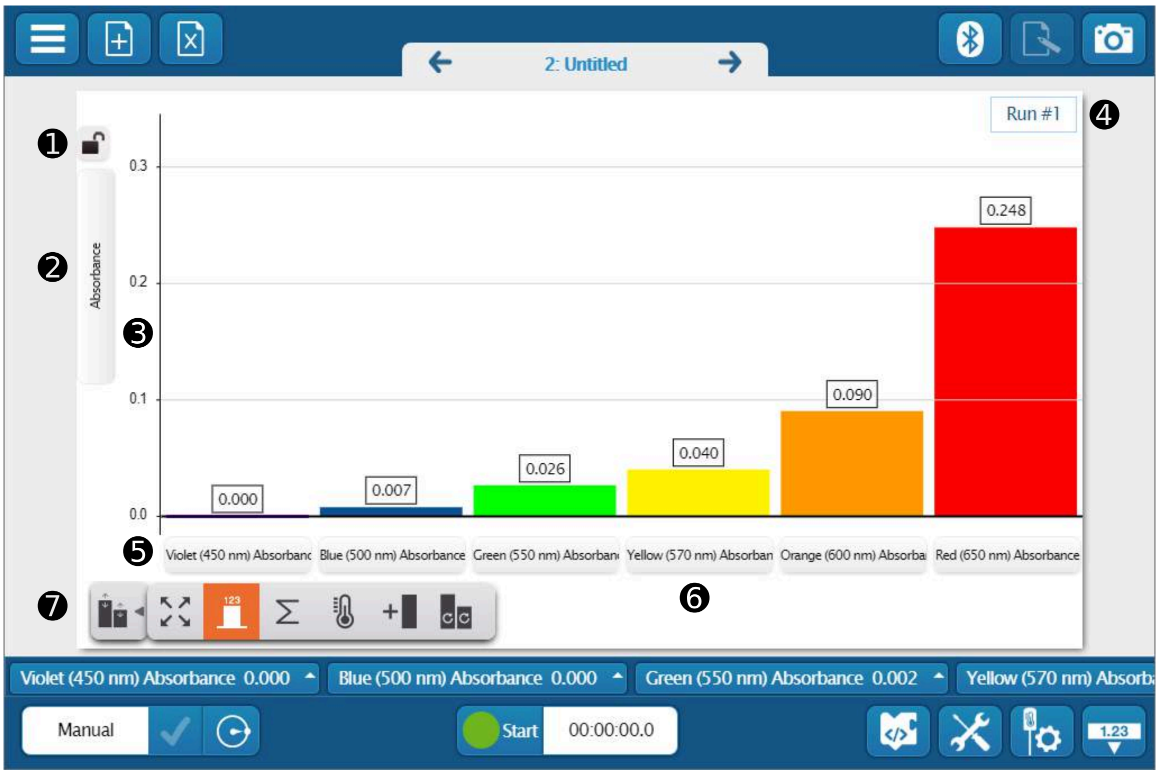
> **Note:**
> You can also configure the other Data Set options, but it's not required.

3. Click **Start** to begin data collection.
4. Take a measurement with your sensor. Once you are done with the measurement click the check to record the measurement. A second bar now appears.
5. Repeat the previous step for all of your data samples. Click **Stop** when finished.

Bar Meter Display

Visually compare the most recently recorded data values across similar measurements.

Displaying data in a bar meter is useful when you want to compare values across multiple sensors of the same type of measurement. The bar meter display is commonly used with the Wireless Colorimeter since it contains six absorbance sensors. If you want to compare a single sensor measurement across multiple categories, a bar graph display would be more appropriate.



1 **Axis Lock**

Click to lock or unlock the axis scaling.

2 **Common Measurement**

Indicates the type of measurement common to all of the measurements on the horizontal axis. Click to change the displayed unit, if available.

3 **Axis**

Click and drag to scale the axis.

4 **Run Selector**

Click to change the displayed data run.

5 **Measurement Selector**

Click to change the displayed measurement.

> **Note:**
> It's only possible to select a measurement that's of the same type as the vertical axis.

6 **Bar Editor**

Click under a measurement to show the Bar Editor tools. Click to remove the bar. Click the left or right arrows to move the bar.

7 **Bar Meter Tools**

Scale to Fit

Automatically scale the entire display to make all measurements visible.

Digits

Show or hide digital values on top of each bar.

Statistics

Display the minimum, maximum, mean, or standard deviation below each measurement.

Thermometer Mode

Display the data as thermometers instead of bars.

Add Bar

Click to add an additional measurement at the far right of the display.

Clear Measurements

Click to reset all the measurements to select new measurements to display.

> **Note:**
> This tool does *not* delete the data. If you wish to delete the data, see Delete Runs.

Map Display

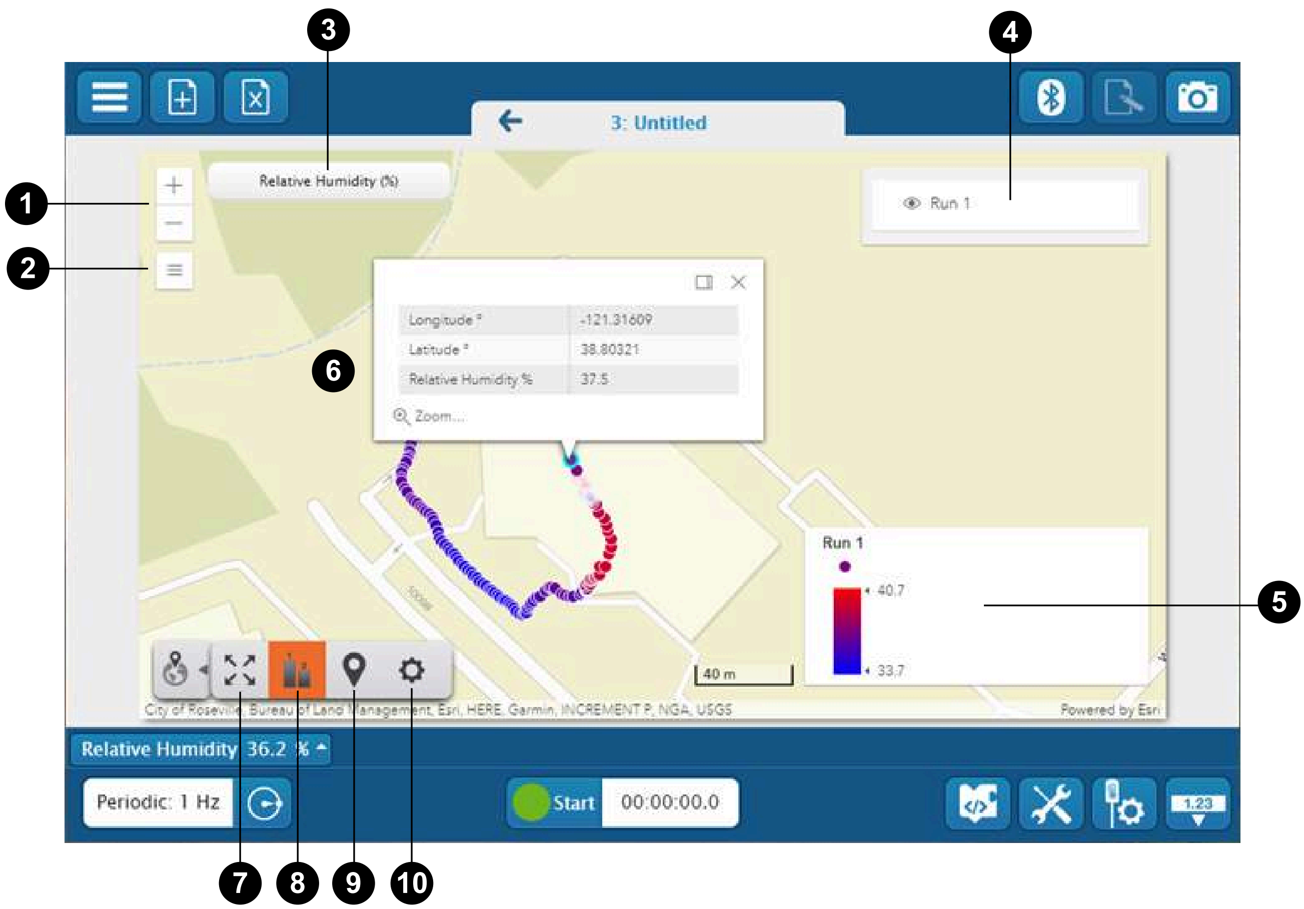
Tag sensor measurements with GPS data and display the measurement locations on a map.

> **Note:**
> The Map display requires an internet connection.

Use the GPS sensor on one of the following devices to tag GPS data with sensor measurements:

• Wireless Weather Sensor
• On-board GPS Sensor on iOS and Android devices
• On-board GPS Sensor on SPARK LXi and SPARK LX Data loggers

Overview of the display



1 **Zoom**

Click + to zoom in or − to zoom out the map.

2 **Map type**

Click to select a variety of map types including satellite, streets, and topographic.

3 **Measurement selector**

Click to select or change the displayed measurement and units.

4 **Run selector**

Click a run to show or hide a run on the map.

5 **Legend**

Shows the displayed range for each run and how the data point colors correspond to the value of the measurement.

6 **Map**

- Move the map by dragging it.
- Zoom in and out on the map by using a mouse scroll wheel or pinching the screen.
- Show the measurement values of a data point by clicking it.

7 **Scale to fit**

Click to automatically scale the map to show all data.

8 **Legend**

Click to show or hide the legend.

9 **Location**

Click to center the map at your current location. This only works if connected to a GPS sensor.

10 **Properties**

Use to change the units between metric and US customary.

Set up the On-board GPS Sensor

SPARKvue can use the GPS sensor on iOS and Android devices to tag sensor measurements with GPS data. Enable the GPS sensor before collecting data.

1. Select **Sensor Data** on the Welcome Screen.
2. Connect a sensor that you want to tag with GPS data.
3. Toggle the **On-board GPS Sensor** switch to enabled.
4. Select any **Template**.
5. Build an experiment page with a Map display. Select the full-page layout.
6. Click **Start** to begin data collection.

> **Note:**
> Latitude and longitude measure 0 if the device doesn't detect any GPS satellites. Set up data collection outside to get the best GPS signal.

Export GPS Data

You can export GPS data as a GPX or KML file to open in another program, such as Google Earth.

> **Note:**
> Exporting GPS data is not supported on iOS or Android devices.

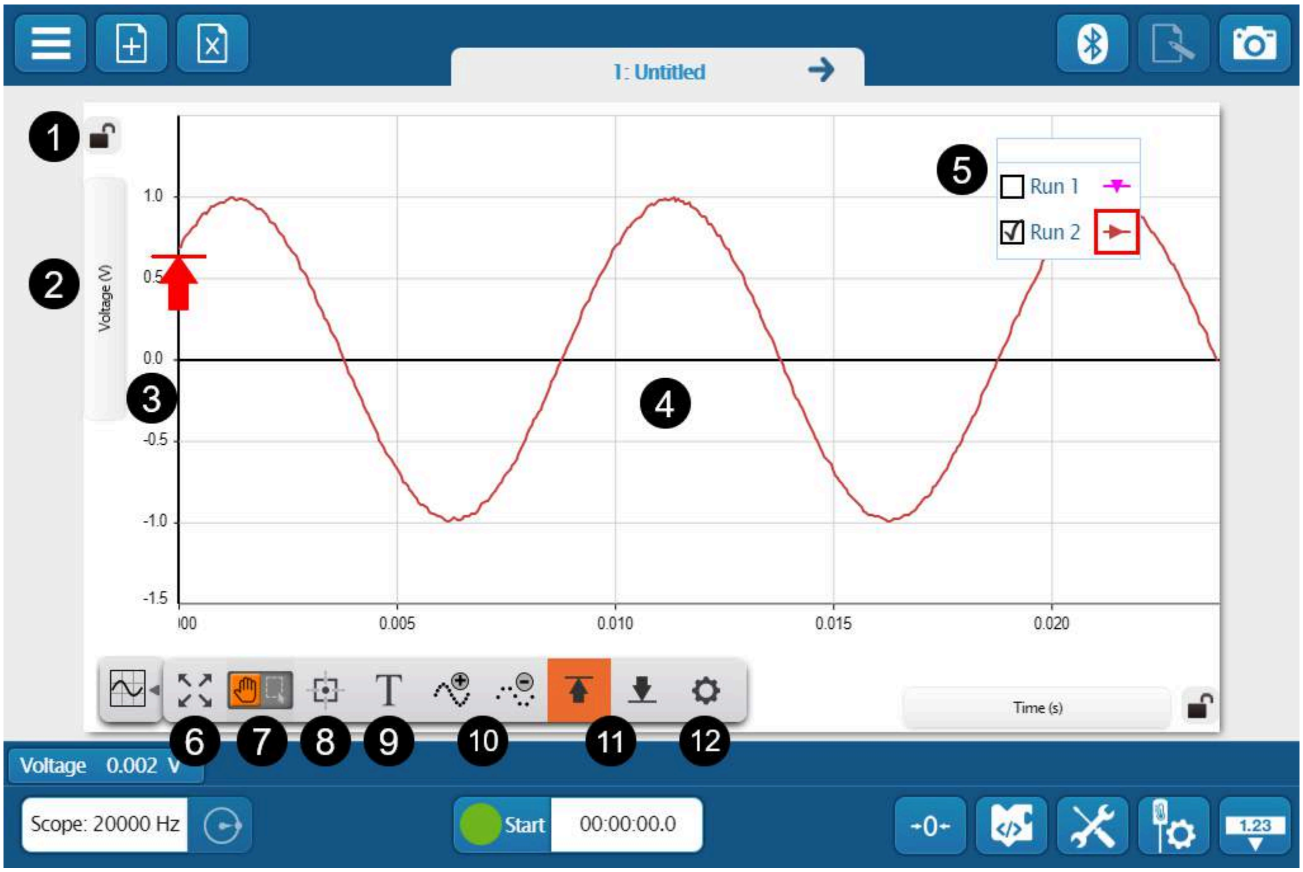
1. Click the **Main menu** then select **Export Data**.
2. Select either KML or GPX for the file format.
3. Select a location on your computer to save the file.
4. Click **Save**.

Scope display

Create a display that functions as a digital oscilloscope. This display is useful for viewing measurements that oscillate rapidly in time such as electrical signals and sound waves.

Note:
When building a new page, connect a supported sensor to show the Scope as a selectable display option.

Overview



1 **Axis Lock**

Click to lock or unlock axis scaling.

2 **Measurement Selector**

Click to change the displayed measurement or units.

3 **Axis**

Click and drag to scale each axis individually.

4 **Plot Area**

Click and drag to move the plot area.

5 **Legend**

Select a run to analyze by clicking the run in the legend. Right-click the run symbol to change the run color and other data set properties.

6 **Scale to Fit**

Change the scale of a line graph to zoom in on a range of data or to view an entire run of data.

7 **Cursor Mode**

Click to toggle the function of the mouse cursor.

- Switch to Move Mode to use the cursor to move the plot area.
- Switch to Select Mode to highlight an area on the plot for data analysis.

8 **Coordinates Tool**

Use to determine the coordinates of a single data point. Use the delta tool to determine the difference between two points of data.

9 **Annotation Tool**

Use the annotation tool to label a data point in a graph with a brief note.

10 **Data Point Tools**

Use these tools to increase or decrease the number of data points in a trace. Increasing the number of data points makes a smoother trace but decreases the displayed range of the trace.

11 **Trigger Tools**

Use the scope trigger to get a stable trace. Click to use a positive edge trigger or click to use a negative edge trigger .

12 **Properties**

Change the visibility of data points, symbols, and lines connecting data points in a line graph.

Get a stable trace

Use the trigger to get a stable trace on the scope display.

1. Click **Start** .
2. Click or to activate the scope trigger. The trigger icon appears on the y-axis of the scope display.

> **Tip:**
> A positive edge trigger displays the trace as the signal is rising at the trigger point. A negative edge trigger displays a trace as the signal is falling at the trigger point.

3. Drag the trigger along the y-axis to the value that the signal must pass to start measuring.

Supported Sensors

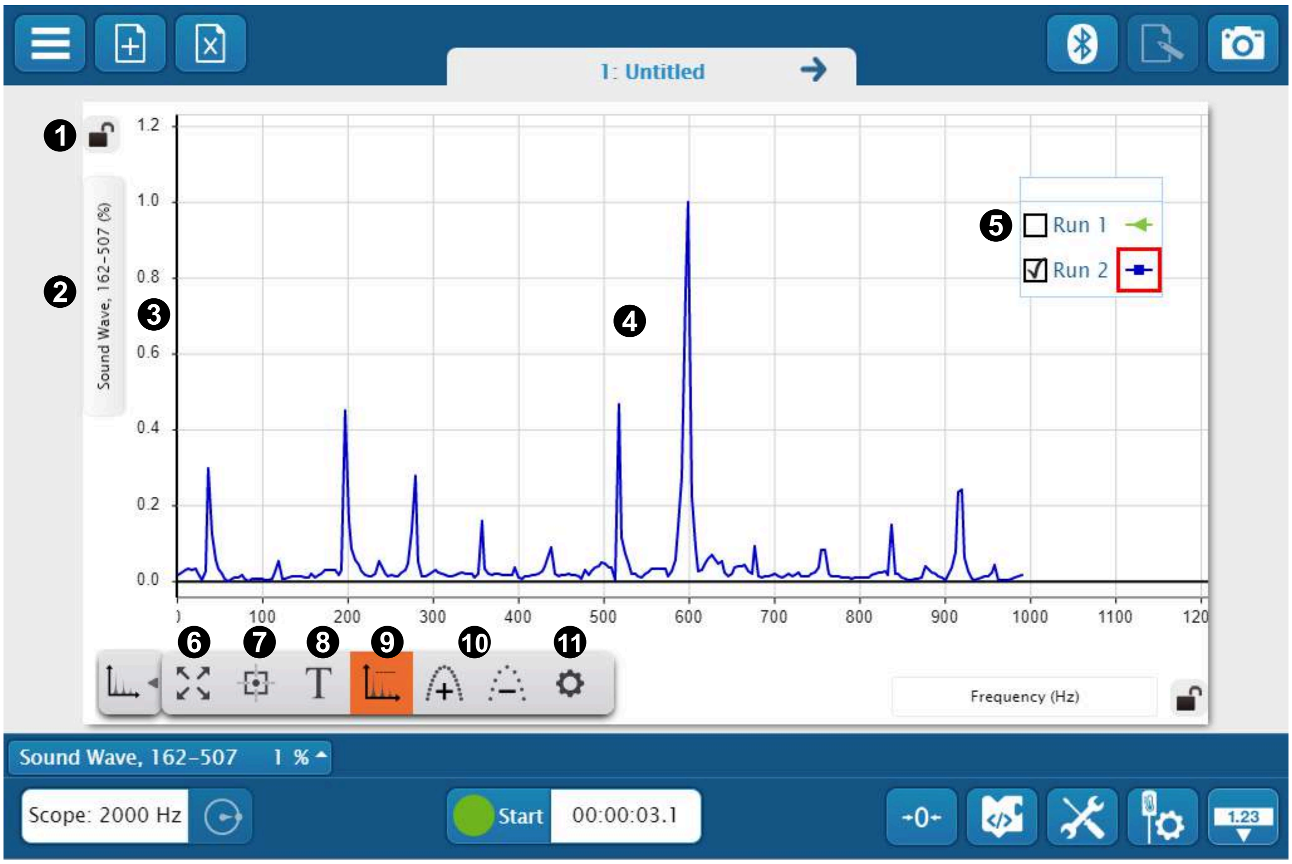
You can use the Scope display with the following devices:

- 550 Universal Interface
- On-board Microphone
- Wireless AC/DC Module
- Wireless Current Sensor
- Wireless Current Sensor Module
- Wireless Sound Sensor
- Wireless Voltage Sensor

FFT Display

Create a spectral analysis display that uses a Fast Fourier Transform (FFT) to measure the relative signal intensity as a function of frequency.

Note:
Connect a supported sensor to make the display visible when building an experiment page.



1 **Axis Lock**

Click to lock or unlock axis scaling.

2 **Measurement Selector**

Click to change the displayed measurement or units.

3 **Axis**

Click and drag to scale each axis individually.

4 **Plot Area**

Click and drag to move the plot area.

5 **Legend**

Select a run to analyze by clicking the run in the legend. Right-click the run symbol to change the run color and other data set properties.

6 **Scale to Fit**

Change the scale of a line graph to zoom in on a range of data or to view an entire run of data.

7 **Coordinates Tool**

Use to determine the coordinates of a single data point. Use the delta tool to determine the difference between two points of data.

8 **Annotation Tool**

Use the annotation tool to label a data point in a graph with a brief note.

9 **Normalize Data**

Select to change the vertical scale to make 1 the maximum value.

10 **Change Sample Rate**

Click these tools to increase or decrease the sample rate.

> **Tip:**
> Set a sample rate that's approximately twice frequency range displayed on the FFT.

11 **Properties**

Change the visibility of data points, symbols, and lines connecting data points in a line graph.

Supported Sensors

You can use the FFT display with the following devices:

- 550 Universal Interface
- On-board Microphone
- Wireless AC/DC Module
- Wireless Current Sensor

- Wireless Current Sensor Module
- Wireless Sound Sensor
- Wireless Voltage Sensor

Configure sensors and hardware

Connect sensors to SPARKvue and control hardware such as Smart Cart accessories.

Connect photogates and other timing devices

Learn about the steps to create a timer, how to determine which timer to select, and how the timer records data.

Timer setup steps

1. Open SPARKvue or click then select **Start New Experiment**.
2. Click **Sensor Data**:

3. Connect a Wireless Smart Gate (PS-3225), Smart Gate (PS-2180), or Photogate Heads (ME-9498A) to SPARKvue.
4. Click configure sensor .
5. Select a timer from the list then click **OK**.

> **Tip:**
> See the Which timer to use section to help decide which timer to use for your experiment.

6. Enter the **Timer Properties** for your experiment setup then click **OK**.

> **Tip:**
> Click the link for the timer in the Which timer to use section for instructions on how to measure the parameters.

7. Select the measurements you want to display.
8. Select a template to display the selected measurements.

Which timer to use

Use the table to help determine which timer to select based on your experiment. Click the links for the timer to get more information on how to set up the timing devices and measure the parameters.

Experiment	One Photogate	Two Photogates	Picket Fence	Smart Pulley	Collision	Pendulum	Time of Flight
Cart position			X	X			

Experiment	One Photogate	Two Photogates	Picket Fence	Smart Pulley	Collision	Pendulum	Time of Flight
Cart speed	X	X	X	X			
Cart acceleration		X	X	X			
Cart collisions (velocity)					X		
Pendulum motion						X	
Rotation motion				X			
Projectile motion							X
Free fall acceleration			X				

How data is recorded

Recording data with timers uses a different method than recording data with sensors. While sensors collect data continuously at a given sample rate, photogates record data at specific events in time, as shown in the following example.

	Time (s)	Position (m)	Velocity (m/s)	acceleration (m/s/s)
		Run 1	Run 1	Run 1
1	4.951	0.00		
2	4.973		1.11	
3	4.993			9.70
4	4.996	0.05		
5	5.013		1.49	
6	5.028			9.84
7	5.029	0.10		
8	5.043		1.79	

Since data is recorded when a timing device detects an event, the first recorded data point usually doesn't occur at zero seconds on the clock. Additionally, data isn't aligned across rows since events are recorded at different points in time. This isn't a problem when displaying the data on a graph or digits display, but can look odd when displaying data in a table.

Set up a pendulum timer

Use this timer to measure the period and speed of a pendulum as it passes through a photogate.

Equipment setup

- **Photogate**

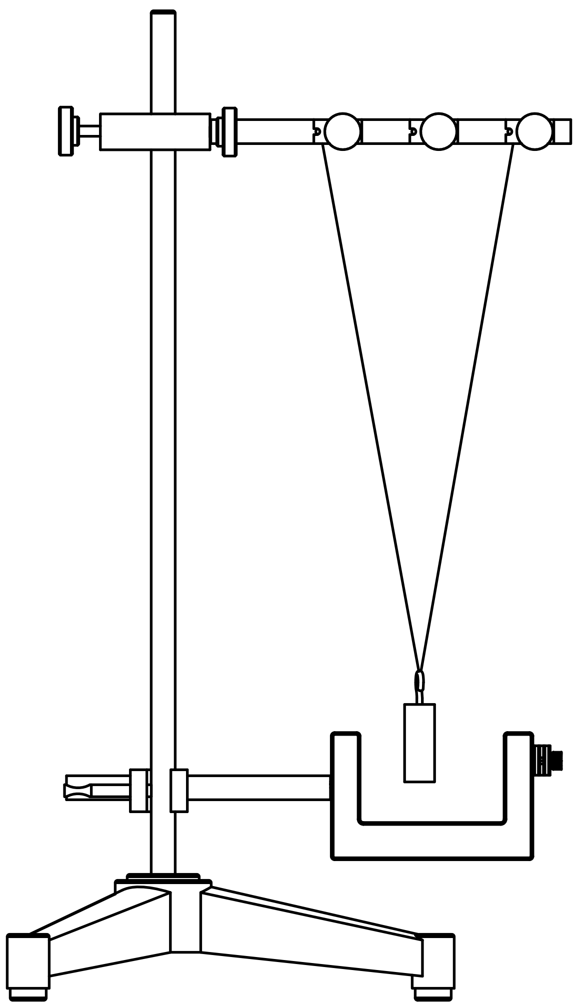
  Use a Wireless Smart Gate (PS-3225), a Smart Gate (PS-2180) with a PASPort interface, or a Photogate Head (ME-9498A) with a digital interface.

- **Pendulum**

  Use a cylinder since it has a constant diameter, such as one from the Photogate Pendulum Set (ME-8452).

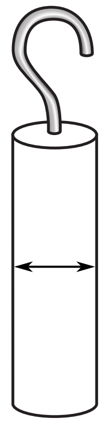
- **Ring Stand with Pendulum Clamp (ME-9506)**

  Set up the pendulum and photogate on a ring stand using a setup such as the example shown here.



Set up the timer

1. Connect the photogate to SPARKvue.
2. Click configure sensor .
3. Click **One Photogate or Device** or **Smart Gate Only**, select **Photogate and Pendulum**, then click **OK**.
4. Enter the **Pendulum Width**. This is equal to the diameter of the cylinder. Click **OK** when finished.



5. Select the measurements you want to display.
6. Select a template to display the selected measurements.

Record data

Click Start then release the pendulum. The software will record the period after the pendulum passes through the photogate three times. The software records the speed each time the pendulum passes through the photogate.

Timing begins when the pendulum first passes through the photogate. The second pass is ignored since it corresponds to the pendulum swinging back in the opposite direction. Timing stops at the third pass since the pendulum completes one full oscillation.

Set up a pulley timer

Use the pulley timer linear measurements to measure the position, speed, and acceleration of a cart on a track or a mass on an Atwood's machine. Use the rotational measurements to measure the angle, angular speed, and angular acceleration of a rotating object.

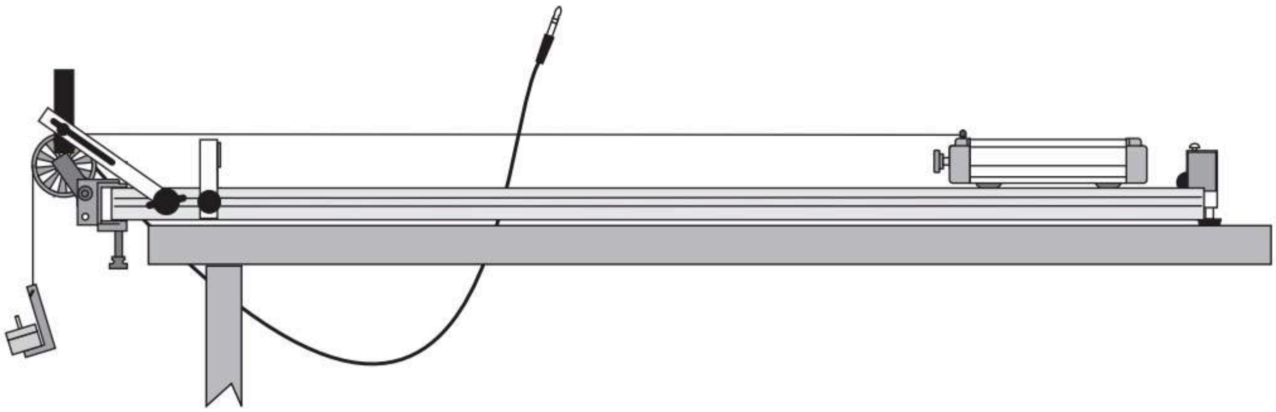
Equipment setup

- **Photogate**

  Use a Wireless Smart Gate (PS-3225), a Smart Gate (PS-2180) with a PASPort interface, or a Photogate Head (ME-9498A) with a digital interface.

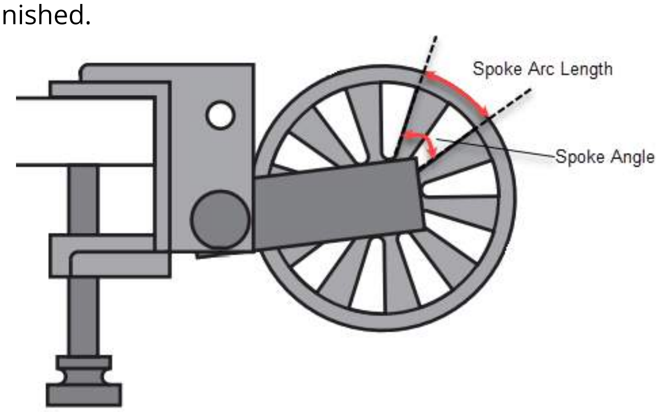
- **Super Pulley (ME-9433)**

  Clamp the Super Pulley to the end of a track, to a ring stand, or a rotational apparatus.



Set up the timer

1. Connect the photogate to SPARKvue.
2. Click configure sensor .
3. Click **One Photogate or Device** or **Smart Gate Only**, select **Smart Pulley (Linear)**, then click **OK**.
4. Enter the **Spoke Arc Length**. This is equal to the diameter of the pulley's inner grove divided by the number of spokes. Click **OK** when finished.



5. Select the measurements you want to display.
6. Select a template to display the selected measurements.

Record data

Click Start then spin the pulley. The software records a measurement each time the pulley blocks then unblocks the photogate beam. Speed requires two initial passes and acceleration required three initial passes before a the first measurement is recorded.

Set up a picket fence timer

Use this timer to measure position, speed, and acceleration of a cart that has a picket fence attached. Also measure the acceleration due to gravity when dropping the picket fence through a photogate.

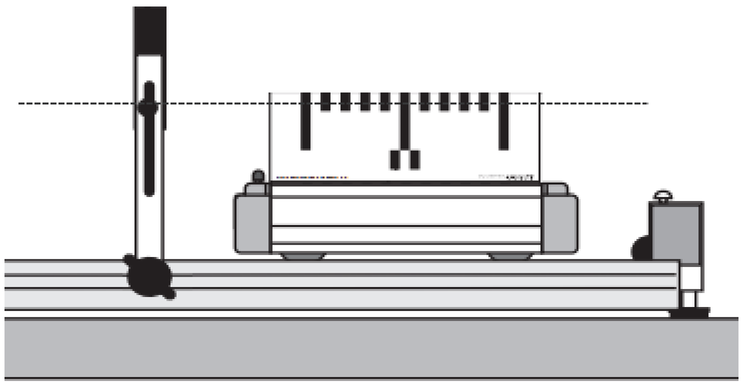
Equipment setup

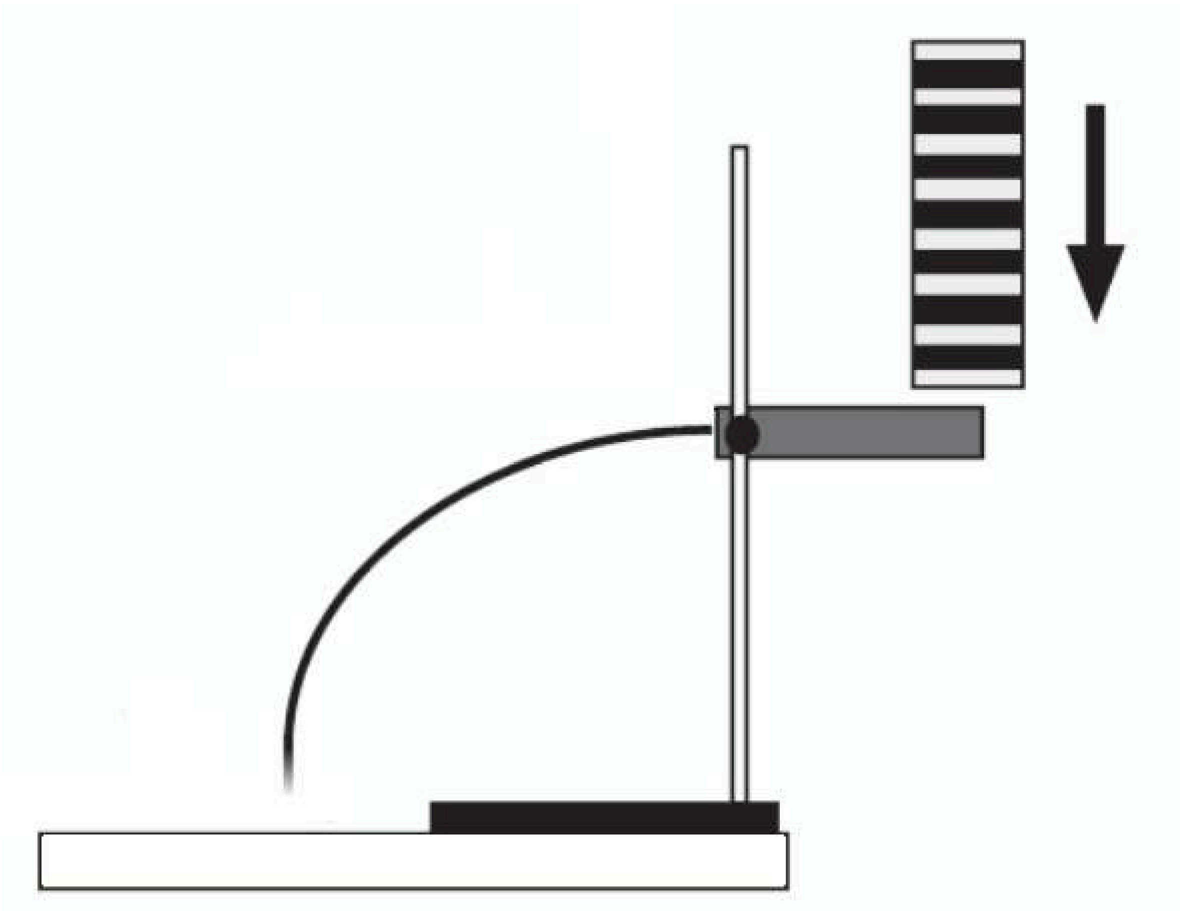
- **Photogate**

  Use a Wireless Smart Gate (PS-3225), a Smart Gate (PS-2180) with a PASPort interface, or a Photogate Head (ME-9498A) with a digital interface. Mount the photogate on a track if measuring the motion of a cart. Mount the photogate to a ring stand if measuring the acceleration due to gravity (as shown below).

- **Picket Fence**

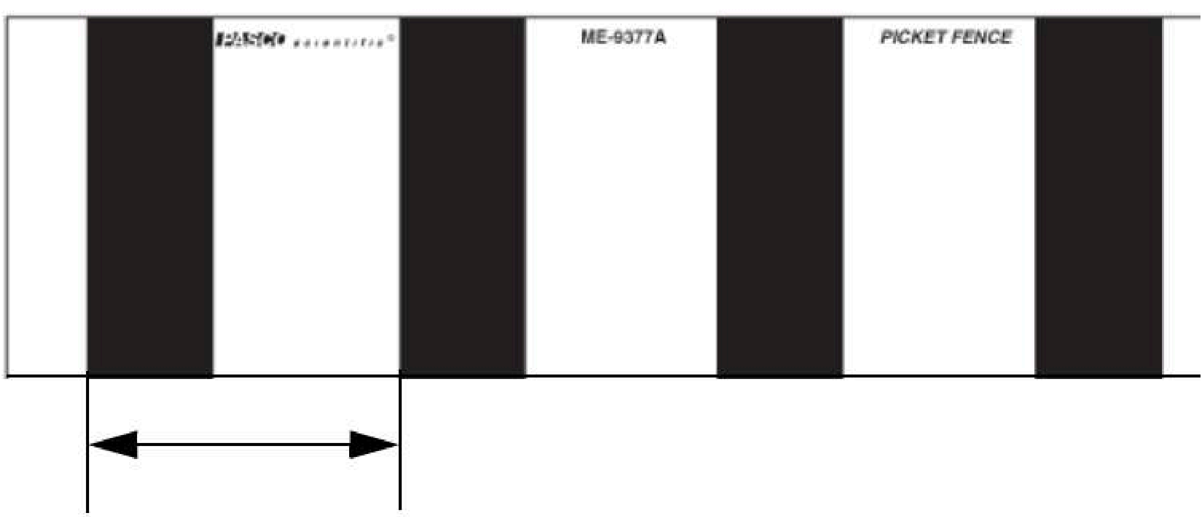
  Use either the Smart Timer Picket Fence (ME-8933) or Cart Picket Fence (ME-9804) if measuring the motion of a cart. Use the Picket Fence (ME-9377) if measuring the acceleration due to gravity.





Set up the timer

1. Connect the photogate to SPARKvue.
2. Click configure sensor .
3. Click **One Photogate or Device** or **Smart Gate Only**, select **Photogate and Picket Fence**, then click **OK**.
4. Enter the **Flag Spacing**. This is equal to the distance between the leading edges of two flags. Click **OK** when finished.



5. Select ☑ the measurements you want to display.
6. Select a template to display the selected measurements.

Record data

Click Start ● then let the entire picket fence pass through the photogate. The first time the photogate is blocked it records a position of zero. The second time the photogate is blocked it records the second position and the speed. The third time the photogate is blocked a third position, a second speed measurement, and the acceleration is recorded.

Set up a timer using one photogate

Use this timer to measure the speed of a cart.

Equipment setup

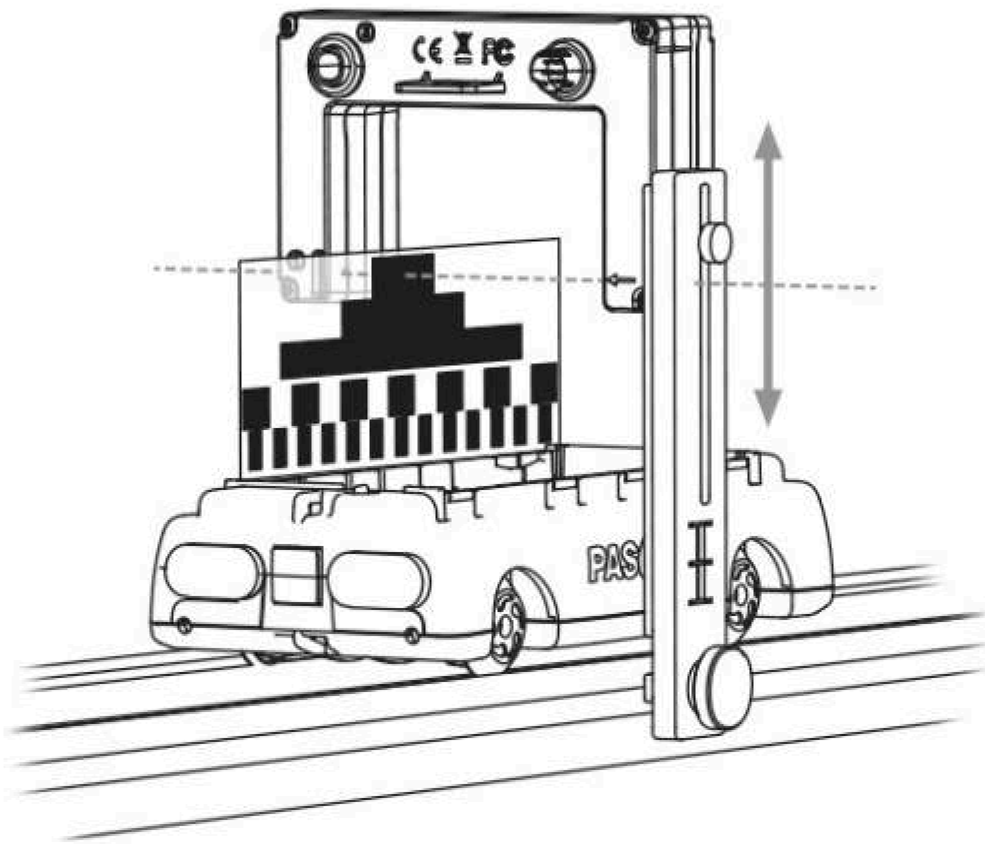
- **Photogate with Photogate Bracket (ME-9806)**

  Use a Wireless Smart Gate (PS-3225), a Smart Gate (PS-2180) with a PASPort interface, or a Photogate Head (ME-9498A) with a digital interface.

- **Cart Picket Fence (ME-9804)**

  The Cart Picket Fence has three flags of different length. Adjust the height of the photogate so that the correct flag passes through the photogate beam as shown below.

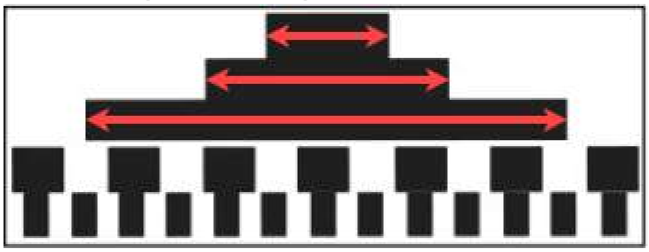
- **Cart and track**



Set up the timer

1. Connect the photogate to SPARKvue.
2. Click configure sensor .
3. Click **One Photogate or Device** or **Smart Gate Only**, select **Photogate Timing (One Photogate)**, then click **OK**.

4. Enter the **Flag Length**. Measure the length of the flag from end-to-end. Click **OK** when finished.



5. Select  the measurements you want to display.
6. Select a template to display the selected measurements.

Record data

Click Start  then allow the flag to pass through the photogate. The software records the speed after the flag blocks then unblocks the photogate beam.

Set up a timer using two photogates

Use this timer to measure the speed and acceleration of a cart.

Equipment setup

- **2×Photogates**

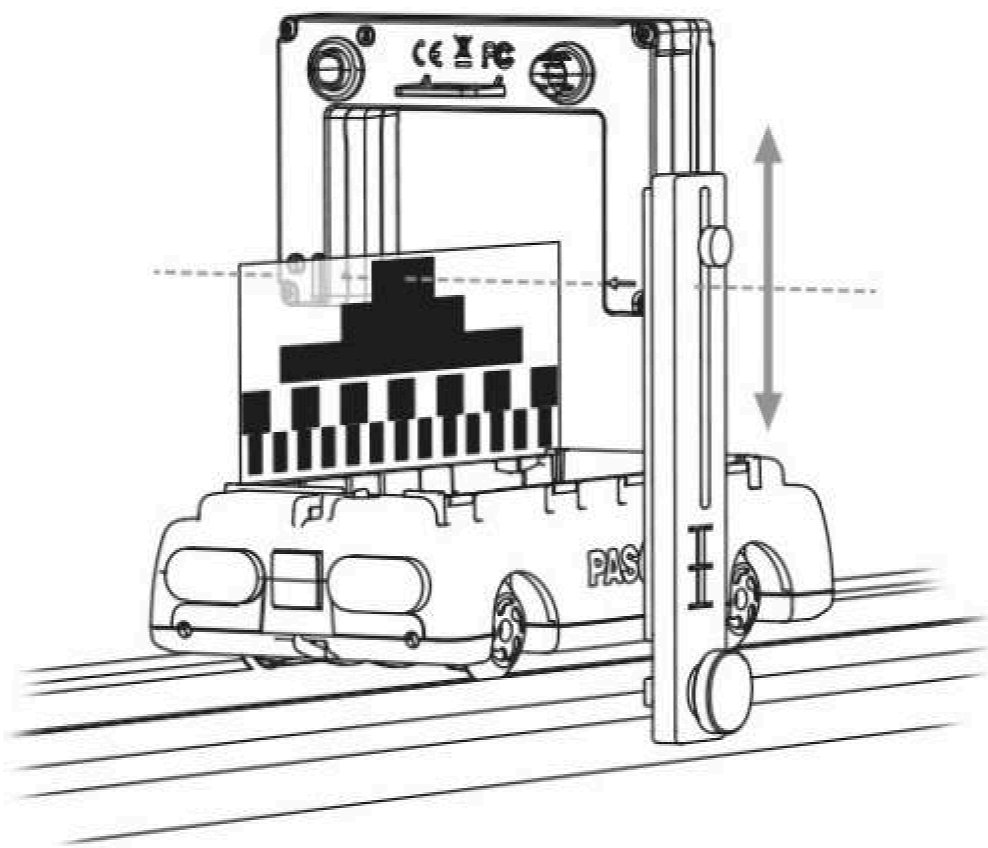
  Use one of the following setups:

  ◦ Wireless Smart Gate (PS-3225) with a Photogate Head (ME-9498A)
  ◦ Smart Gate (PS-2180) with a Photogate Head (ME-9498A) connected to a PASPort interface
  ◦ Two Photogate Heads (ME-9498A) connected to a digital interface

- **Cart Picket Fence (ME-9804)**

  The Cart Picket Fence has three flags of different length. Adjust the height of the photogate so that the correct flag passes through the photogate beam as shown below.

- **Cart and track**

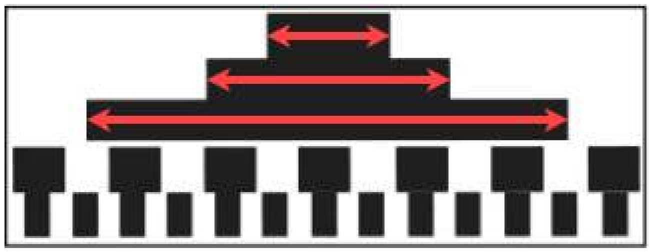


Set up the timer

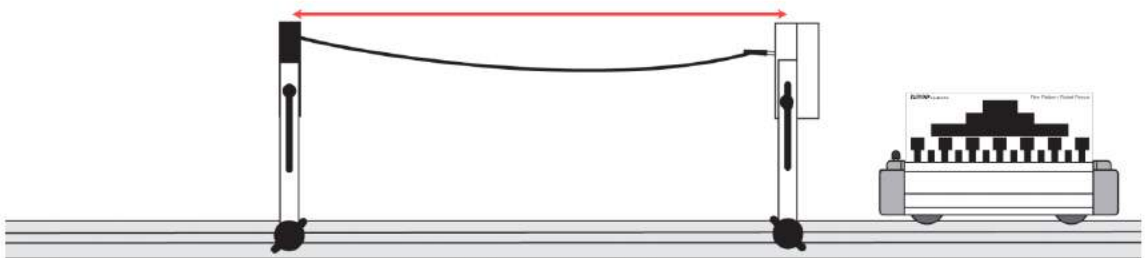
1. Connect the photogate to SPARKvue.
2. Click configure sensor .
3. Click **Two Photogates** or **Smart Gate and Auxiliary Port**, select **Photogate Timing (Two Photogates)**, then click **OK**.

4. Enter the parameters:

    - **Flag Length**: Measure the length of the flag from end-to-end.



    - **Photogate Spacing**: This is equal to the distance between the photogate beams.



5. Select ☑ the measurements you want to display.
6. Select a template to display the selected measurements.

Record data

Click Start then allow the flag to pass through the photogate. The software records the speed after the photogate is blocked then unblocked. A second speed measurement and the acceleration is recorded when the flag passes through the second photogate.

Set up a collision timer

Use this timer to measure the speed of two carts before and after a collision.

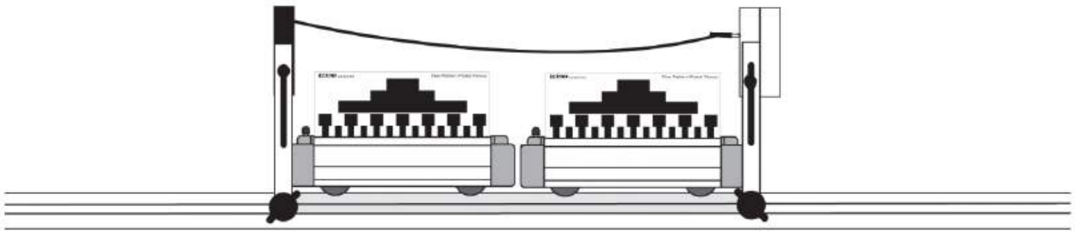
Equipment setup

- **2×Photogates**

  Use one of the following setups:

  ◦ Wireless Smart Gate (PS-3225) with a Photogate Head (ME-9498A)
  ◦ Smart Gate (PS-2180) with a Photogate Head (ME-9498A) connected to a PASPort interface
  ◦ Two Photogate Heads (ME-9498A) connected to a digital interface

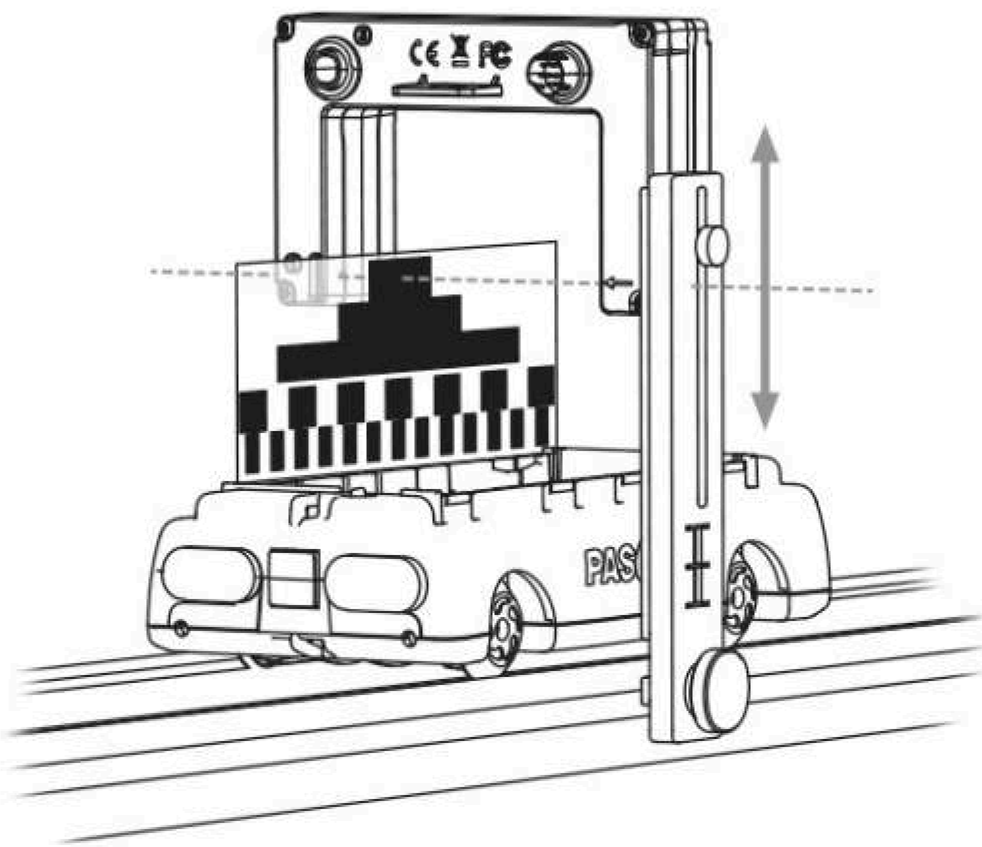
  The following image shows a setup using a Smart Gate with a connected Photogate Head.



- **2×Cart Picket Fence (ME-9804)**

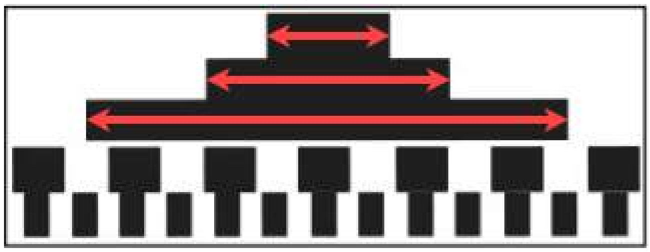
  The Cart Picket Fence has three flags of different length. Adjust the height of the photogate so that the correct flag passes through the photogate beam as shown below.

- **2×Carts and track**



Set up the timer

1. Connect the photogate to SPARKvue.
2. Click configure sensor .
3. Click **Two Photogates** or **Smart Gate and Auxiliary Port**, select **Collision Timer**, then click **OK**.
4. Enter the **Flag Length**. Measure the length of the flag from end-to-end.



5. Select  the measurements you want to display.
6. Select a template to display the selected measurements.

Record data

Click Start  then allow the flag to pass through the photogate. The timer keeps track of the speeds through each photogate individually. It measures the speed before the collision when it first goes through the photogate and the speed after collision when it returns back through the photogate.

Set up a time of flight timer

Use this timer to measure the time it takes for a launched projectile to reach the floor and the initial speed of the projectile.

Equipment setup

- **2×Photogates**

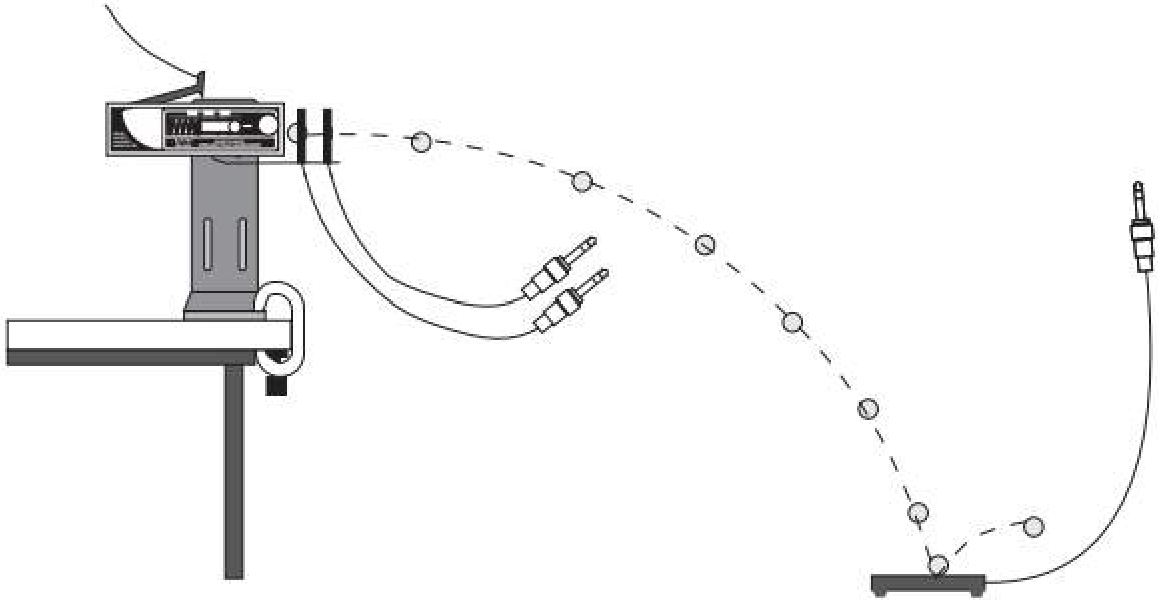
  Use a Wireless Smart Gate (PS-3225), a Smart Gate (PS-2180) with a PASPort interface, or two Photogate Heads (ME-9498A) with a digital interface.

- **Time-of-Flight Accessory (ME-6810A)**

  Connect the accessory to the auxiliary port on the Smart Gate or to a digital interface. Place the accessory on the floor as shown in the diagram below.

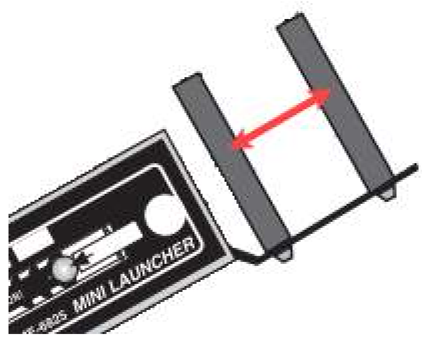
- **Projectile Launcher with Photogate Mounting Bracket (ME-6821A)**

  Use either the Projectile Launcher (ME-6800) or Mini Launcher (ME-6825) with a Photogate Mounting Bracket attached. Attach the photogates to the bracket as shown in the diagram.



Set up the timer

1. Connect the photogate to SPARKvue.
2. Click configure sensor .
3. Click **Two Photogates and Device** or **Smart Gate and Auxiliary Port**, select **Time of Flight**, then click **OK**.
4. Enter the **Photogate Spacing**. This is equal to the distance between the photogate beams. SPARKvue enters the value automatically when using a Smart Gate.



5. Select the measurements you want to display.
6. Select a template to display the selected measurements.

Record data

Click Start then launch the projectile. The timer starts when the projectile goes through the photogates and stops when it hits the Time-of-Flight Accessory.

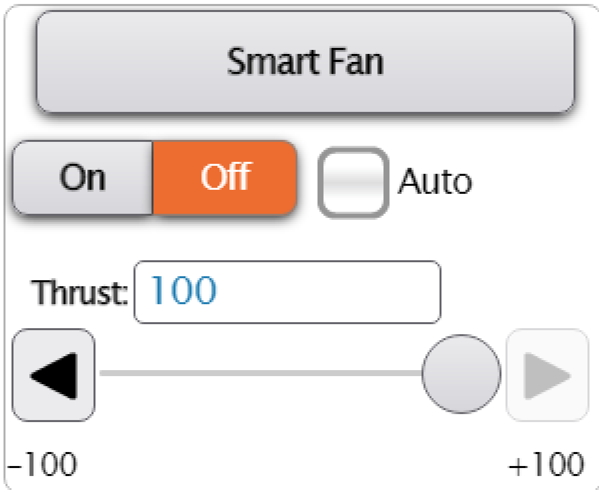
Smart Cart accessories

Add a display to SPARKvue to control a Smart Cart accessory.

Control the Smart Fan Accessory

Control the thrust and direction of a Smart Fan Accessory attached to a Smart Cart.

Access the Control Panel



1. Connect the Smart Fan Accessory to a Smart Cart.
2. Connect the Smart Cart to SPARKvue.
3. Build a new experiment page and select any layout.
4. Select the Smart Fan Display  .

Turn the Fan On or Off

Do either of the following to turn the fan on or off:

- Click the **On** or **Off** buttons.
- Select the **Auto** checkbox. The fan automatically turns on when collecting data and turns off when data collection stops.

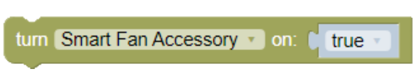
Set the Thrust and Direction

Do either of the following to set the thrust and direction of the fan:

- Enter a value between -100 and +100 in the **Thrust** field.
- Click and drag the **Thrust** slider between -100 and +100.

Create a Program

Create a program to control the Smart Fan using the Code  tool. The **Hardware** group provides two blocks to control the Smart Fan:

On/Off Block 

Use this block to turn the fan on or off. Set the parameter to `true` to turn on the fan or to `false` to turn off the fan. Use a block in the **Logic** group in place of the true/false block to create a logical expression to turn the fan on or off.

Thrust Block 

Use this block to set the thrust of the Smart Fan between -100 and +100. You can either use the default number block to set the thrust or replace this block with a block from the Math group.

Launch a projectile from the ballistic cart

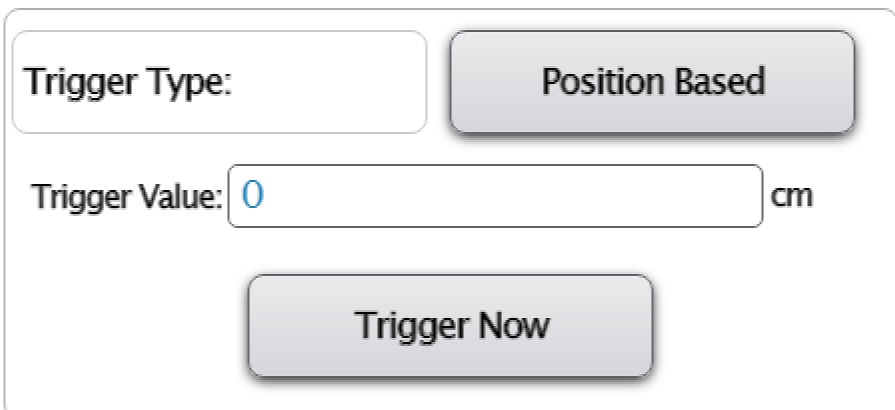
Launch a projectile from the Smart Ballistic Cart Accessory by setting a position or time based trigger.

> **Warning:**
> Wear safety glasses while using the Smart Ballistic Cart Accessory. Don't look down the barrel.

Access the Control Panel

1. Connect the Smart Ballistic Cart Accessory to a Smart Cart.
2. Connect the Smart Cart to SPARKvue].
3. Build a New Experiment. Select a layout that contains at least a half-page display element.
4. Select the Smart Ballistic Cart Accessory display .

Launch Options



Trigger Type

Select to trigger the launch mechanism based on the Smart Cart's position or a duration of time after starting data collection.

Trigger Value

Enter a value that triggers the launch mechanism based on the **Trigger Type**.

> **Tip:**
> Set a **Position Based** measurement with a value of 0. Start data collection, pull the cart back, then roll it forward. The projectile launches when the cart returns to its starting position.

Trigger Now

Click to launch the projectile manually. You can use this button while collecting or not collecting data.

Create a Program

Create a program to launch a projectile from the Smart Ballistic Cart Accessory using the Code tool. Access the block in the **Hardware** group.

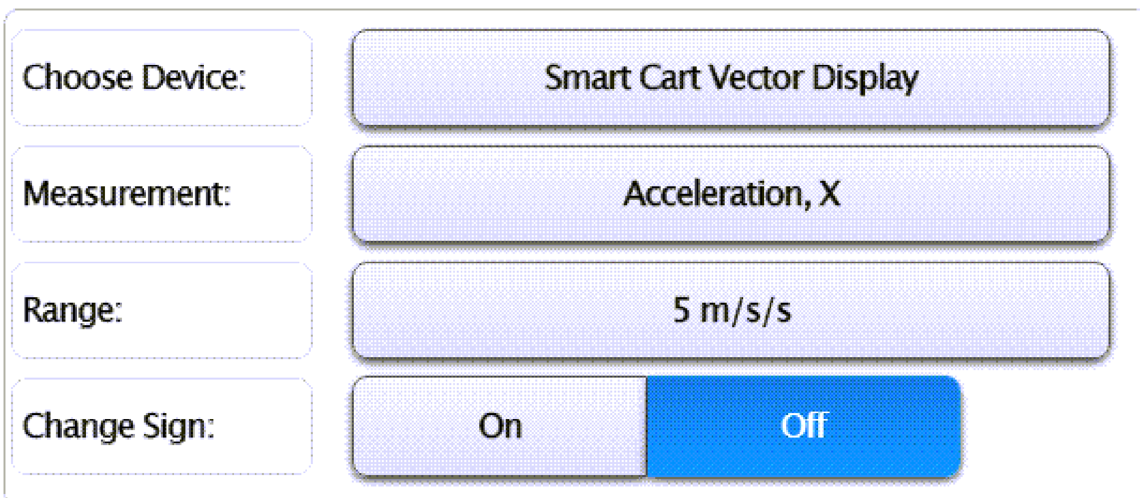
Change the Smart Cart Vector Display measurement

Change the measurement displayed on a Smart Cart Vector Display and configure the range and sign of the measurement.

Access the Control Panel

1. Connect the Smart Cart Vector Display to a Smart Cart.
2. Connect the Smart Cart to SPARKvue.
3. Build a New Experiment. Select a layout that contains at least a half-page display element.
4. Select the Smart Cart Vector Display .

Configuration Options



Choose Device

Indicates the selected Vector Display. To change, click the current device then select the display you want to configure.

Measurement

Indicates the current measurement displayed on the Vector Display. To change, click the current measurement then select the measurement you want to display.

Range

Indicates the maximum magnitude displayed on the Vector Display. To change, click the current range then select a range to display.

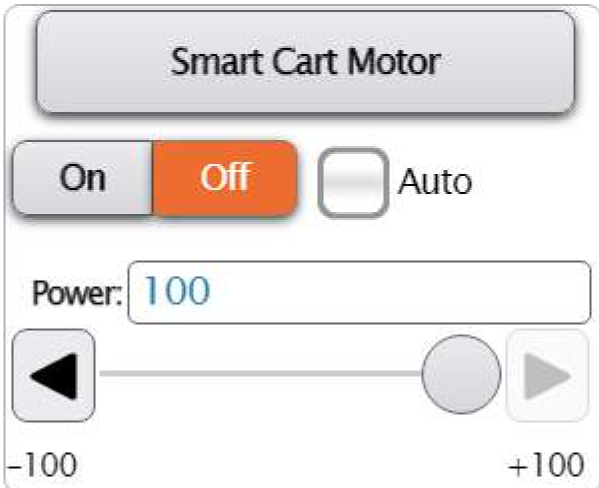
Change Sign

Toggle the switch to change the positive direction of the measurement.

Control the Smart Cart Motor

Control the speed and direction of a Smart Cart Motor attached to a Smart Cart.

Access the Control Panel



1. Connect the Smart Cart Motor to a Smart Cart.
2. Connect the Smart Cart to SPARKvue.
3. Build a new experiment page and select any layout.
4. Select the Smart Cart Motor Display .

Turn the Motor On or Off

Do either of the following to turn the motor on or off:

- Click the **On** or **Off** buttons.
- Select the **Auto** checkbox. The motor automatically turn on when collecting data and turns off when data collection stops.

Set the Speed and Direction

Do either of the following to set the speed and direction of the motor:

- Enter a value between -100 and +100 in the **Power** field.
- Click and drag the **Power** slider between -100 and +100.

Create a Program

Create a program to control the Smart Cart Motor using the Code tool. Access the following two blocks in the **Hardware** group:

On/Off Block 

Use this block to turn the motor on or off. Set the parameter to `true` to turn on the motor or to `false` to turn off the motor. Use a block in the **Logic** group in place of the true/false block to create a logical expression to turn the motor on or off.

Power Block 

Use this block to set the power of the Smart Cart Motor between -100 and +100. You can either use the default number block to set the power or replace this block with a block from the Math group.

Signal Generator

Use the Signal Generator tool to control a signal generator on the 550 Universal Interface and the Wireless AC/DC Module.

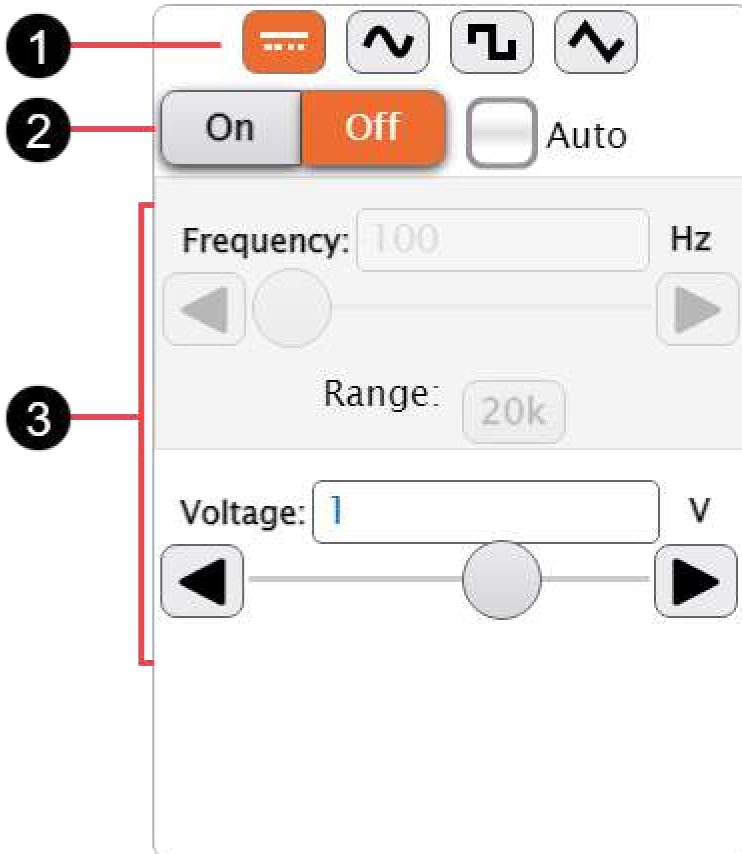
Display the output waveform

Use the Signal Generator tool and Scope display to an output waveform from a signal generator.

1. On the Welcome Screen, click Sensor Data.
2. Connect your device to SPARKvue.
3. Click Cancel to return to the Welcome Screen.
4. Click **Build New Experiment** on the Welcome Screen.
5. Select the 1/3-2/3 layout.

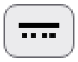
6. For the 1/3 section, select **Signal Generator** .
7. For the 2/3 section, select one of the following:
   - Select **Graph** for DC or low-frequency sampling.
   - Select **Scope** for high-frequency sampling.
8. Select a waveform.
9. Select **Auto** in the power control so that the signal generator automatically turns on while recording data.
10. Set the **Frequency** and **Amplitude** of the waveform.
11. If using the 550 Universal Interface, enable the Output Voltage sensor:
    a. On the bottom of the signal generator, click Signal Generator Properties .
    b. Toggle **Record Output Voltage** to **On**.
    c. Click **OK**.
12. In the data display, click **Select Measurement** then select **Output Voltage**.
13. Click **Start** to display the waveform on the data display. Use the Graph or Scope display tools to adjust the display.

Overview


1
2
On
Off
Auto
3
Frequency:
100
Hz
Range:
20k
Voltage:
1
V

1 Waveform

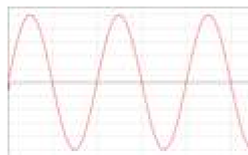
Select the type of waveform the signal generator outputs:

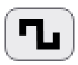
- **DC** 

  Outputs a constant voltage equal to the **Voltage**.

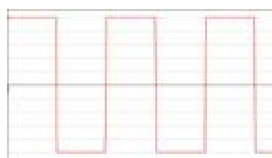
- **Sine** 

  Outputs an sinusoidal oscillation with positive and negative peaks equal to the **Amplitude**.

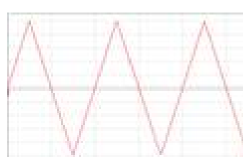
- **Square** 

  Outputs an oscillation that remains at a constant positive and negative voltage equal to the **Amplitude**. The voltage is positive for half of the oscillation and negative for the other half of the oscillation.

- **Triangle** 

  Outputs a linear oscillation with positive and negative peaks equal to the **Amplitude**.

2 Power Control

Control the signal generator power manually, automatically with data collection, or with a program in the Code tool:

- Click the **On** and **Off** buttons to control the power manually.
- Select the **Auto** checkbox to have the power automatically turn on when you click **Start**  and automatically turn off when you click **Stop** .

3 **Waveform Parameters**

Set the waveform parameters manually by entering a number in the box, clicking arrows, or dragging the slider. You can also use sensors to control the parameters by creating a program using the Code tool. The following parameters are available:

- **Frequency**

  Sets how often the waveform oscillates each second. Click the **Range** button to change the maximum frequency.

- **Amplitude**

  Sets the positive and negative peak voltages.

Calibration

Calibrate sensor measurements.

Calibrate a pH sensor

Improve the accuracy of pH measurements by calibrating the pH sensor.

Prepare the buffer solutions

Prepare two buffer solutions of different pH values to calibrate the pH sensor.

You can create buffer solutions using the pH Buffer Capsule Kit (SC-2321) or by purchasing buffer solutions from a chemical supplier. For best accuracy, use two pH buffers that are as close to the measured range as possible. For example, if measuring acids, use the pH 4 and pH 7 buffers for calibration.

Set the calibration points

Calibrate the sensor in the software after preparing buffer solutions.

1. Connect the sensor to SPARKvue and set up an experiment screen with a pH measurement.
2. Click the **pH** measurement in the live data bar then select **Calibrate measurement**.
3. In the **Calibrate Sensor** window, set each value as shown below then click **Continue**.

   - Select a sensor: **Wireless pH Sensor**
   - Select a measurement: **pH**
   - Select a Calibration Type: **2 point (Adjust Slope and Offset)**

4. Rinse the pH probe with distilled water and blot dry with a towel.
5. Place the probe in the lowest pH buffer.
6. Under **Calibration Point 1**, enter the value of the pH buffer in **Standard Value** then click **Set Calibration**.
7. Repeat steps 4 to 6 for **Calibration Point 2** using the higher pH buffer.
8. Click **OK**.

> **Note:**
> For the Wireless pH Sensor, the calibration remains on the sensor. You can restore the factory calibration, if needed. For all other pH sensors, the calibration remains on the SPARKvue file, if saved.

Restore the factory calibration

The Wireless pH Sensor stores calibration settings directly on the sensor. If needed, restore the calibration to the factory calibration settings.

1. Connect the sensor to SPARKvue and set up an experiment screen with a pH measurement.
2. Click the **pH** measurement in the live data bar then select **Calibrate measurement**.
3. In the Calibrate Sensor window, select Restore Factory Calibration.
4. Set each value as shown below then click **Restore**.
    - Select a sensor: **Wireless pH Sensor**
    - Select a measurement: **pH**
5. Click **OK**.

Calibrate a Wireless CO_2 Sensor

Calibrate the Wireless CO_2 sensor so that it reads 400 ppm for fresh air.

Standard calibration

1. Connect the sensor to your device and create a Graph display.
2. Insert the probe into the sampling bottle. Form an airtight seal with the rubber collar.
3. Allow the sensor to equilibrate for 3 minutes.
4. In the live data bar, click **CO2 Concentration** then select **Calibrate measurement**.
5. Click **Calibrate** then click **OK**.

Note:
If there is significant drift or the value is outside the expected range after performing the initial calibration, perform the **Extended calibration** procedure in the next section.

Extended calibration

Tip:
This procedure can be performed with more than one sensor simultaneously.

1. Take your sensor and computing device outside in fresh air. You do not need the sampling bottle. Place in the shade away from CO_2 sources such as cars and people.
2. Connect the sensor to your device.
3. Leave the sensor undisturbed and connected for about 25 minutes.
4. Return after 25 minutes and perform the standard calibration procedure.

Manage data

Set the data collection mode and sample rate, set a stop condition, and delete or rename data.

Delete data runs

Permanently delete data runs from an experiment.

1. On the bottom of the Experiment Screen click then click Manage Runs.
2. Select a delete run option:

   Delete Last Run

   Deletes the most recent data run.

   Delete All Runs

   Deletes all the runs collected in the file.

   Choose Run To Delete

   Select a specific run to delete.

3. Click OK to confirm to delete the run.
4. Click Done to exit Manage Runs.
5. Click Done to exit Experiment Tools.

Rename a run

Provide a useful description your data by renaming runs. You can also customize your experiment further by renaming sensors and measurements.

1. On the bottom of the Experiment Screen, click then click **Manage Runs**.
2. Click **Choose Run To Rename** then select the run you want to rename.
3. Click the **Edit Run Name** field and type a new name.

> **Tip:**
>
> Click Symbols or Greek Letters to enter special characters including subscripts and superscripts.

4. Click **Done** to exit Manage Runs.
5. Click **Done** to exit Experiment Tools.

Change the numerical format

Change the number of decimal places displayed by a display tool when analyzing data. Additionally, select the display style as fixed decimals, scientific notation, or significant figures.

1. On the bottom of the Experiment Screen, click  then click **Data Properties**.
2. Click **Select Measurement** then select a measurement to modify.
3. Select **Number Format** to expand the menu.
4. Click the buttons in the different fields to change the numerical format:

   - **Number Style**

     Display the measurement as fixed decimals, in scientific notation, or using significant figures.

   - **Digits**

     Select the number of digits to appear after the decimal place.

   - **Scientific Notation Transition**

     Select **On** to have the measurement switch to scientific notation when the measurement is a very small or large value. Set the value at which the measurement switches to scientific notation in the **Exponent Threshold** settings below.

   - **Small Exponent Threshold**

     The measurement switches to scientific notation when it's *less* than this value. For example, if you set the threshold to -4, the measurement displays scientific notation for values equal to or less than 0.0001.

   - **Large Exponent Threshold**

     The measurement switches to scientific notation when it's *more* than this value. For example, if you set the threshold to 7, the measurement displays scientific notation for values equal to or more than 10,000,000.

5. Click **OK**.

> **Note:**
> If a display tool is already active, you may need to remove the tool then reselect it for the numerical format changes to take effect.

Change the sample rate of a sensor

Change the amount of collected data points per unit of time.

The default sample rate works well for most cases. It may be desirable to use a higher sample rate when trying to record an event that occurs very quickly to gather additional data points. For long-term data collection over multiple hours, lower the sample rate to conserve memory and reduce the data file size.

1. On the bottom of the Experiment Screen click .
2. Click the menu in the Sensor field then select the sensor for the sample rate you want to set. If you want all sensors to have the sample rate, select Common Sample Rate.

> **Note:**
> SPARKvue may limit higher sampling rates for wireless sensors due to limitations in Bluetooth transfer rates. In general, avoid using a Common Sample Rate for high sample rates when using a wireless sensor that contains several sensors.

3. Under Sample Rate, use the left and right arrows to decrease or increase the sample rate.
4. Click OK.

SPARKvue displays the sample rate at the bottom of the Experiment Screen. The sample rate displays "Mixed" when setting different sample rates for each sensor.

Set Up and Use Manual Sampling

Manually record individual data points instead of continuously recording data.

While in Manual Sampling mode, data is continually *monitored* but not *recorded* until you click the Keep Sample button. Use this mode when recording a sensor measurement versus a non-sensor measurement, such as in a Boyle's Law or Beer's Law experiment. It's also useful to use this mode when recording a sensor measurement versus another sensor measurement.

1. Build a page that includes a table and a graph display. Add measurements to each display.

> **Tip:**
> Select either Sensor Data or Build New Experiment from the Welcome Screen to help set up your experiment.

2. Click Sampling Options .
3. Under Sampling Mode select Manual.
4. Click OK.

> **Note:**
> The sampling window now displays Manual and the keep button is now visible. The keep button also appears in the table after clicking Start.

5. Click Start to begin data collection. SPARKvue displays the current measurement but doesn't record it.
6. Click Keep to record a data point. Repeat for each data point you want to record.
7. Click Stop when finished collecting data.

Stop data collection automatically

Automatically stop data collection after a set period of time.

1. Click Sampling Options .
2. Under the Automatic Stop Condition section, click the Condition menu then select **Stop after duration**.
3. Click the Units menu to set the time units.
4. In the value field, enter a value for time.
5. Click OK.

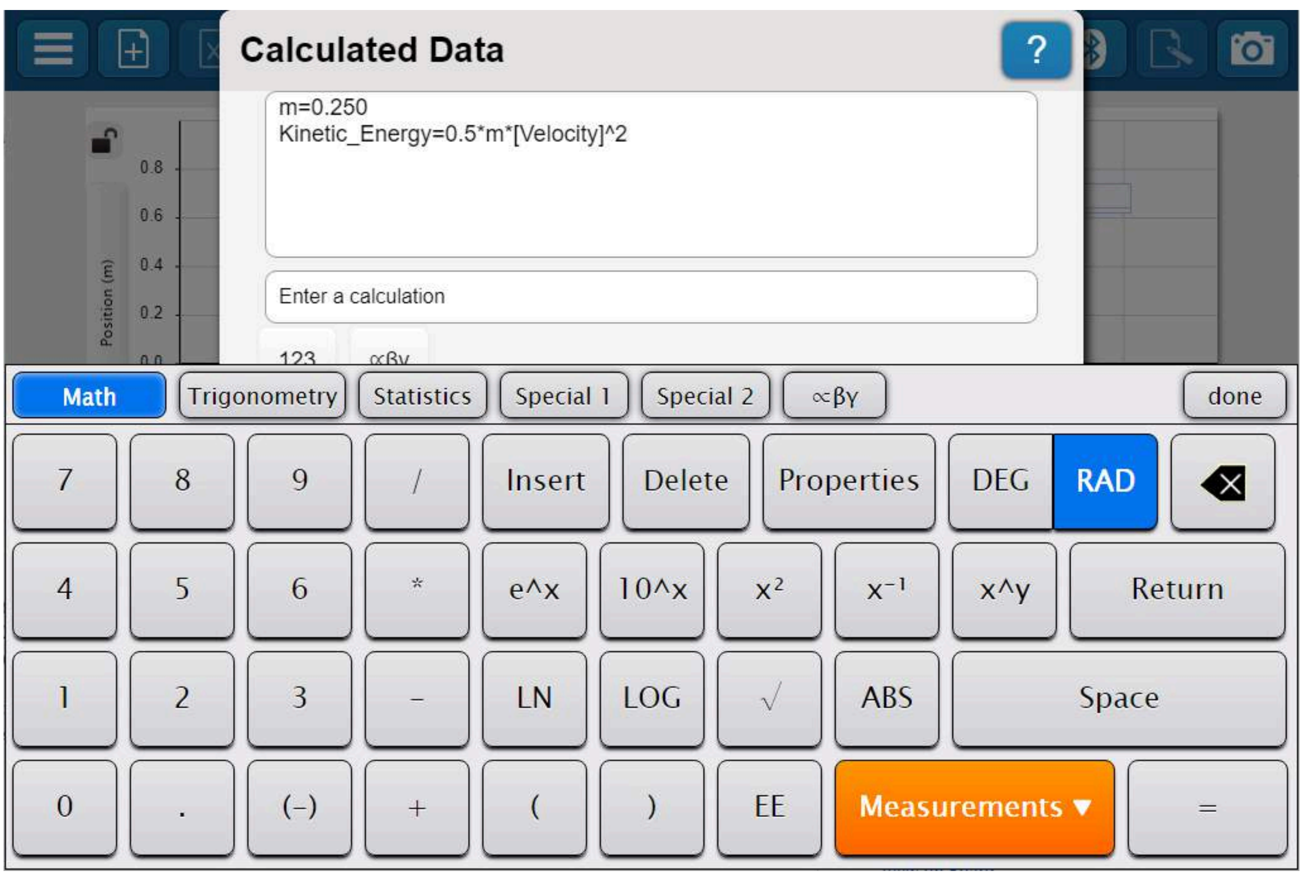
Click Start to begin data collection. Data collection automatically stops after the specified time unless you click Stop .

Create and display calculated data

Create an equation using sensor measurements and view the calculated data in a display.

Create a calculation

Use the Calculator to create equations and constants.



1. Open the Calculator by going to **Experiment Tools** > **Calculated Data**.
2. Enter a name for the calculation followed by an equals operator `=`.

> **Note:**
> Calculation names cannot contain a space.

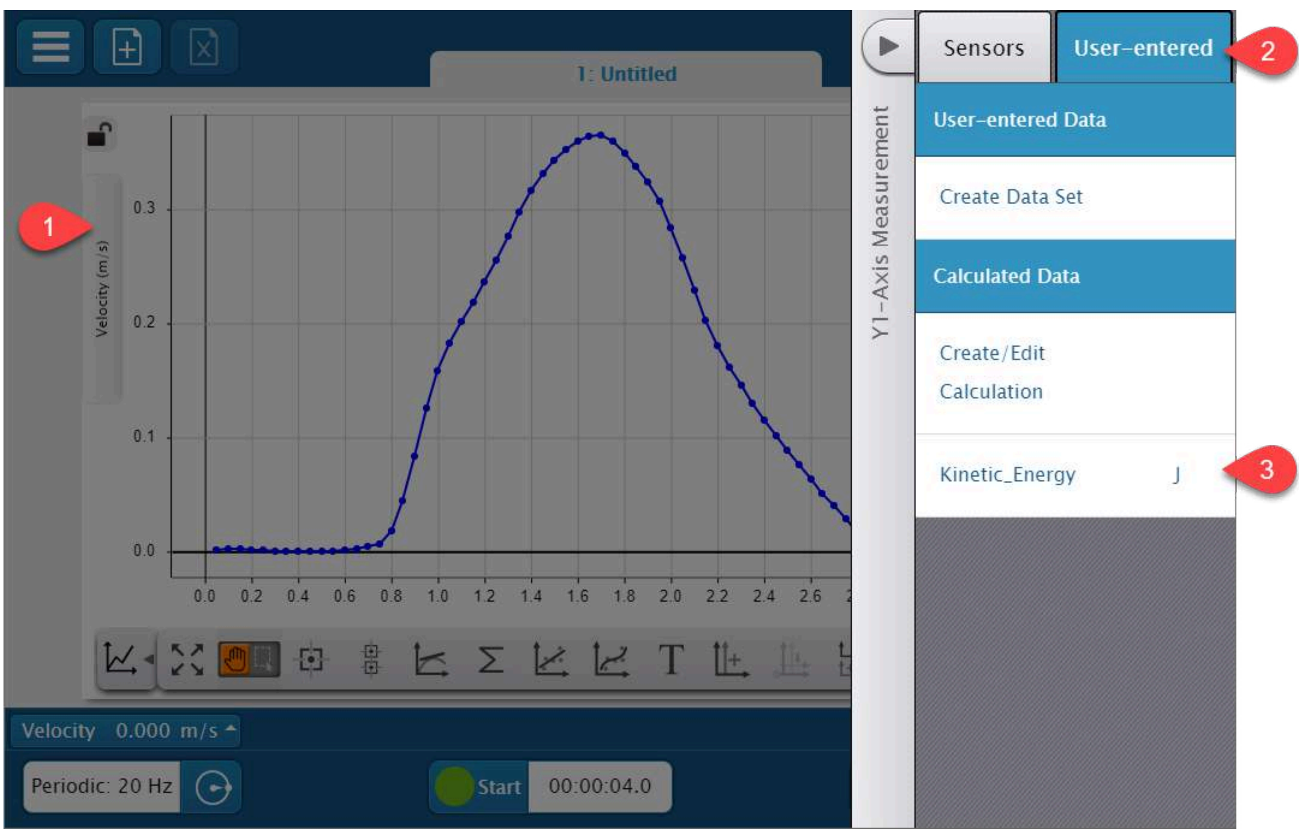
3. Enter the calculation on the right side of the equation.
   - To insert a measurement, click the Measurements button, then select a measurement from the list.
   - To insert a function or operator, use the function buttons. Click the calculator keyboard button 123 if the calculator buttons aren't appearing on the screen. See Function descriptions for explanations on how to use each function in the calculator.

> **Note:**
> If your calculation contains a constant (such as *m* in the preceding example), the Calculator automatically creates a definition for the constant on the preceding line. You need to provide a value for the constant for it to be valid.

4. Click **done** then click **OK**.

Display a calculation

Visualize calculated data by showing it a data display, such as a graph.



1. Click the current measurement in the display.
2. In the measurement pane, click the **User-entered** tab.
3. Under **Calculated Data**, select the calculation.

Function descriptions

This reference provides descriptions of each function included in the Calculator.

Math

e^x

Raises e to the x power.

```
exp(x)
```

`x` = data or value

10^x

Raised 10 to the succeeding value.

x^2

Squares the preceding value.

x^{-1}

Inverts the preceding value.

x^y

Raises x to the y power.

LN

Calculates the natural logarithm of x.

```
ln(x)
```

`x` = data or value

LOG

Calculates the logarithm (base 10) of x.

```
log(x)
```

`x` = data or value

√

Calculates the square root.

```
sqrt(x)
```

`x` = data or value

ABS

Calculates the absolute value of x.

```
abs(x)
```

`x` = data or value

Trigonometry

SIN

Trigonometric sine function.

```
sin(x)
```

`x` = data or value

COS

Trigonometric cosine function.

```
cos(x)
```

`x` = data or value

TAN

Trigonometric tangent function.

```
tan(x)
```

`x` = data or value

SIN^{-1}

Trigonometric arcsine function.

```
arcsin(x)
```

`x` = data or value

COS^{-1}

Trigonometric arccosine function.

```
arccos(x)
```

`x` = data or value

TAN^{-1}

Trigonometric arctangent function.

```
arctan(x)
```

`x` = data or value

Statistics

Σ min

Determines the running minimum value of the data source x.

```
min(x)
```

`x` = data only

Σ max

Determines the running maximum value of the data source x.

```
max(x)
```

`x` = data only

Σ avg

Calculates the running average value (mean) of the data source x.

```
avg(x)
```

`x` = data only

Σ σ

Calculates the running standard deviation of the data source x.

```
stddev(x)
```

`x` = data only

Σ count

Calculates the running count (number of data points) of the data source x.

```
count(x)
```

`x` = data only

Σ sum

Calculates the running sum of the data source x.

```
sum(x)
```

`x` = data only

Special 1

integral

Calculates the numerical integral of the data source y with respect to the data source x by determining the running sum of the area under each y data point with respect to its corresponding x data point.

```
integral(y,x)
```

`y` = data only

`x` = data only

derivative

Calculates the numerical derivative of the data source y with respect to the data source x by calculating the slope of a line segment from one data point in the sequence to a second data point in the sequence n-1 points away. If n = 2, the derivative values will be the slope between the first and second point, second and third point, and so on. If n = 5, the derivative values will be the slope between the first and fifth point, second and sixth point, third and seventh point, and so on.

```
derivative(n,y,x)
```

`n` = integer value ≥ 2

`y` = data only

`x` = data only

first

Returns a data set with all values equal to the first value of the data source x. The length of the returned data set will be identical to the length of the data source x.

```
first(x)
```

`x` = data only

period

Determines the period of the data source x (assuming it is oscillating) by detecting peaks in the data and measuring the time between peaks. The values 'peak%' and 'valley%' set the thresholds to finding the peaks in the data source x. If peak% and valley% are both equal to 10, the function searches for peaks in the top and bottom 10% of the data range. The value 'time' is the time range over which the algorithm is detecting peaks. The peak%, valley%, and time values can be adjusted to tune the function for best results.

```
period(peak%, valley%, time, x)
```

`peak%` = value only

`valley%` = value only

`time` = value only

`x` = data only

amplitude

Determines the amplitude of the data source x (assuming it is oscillating) by detecting peaks and valleys in the data and measuring half the thresholds to finding the peaks and valleys in the data source x. If peak% and valley% are both equal to 10, the function searches for peaks and valleys in the top and bottom 10% of the data range. The value 'time' is the time range over which the algorithm is detecting peaks and valleys. The peak%, valley%, and time values can be adjusted to tune the function for best results.

```
amplitude(peak%, valley%, time, x)
```

`peak%` = value only

`valley%` = value only

`time` = value only

`x` = data only

peakamp

Calculates half the distance between the maximum and minimum values in every time interval 'time' for the data source x.

```
peakamp(time, x)
```

`n` = integer value ≥ 1

`x` = data only

Special 2

filter

Returns a copy of the data source x with all values less than 'min' and greater than 'max' removed.

```
filter(min, max, x)
```

`min` = value only

`max` = value only

`x` = data only

smooth

Smooths the data from the data source x using the Savitzky-Golay method. Increasing n increases the amount of smoothing. The parameter n must be an odd number.

```
smooth(n, x)
```

`n` = integer value ≥ 5

`x` = data only

avgfilter

Smooths the data from the data source x by averaging every n values into one value. The number of points in the resultant data set will have a factor of n fewer points than the original data source x.

```
avgfilter(n, x)
```

`n` = integer value ≥ 2

`x` = data only

random

Returns a random value between zero and one.

```
random()
```

Share and save data

Share live data collection with other devices, export data to a spreadsheet, and create a journal for a lab report.

Share Live Data Collection

Use the Shared Session feature to share live data collection with multiple devices from a single device.

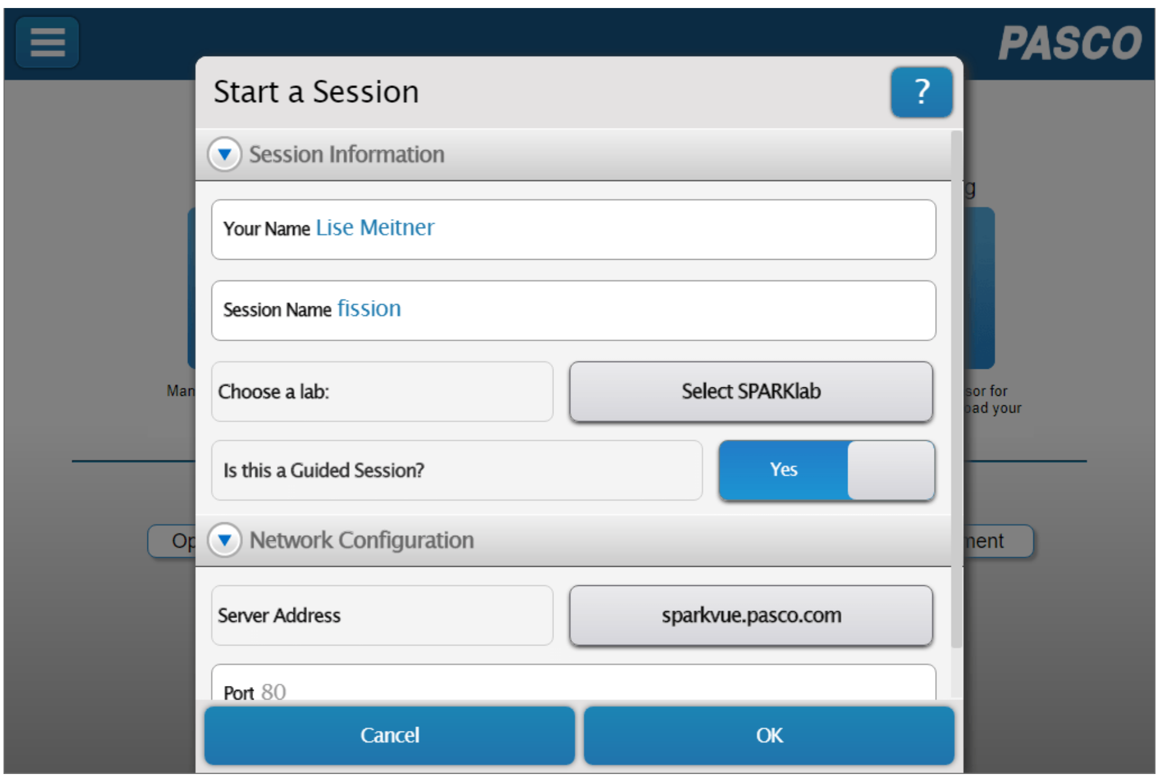
Use a Shared Session to:

- Share data with students during a demonstration.
- Distribute an experiment file to multiple devices.
- Share data with all members of a lab group while one device collects the data.

Start a Shared Session

Start a Shared Sessions from the Welcome Screen or the Experiment Screen:

- If want to share an existing file, start the session from the Welcome Screen.
- If you want to share data from a new experiment or one in progress, start the session from the Experiment Screen.



1. Click the Main Menu then select **Shared Session**.
2. Click **Start a Shared Session**.
3. Click the **Your Name** field and type a name for your device.

4. Click the **Session Name** field and type a name for the session. Use this Session Name to allow users to connect to the session.

> **Tip:**
> Create a one-word, lowercase name to make it easy for others to connect. The Session Name is case sensitive.

5. (Welcome Screen only): Click **Select SPARKlab** then select the file you want to use for the Shared Session.
6. Toggle the switch to select whether or not this is a **Guided Session**.
    - Select **Yes** to only allow your device to have control over data collection.
    - Select **No** to allow any device to have control over data collection.
7. Click **OK**.

The Shared Session has begun and the **Connected Users** panel appears on the right side of the screen.

Connect to a Shared Session

Connect to a Shared Session started by another device.

1. Click the Main Menu  then select **Shared Session**.
2. Click **Manually connect to an opened session**.
3. Click the **Your Name** field and enter a name for your device.
4. Click the **Session Name** field and enter the name of the session provided to you by the session host.
5. Click **OK**.
6. Wait for the session host to approve your connection.

Manage a Shared Session

After starting a Shared Session the Connected Users panel appears. Use the Connect Users panel to manage the Shared Session.

Approve and Reject Users

Users appear in the Connected Users panel when trying to connect to the shared session. You can approve or reject users on an individual basis or click Approve All to approve all users at once.

User List

Click to show or hide the Connected Users panel.

Disconnect Session

Click to disconnect the Shared Session. The file and data from the session remains on the disconnected devices.

Session Info

Click to show the name of the Shared Session.

Export and open data in a spreadsheet

Export data as a CSV (comma separated values) file to open with a spreadsheet app such as Excel or Google Sheets.

PC, Mac, and Chromebook

1. Click Main Menu then select **Export Data**.
2. Select a location to save the CSV file then click **Save**.
3. Open the CSV file in your spreadsheet app.

Tablets and mobile devices

1. Tap Share .
2. Tap **Share Data**.
3. In the share sheet, select an app to share the data with.

Code with Blockly

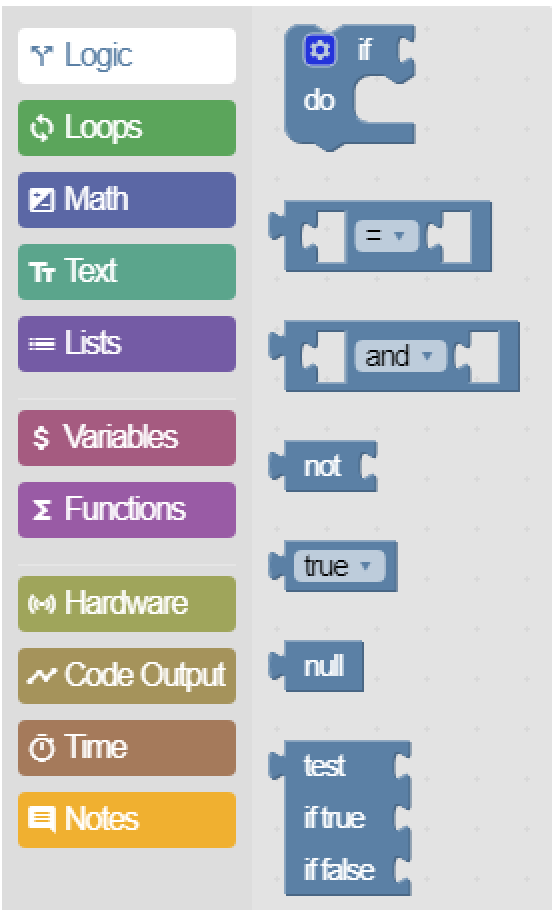
Create programs with the Code Tool using Blockly. Use sensor measurements to report data and control output devices.

Blockly Toolbox

Learn how to use the blocks available in each Blockly Toolbox category.

Logic

Use the Logic blocks to create if-then statements and comparison operations.



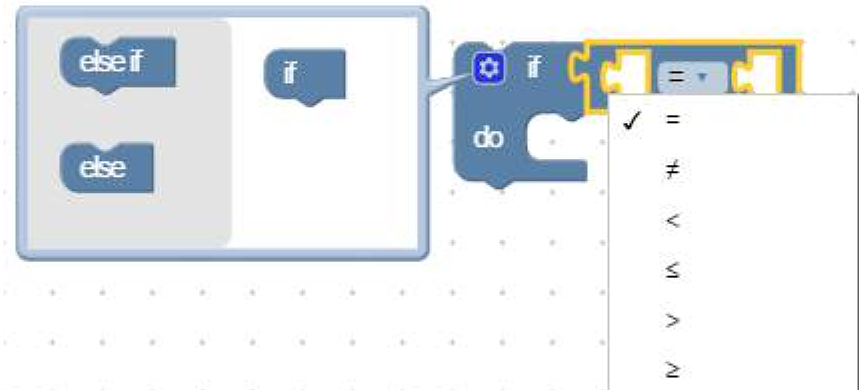
If Statement Block

An **if** statement does actions if a value is true. Add an **else if** or **else** statement to the **if** block by clicking .

The **else if** statement gets checked if the first **if** statement fails. If all the **else if** statements fail, then the code does the else statement.

If Statement with the Comparator Block

The comparator block returns true or false based on the comparison between two values.



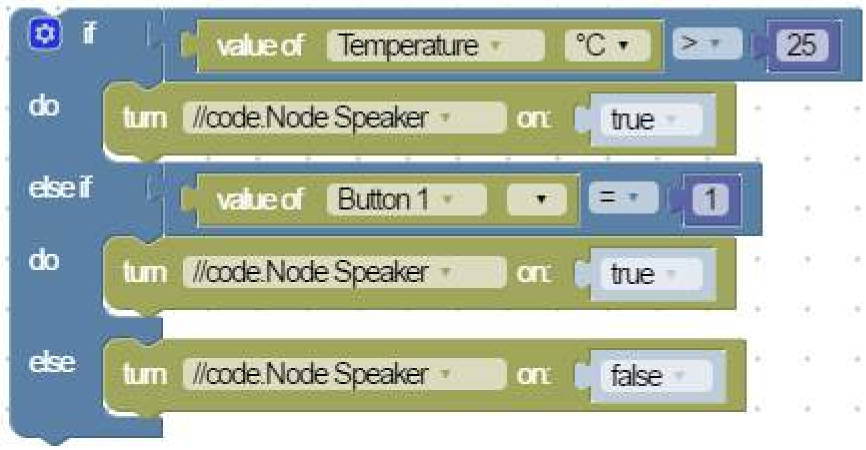
Possible comparisons include:

- Equal to `=`
- Not equal to `≠`
- Less than `<`
- Less than or equal to `≤`
- Greater than `>`
- Greater than or equal to `≥`

Example 1

In this example, if the temperature of a sensor is greater than 25 °C, the speaker turns on.

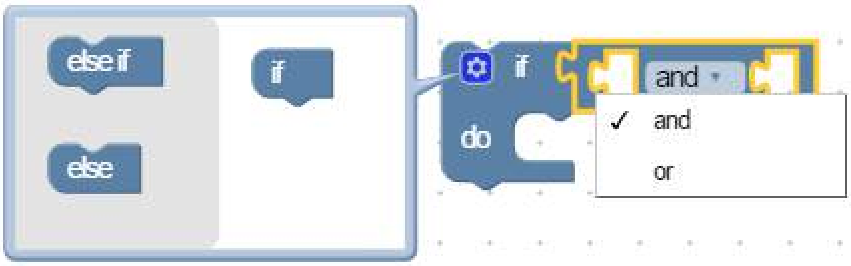
Example 2



This example uses **else if** and **else** statements. The speaker turns on if the temperature is greater than 25 °C or you press Button 1. The speaker turns off if neither scenario is true.

If Statement with the Comparator and Bitwise Block

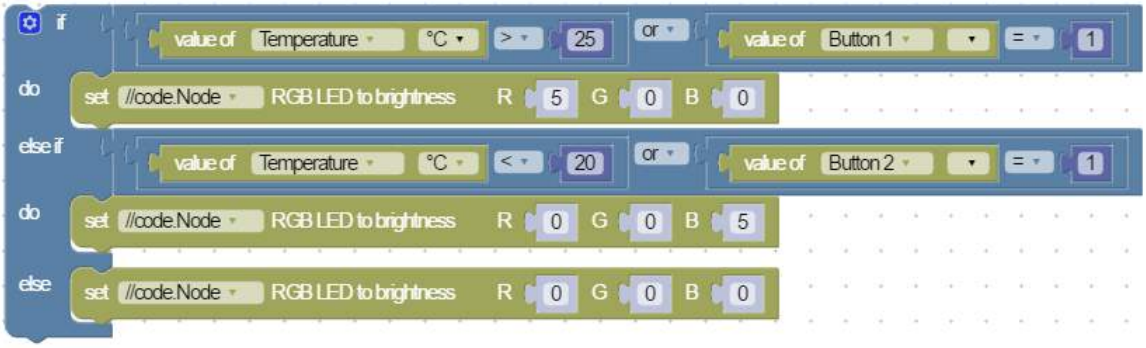
The bitwise block returns true or false based on whether two statements are both true (`and`) or if at least one of the statements is true (`or`).



Example 3

In this example, if the temperature of a sensor is greater than 25 °C *or* you press Button 1, the red `R` LED turns on.

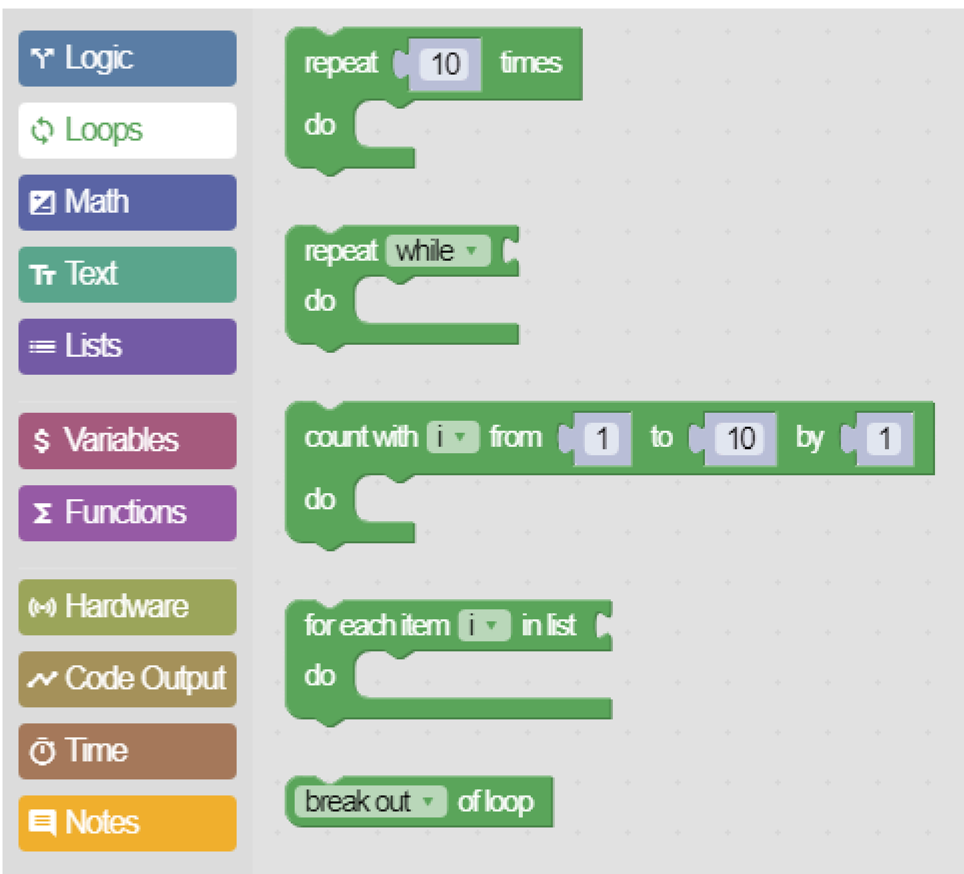
Example 4



Adding to Example 3, if the temperature is less than 20 °C *or* you press Button 2, the blue `B` LED turns on. Otherwise, all the LEDs turn off.

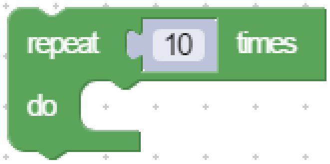
Loops

Use the loop blocks to control the number of times a program executes lines of code.

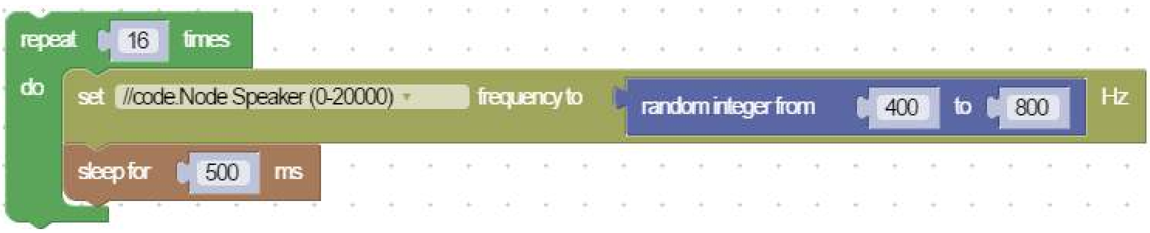


Repeat [number of] times

This block repeats code a specified number of times.



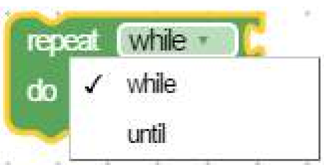
Example 1



In this example, the //code.Node speaker plays 16 random frequencies for 0.5 seconds each.

Conditional Repeat

The block repeats a task *while* a condition is valid or *until* it meets a condition.



Example 2: *While* Loop

A while loop continues as long as a condition is valid. In this example, the LED remains set to cyan until the program stops.

Example 3: *Until* Loop

The until statement keeps looping until it meets a condition. In this example the loop is keeping the blue LED on. If the light sensor detects darkness, the loop stops then turns the speaker on for 8 seconds.

Counting Loop

This block counts up at a specified integer until it reaches the range limit.

Example 4

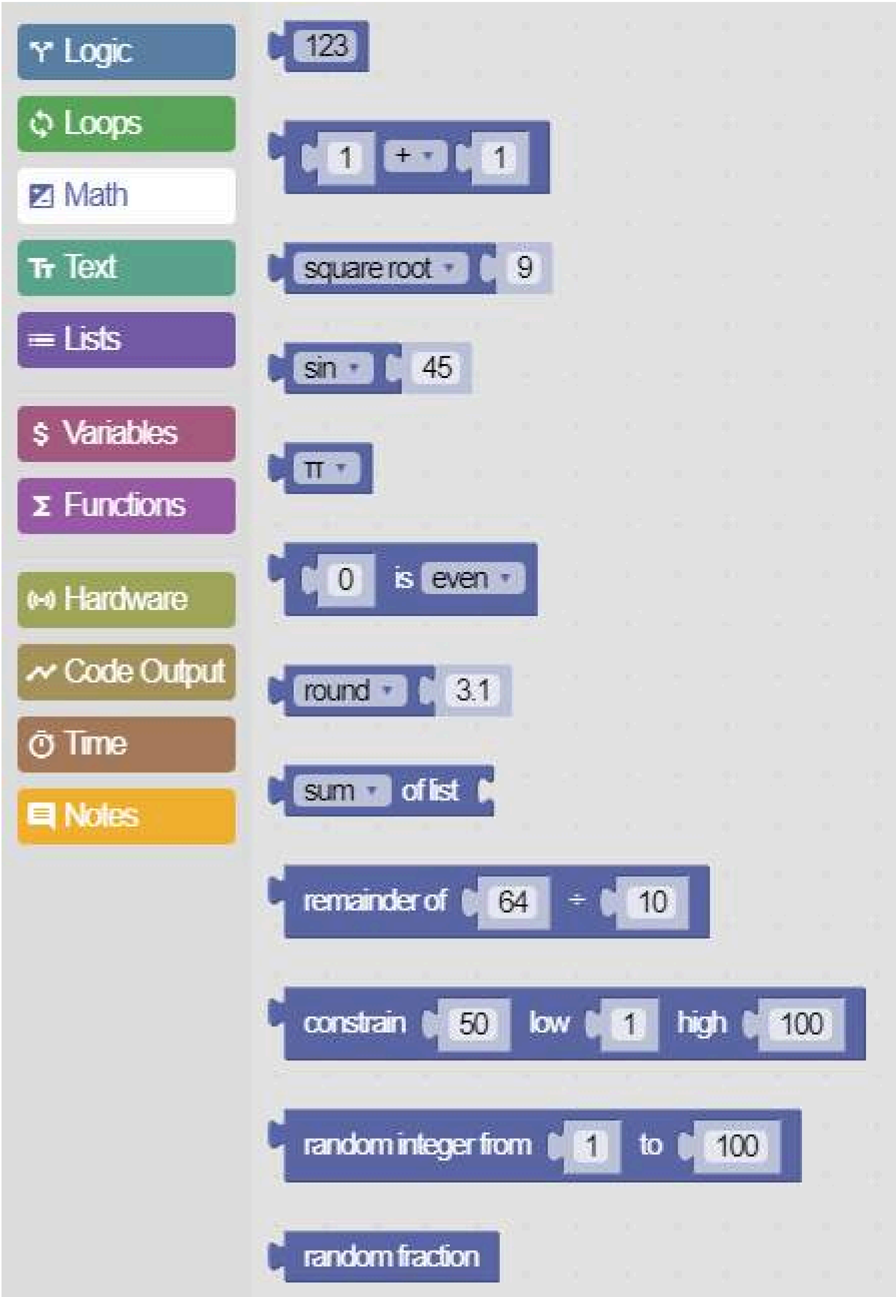
This loop initially sets the green LED's brightness to 1 and increments by 1 until it reaches 10. The LED stays at each brightness level for 2 seconds.

Example 5

This loop starts the speaker at a frequency of 200 Hz and increases by 100 Hz until the loop reaches 1400 Hz. Each frequency plays for 0.5 seconds.

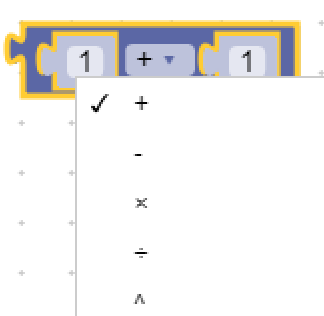
Math

Use the Math blocks to include numbers in code and perform mathematical operations.



Basic Math

Basic Operators

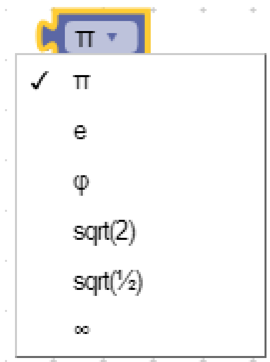


Use this block to perform basic mathematical operations on two numbers and return the result. Available operations include addition, subtraction, multiplication, division, and exponents.

Remainder

This block returns the remainder from dividing two numbers. For example, the block shown here returns a value of `4`.

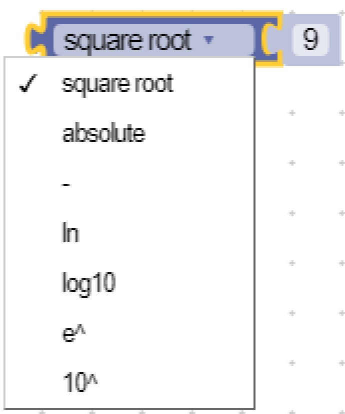
Common Constants



Use this block to return the value of a common constant including pi, natural exponent, golden ratio, square root of 2, square root of ½, and infinity.

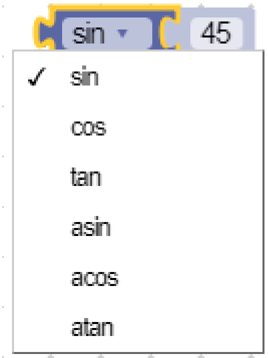
Functions

Scientific Functions



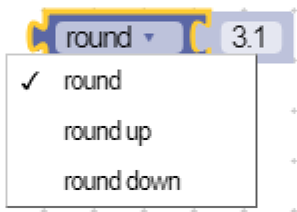
Use this block to performs a function on a single number and return the result. Available function include square root, absolute value, inverse, natural logarithm, base 10 logarithm, exponential, and power of 10.

Trigonometric Functions



Use this block perform a trigonometric function on a single number in *degrees* and return the result. Available functions include sine, cosine, tangent, arcsine, arccosine, and arctangent.

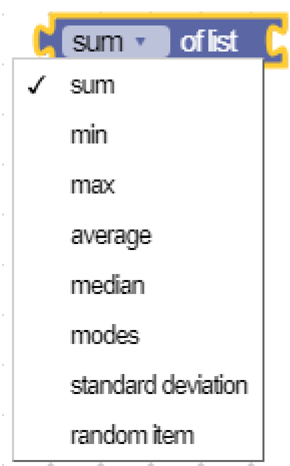
Rounding Functions



Use this block to round numbers to the nearest integer and return the result. Available options include:

- `round`: Round down for decimals less than 0.5, round up for decimals greater than 0.5.
- `round up`: Always round up the number regardless of the decimal value.
- `round down`: Always round down the number regardless of the decimal value.

Statistical Functions



Use this block to calculate statistics on two or more numbers and return the result. Unlike the other blocks in the Math group, this block requires a list of numbers instead of a single number. Available options include:

- `sum`: Add together all numbers in the list.
- `min`: Return the smallest number in the list.
- `max`: Return the largest number in the list.
- `average`: Return the mean of the numbers in the list.
- `median`: Return "the middle" value in the list.

- `mode`: Return the values that appear most often in the list.
- `standard deviation`: Return the standard deviation of the numbers in the list.
- `random item`: Return a random number in the list.

Random Numbers

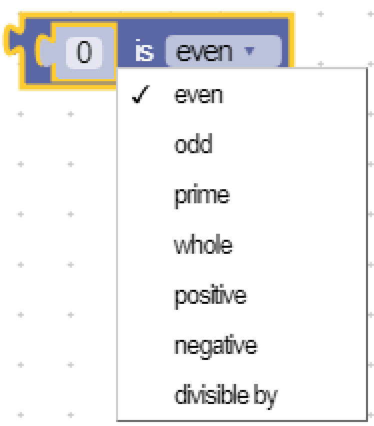
Random Integer

This block returns an random integer between the two specified numbers.

Random Fraction

This block returns a random decimal between 0 and 1.0.

Check a Number



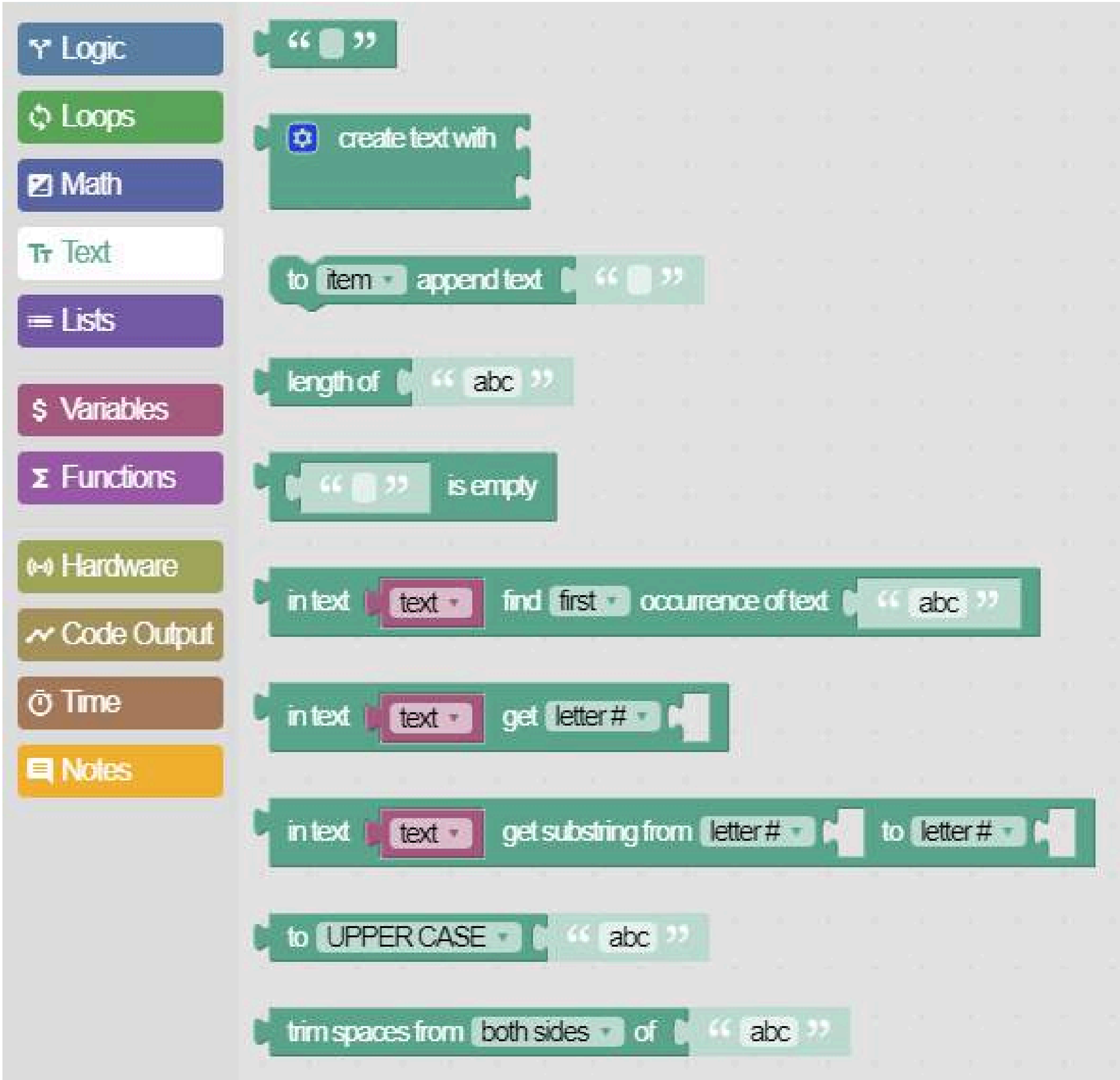
Use this block to check the type of number and return a value of `true` or `false`. The block can check if a number is even, odd, prime, whole, positive, negative, or divisible by a given number.

Constrain a Number

This block returns a number that's constrained between two specified values. If the number is higher than the specified high number, the block returns the high number. If the number is lower than the specified low number, the block returns the low number.

Text

Use the Text blocks to create and perform operations on strings of text.


Logic
Loops
Math
Text
Lists
Variables
Functions
Hardware
Code Output
Time
Notes
create text with
to item append text
length of abc
is empty
in text text find first occurrence of text abc
in text text get letter #
in text text get substring from letter # to letter #
to UPPER CASE abc
trim spaces from both sides of abc

Example 1


set myName to Paul
in text output firstInitial enter
in text myName get first letter

In this example, the first line sets the variable `myName` to the string `Paul`. The second line displays a text output as the first letter of `myName` in a digits display.

Example 2

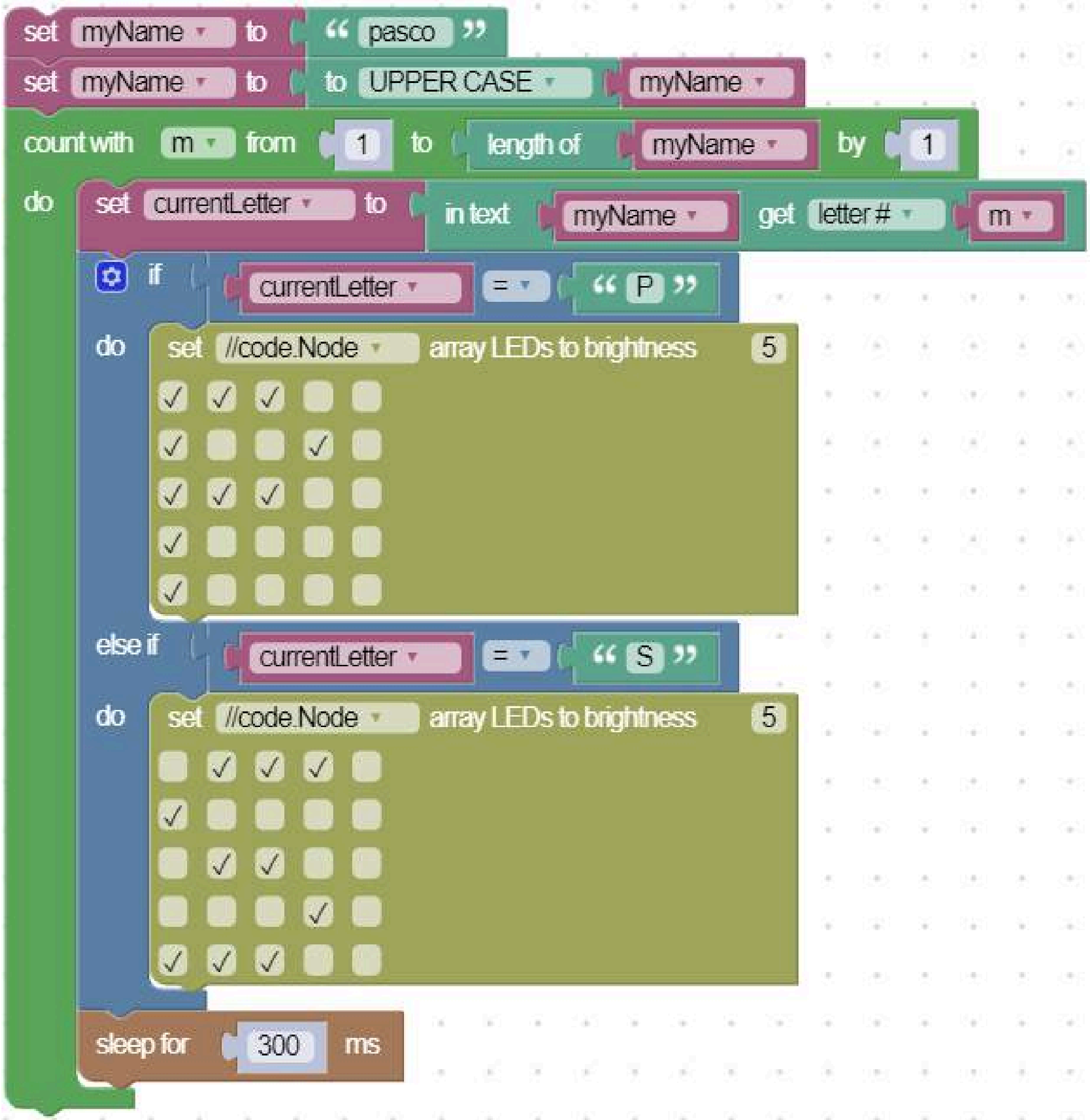
This example uses an append block that adds text onto an existing text variable. Set the variable `myName` to "`Paul`" then append "` Stokstad`". The full name displays into the `fullName` Text Output block.

Note:
Did you notice the space added before the second string? Add a space so that there is a space between the two strings.

Example 3

The **create text with** block joins together text from two or more blocks. In this example, the string "`Random di: `" joins with the **random integer** math block, which displays a random number between 1 and 6.

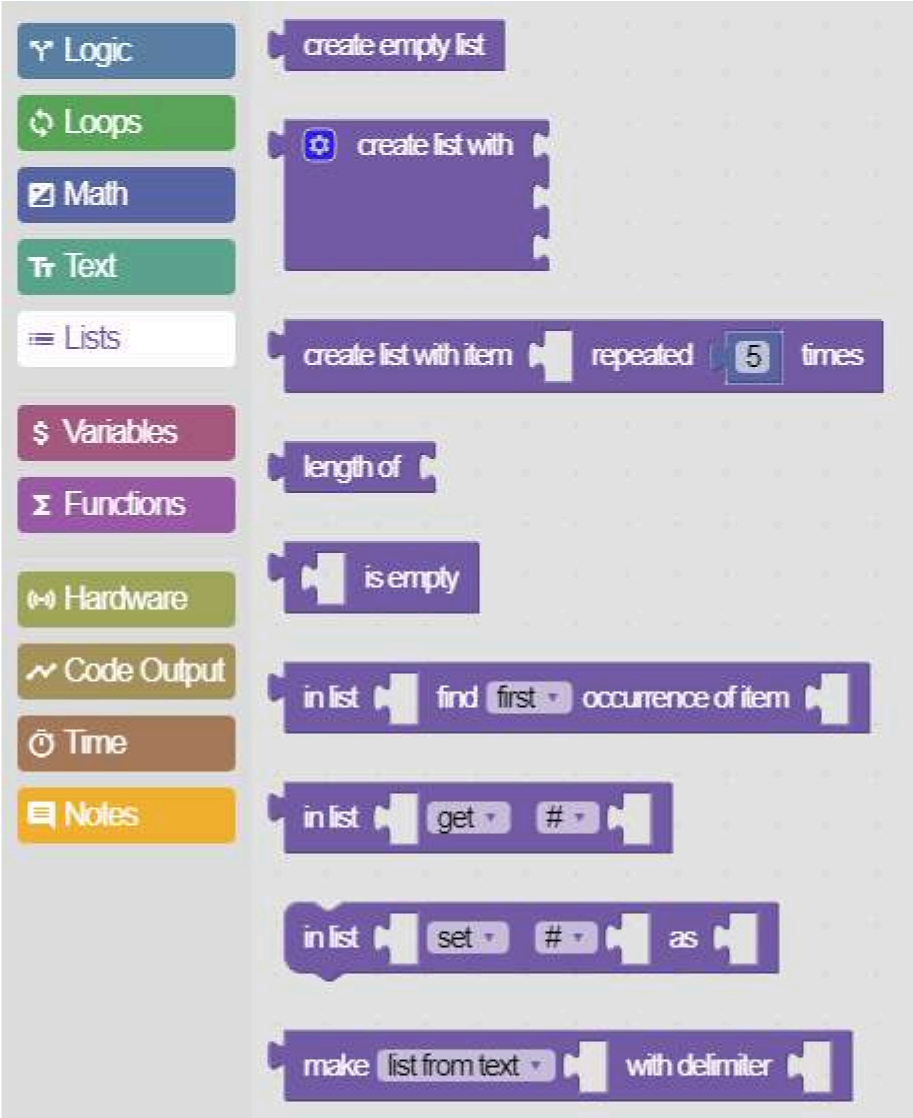
Example 4



In this example the string "`pasco`" changes to uppercase and displays the letters "`P`" and "`S`" on the //code.Node LED array.

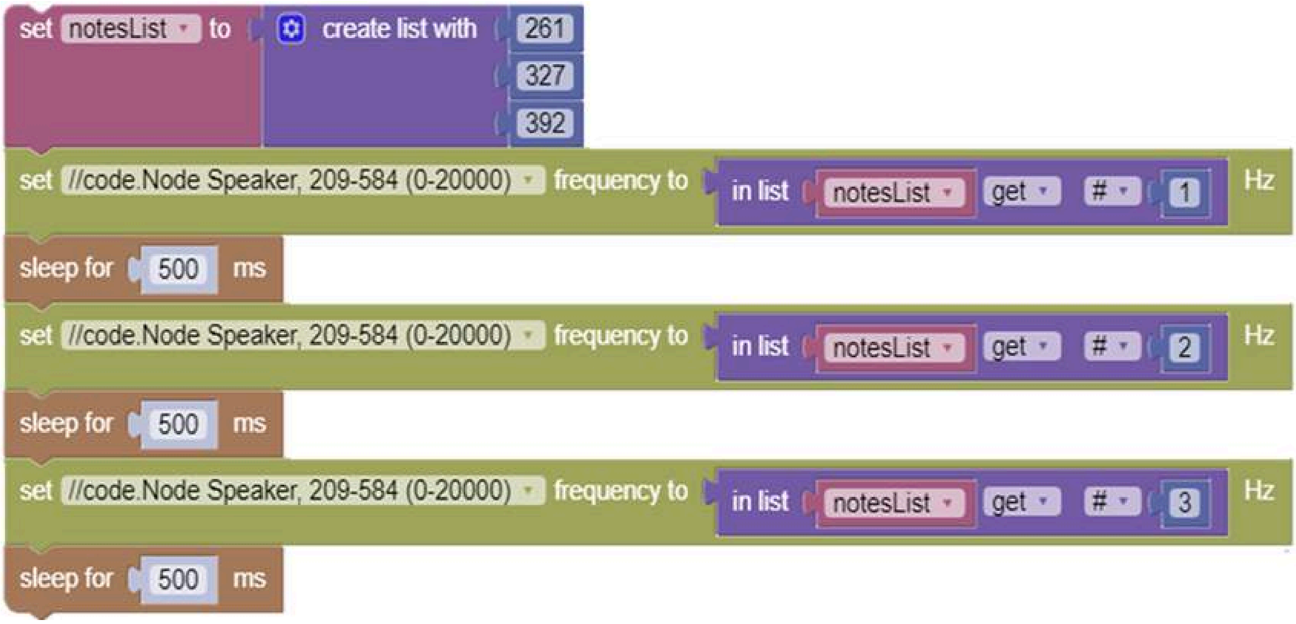
Lists

Use the List block to create an ordered collection of text or numbers.



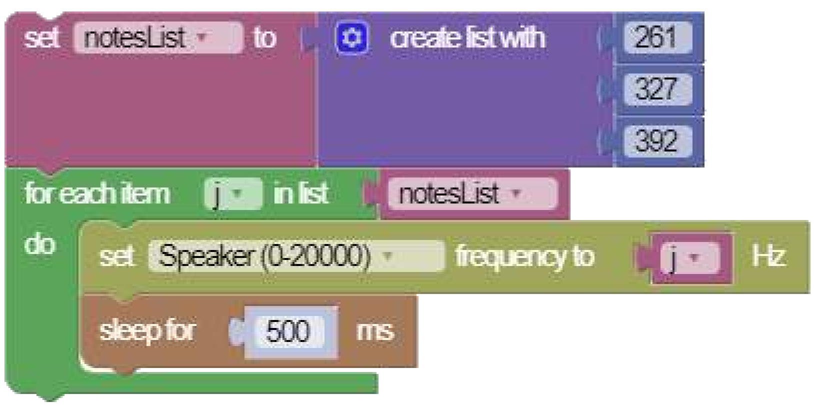
Get values from a list

The simplest use of a list is to create a list of values and then call each value individually. In this example, we're assigning a list of three numbers to the variable `notesList`. The program calls each value individually by referencing its position in the list: 1, 2, and 3. The program then uses each value with a frequency block to output a tone on the //code.Node speaker for 0.5 seconds (500 ms).



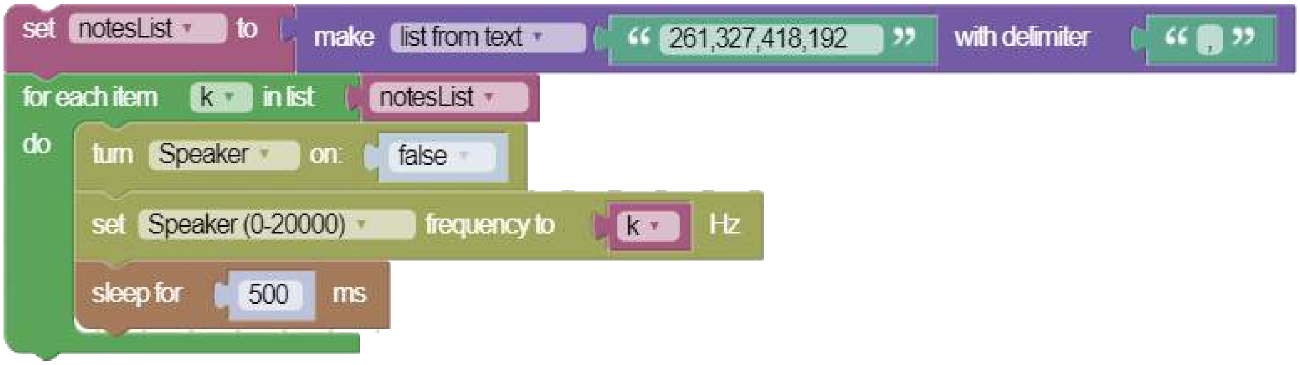
Using a loop

We can reduce the amount of code in the previous example by using a loop. The Loops category has a loop specifically for use with lists. This loop works like a counting loop, where the program calls each item in the list with each cycle of the loop. In this example, the loop assigns the `j` variable to the first list item (1), then repeats the code for the second list item (2), and repeats the code once more for the last list item (3).



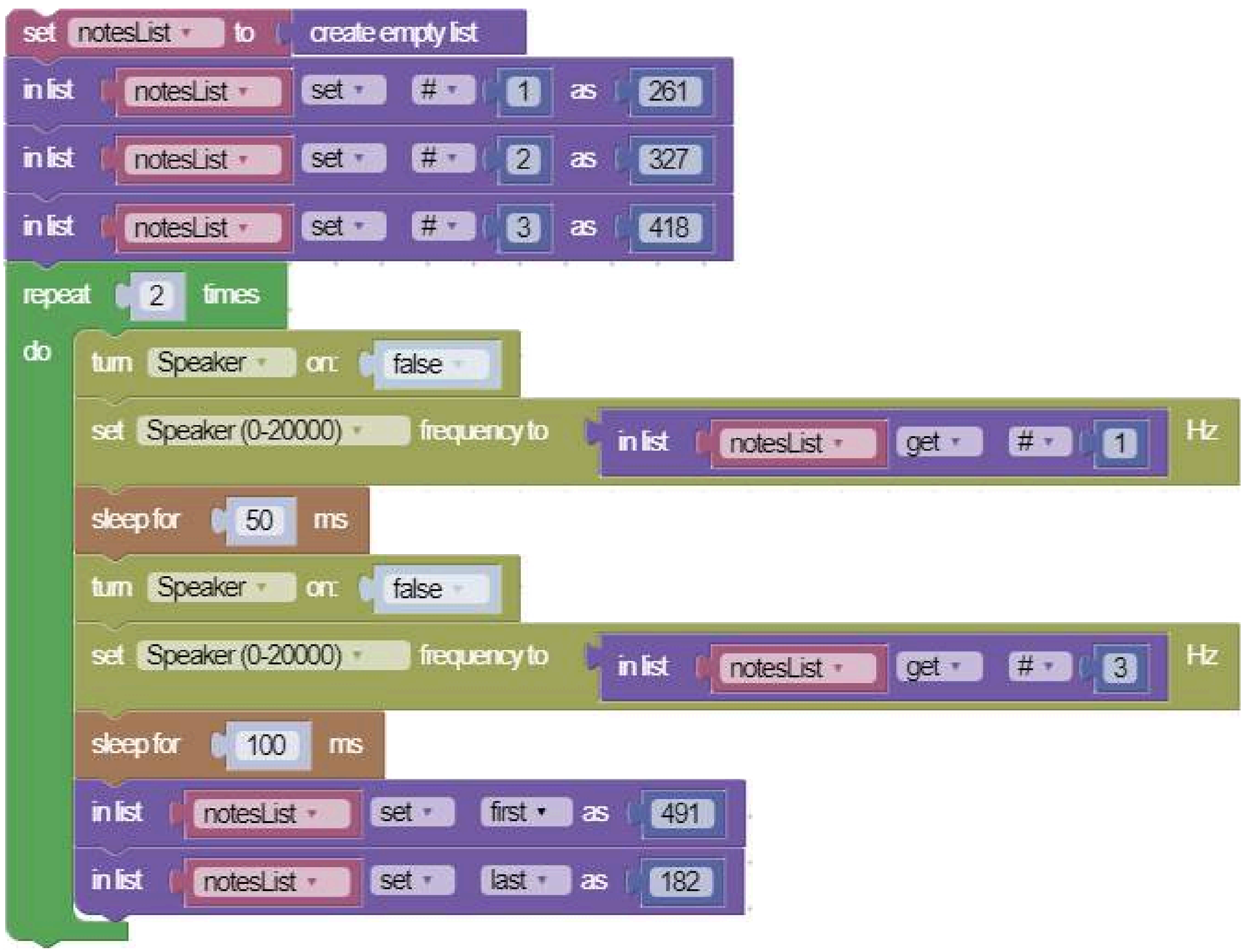
Create a list from text

You can create lists from a string of text containing multiple values separated by a delimiter. In this example, we create a list of numbers by entering values in a text block and separating them with a comma. In the list block, we also tell the program to use the comma as the delimiter. As a result, the //code.Node speaker outputs the frequencies of 261, 327, 418, and 192, playing each for 0.5 seconds.



Create a list one value at a time

You can set a variable from a list of numbers one value at a time. In this example, set a variable to a list containing three different integers. In the loop, the //code.Node speaker plays the frequencies of the #1 and #3 values in the list, change the first and last values of the list, and play those values again.

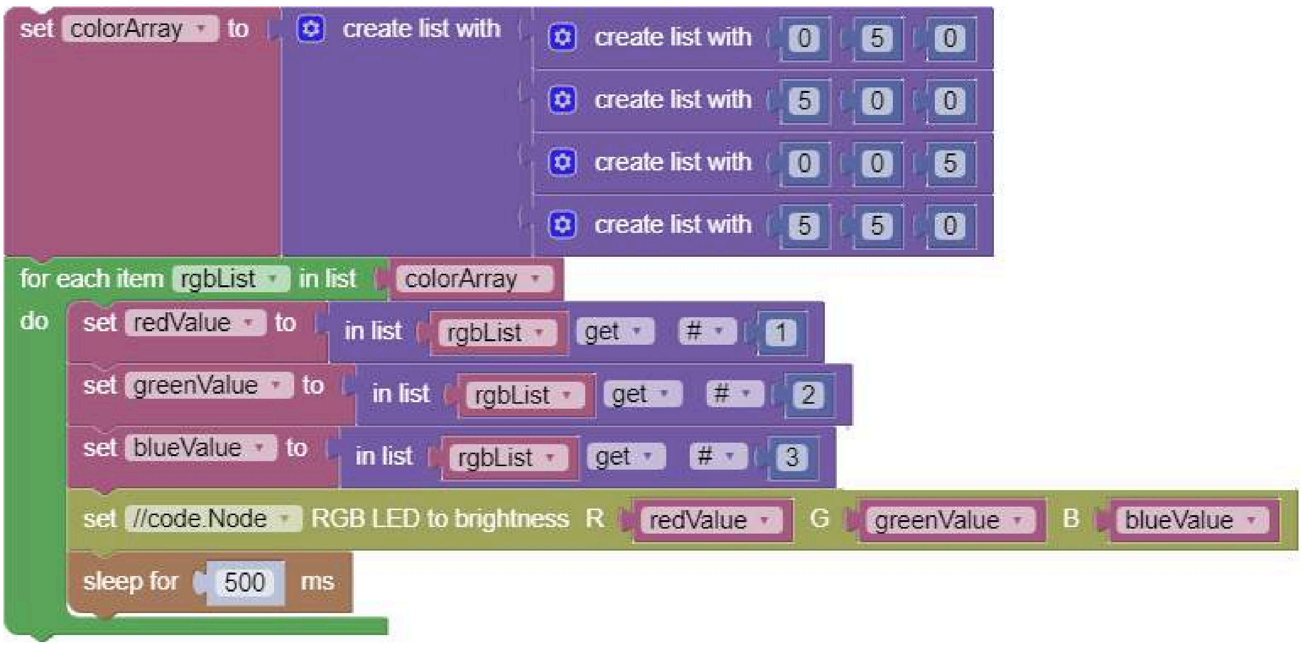


The speaker output sounds different than the first round because the code changed the frequencies after going through the loop one time. The end result is that the //code.Node speaker outputs the frequencies 491 and 182.

Create an array of values

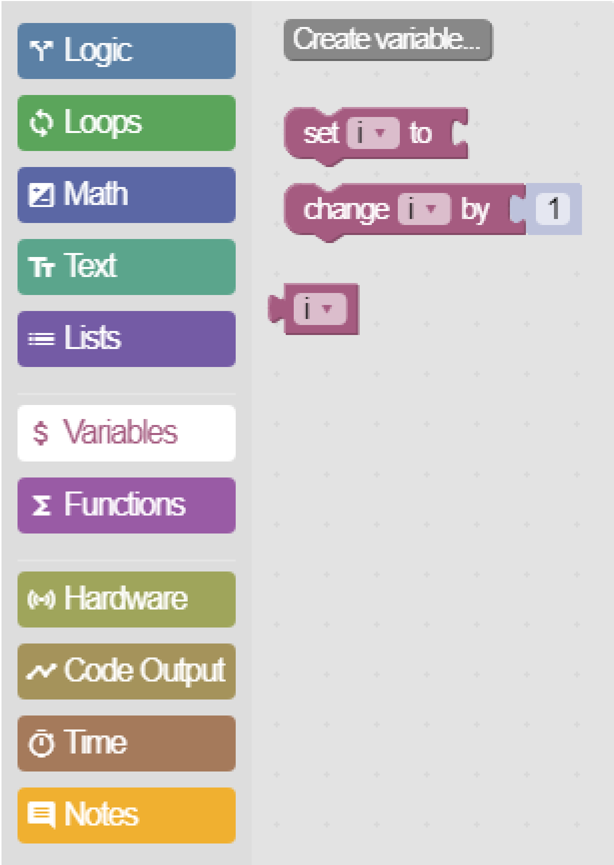
An array is a list within a list. This is useful for passing values to an instruction that has more than one parameter. For example, the //code.Node RGB LED block has three parameters to create a particular color: red (R), green (G), and (B). An array is a convenient method to use if we want to change the color of the LED multiple times.

In the example below, we create four lists containing three values each and place them inside another list. This creates a 3×4 array assigned to the variable `colorArray`. A loop cycles through each row in the array, assigning the row to the `rgbList` variable. Inside the loop, we assign the 1st, 2nd, and 3rd value in the row to the `redValue`, `greenValue`, and `blueValue` variables, respectively. These variables are then used in the RGB LED block to create a color. As a result, the //code.Node RGB LED outputs green, red, blue, and yellow for 0.5 seconds each.



Variables

Use the Variable blocks to store a value that the code can refer to later.


Logic
Loops
Math
Text
Lists
Variables
Functions
Hardware
Code Output
Time
Notes
Create variable...
set i to
change i by 1
i

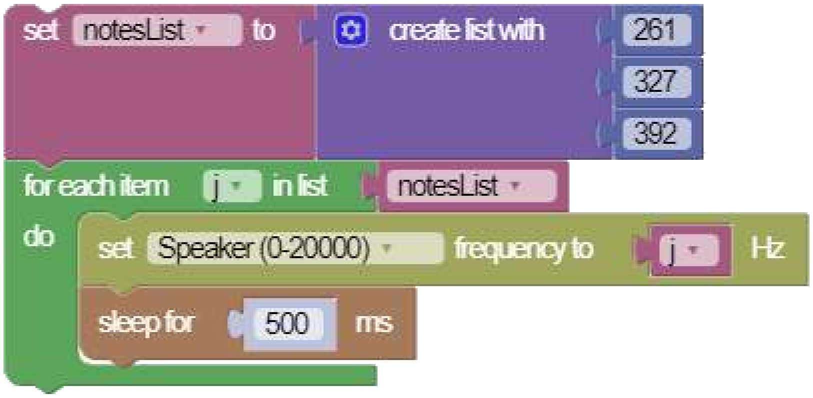
Example 1


set diceValue to random integer from 1 to 6
in text output diceDisplay enter diceValue


diceDisplay
4

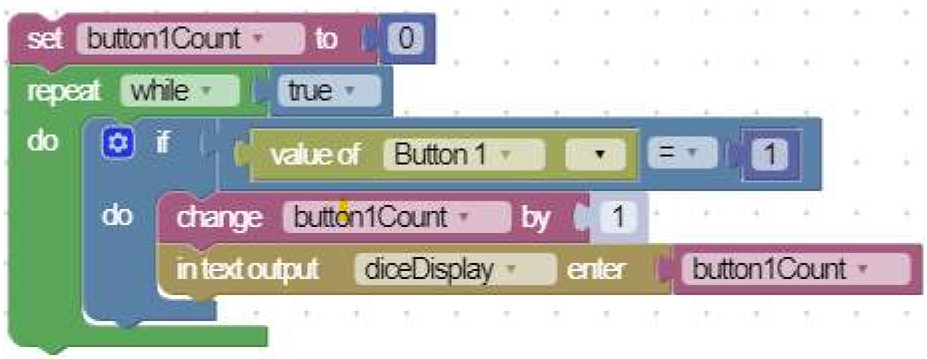
In this example, we create the variable `diceValue` and set it to a random number between `1` and `6`. The value of `diceValue` is then displayed in a text output called `diceDisplay`. A Digits display shows the text output.

Example 2



This example contains two variables. The first variable `notesList` stores a list of numbers. The second variable `j` is defined as the item in each list. The first time the loop circles through, `j` is 261, the second time it's 327 and the third it's 392.

Example 3

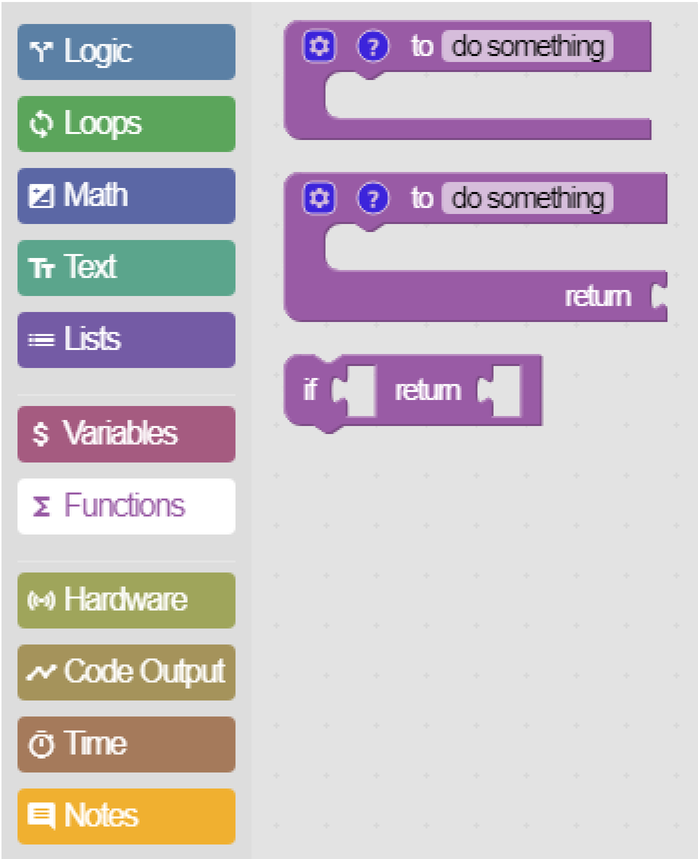


If a variable is a number, you can use the **change by** block to increase or decrease the value of the variable. In this example, the variable `button1Count` increases by 1 each time you press Button 1 on the //code.Node.

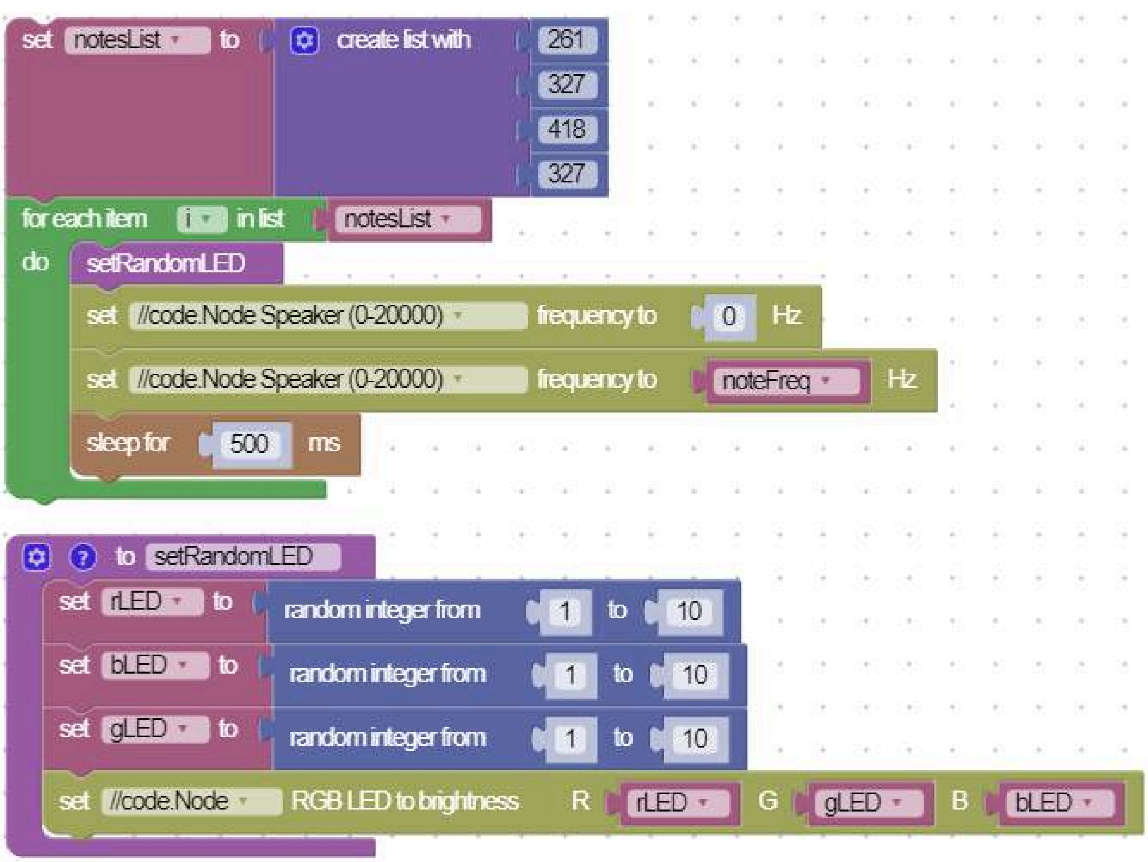
> **Note:**
> Set the value of `button1Count` to 0 at the beginning of the code since Blockly doesn't assume that a variable starts at 0.

Functions

Use the Function blocks to create functions that structure specific tasks.

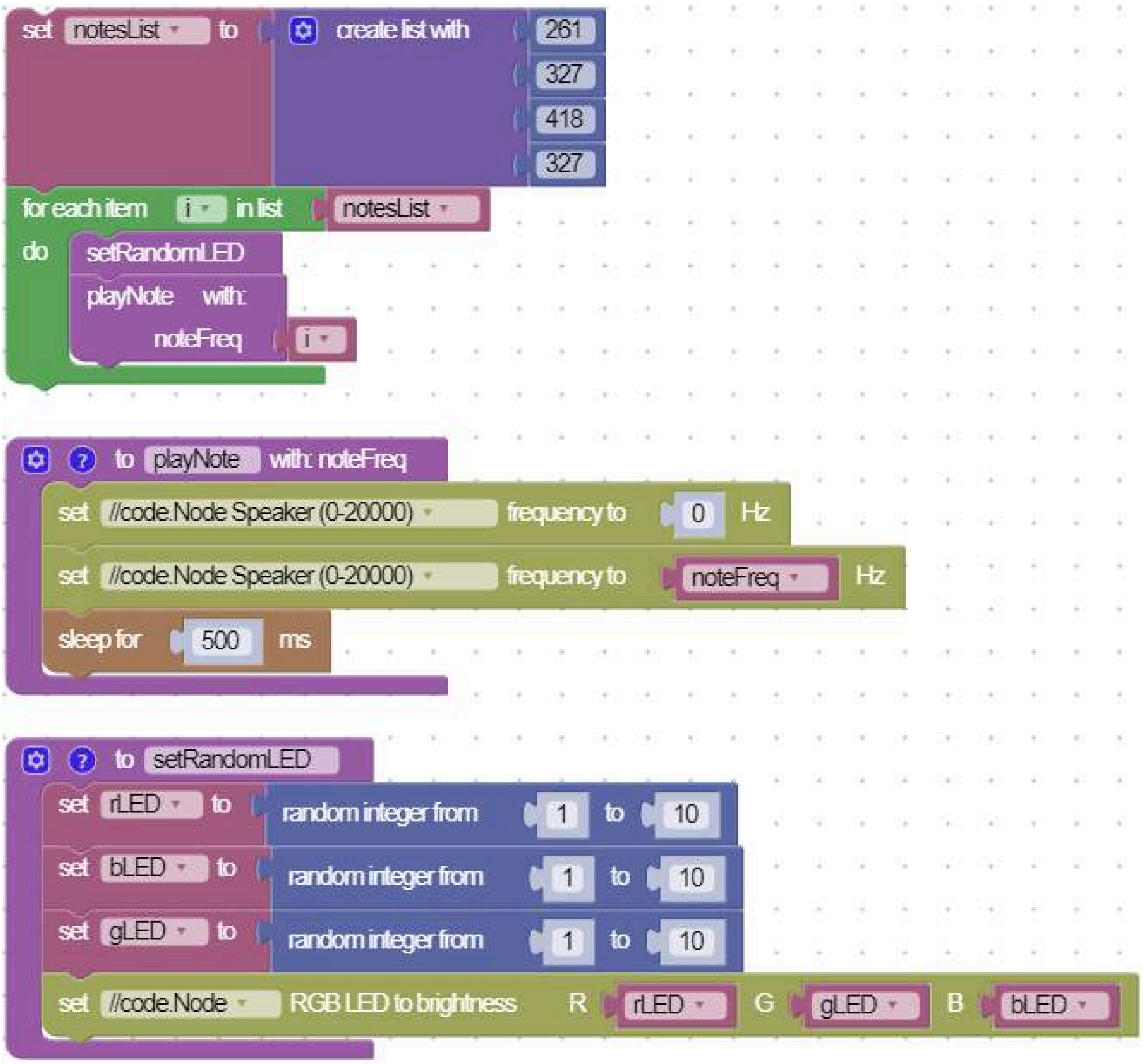


Example 1: Simple function



Programmers use functions to store repeatable tasks and make code easier to read. In this example, a function named `setRandomLED` sets the //code.Node RGB LED to a random color. By putting this code in a separate function, the main code block is easier to read.

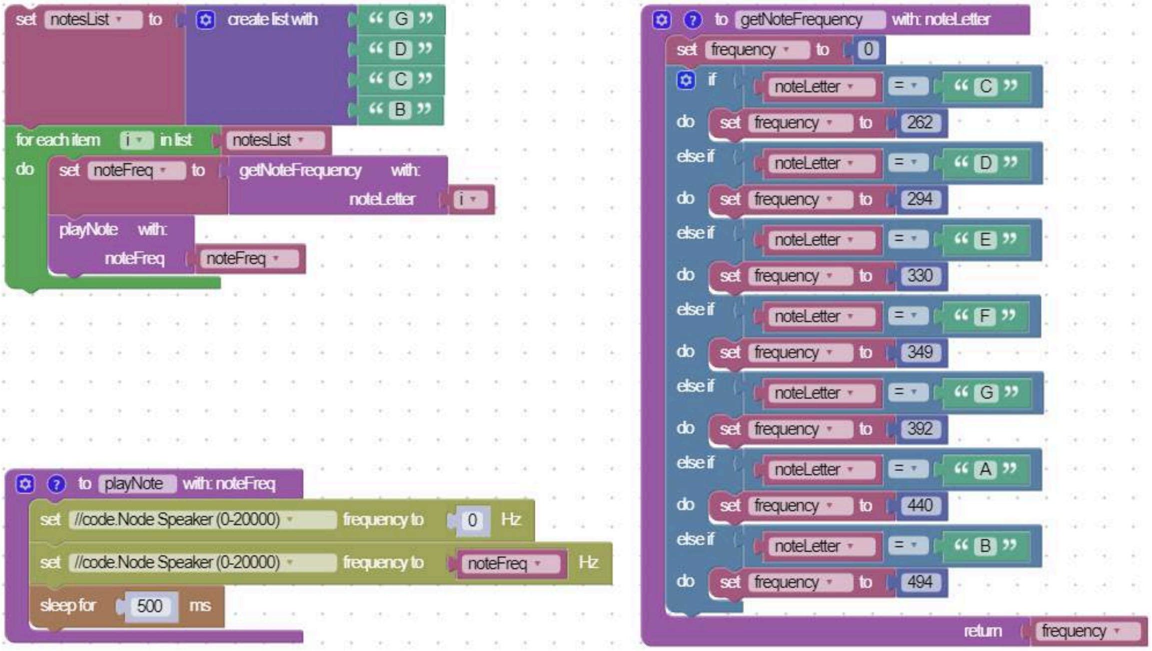
Example 2: Passing variables into a function



Let's take Example 1 a step further by also creating a function to play a musical note. In order to play a note, we need to place a value in the function so the program knows which note to play. To accomplish this, select properties  on the function block and add the **input name** block to the function.

In the loop, the code stores the note frequency in the `i` variable, which gets its value from the `notesList` variable. The function called `playNote` takes an input called `noteFreq`. The code passes the `i` variable into this function. The function turns off the speaker and plays each note in the list for 0.5 seconds.

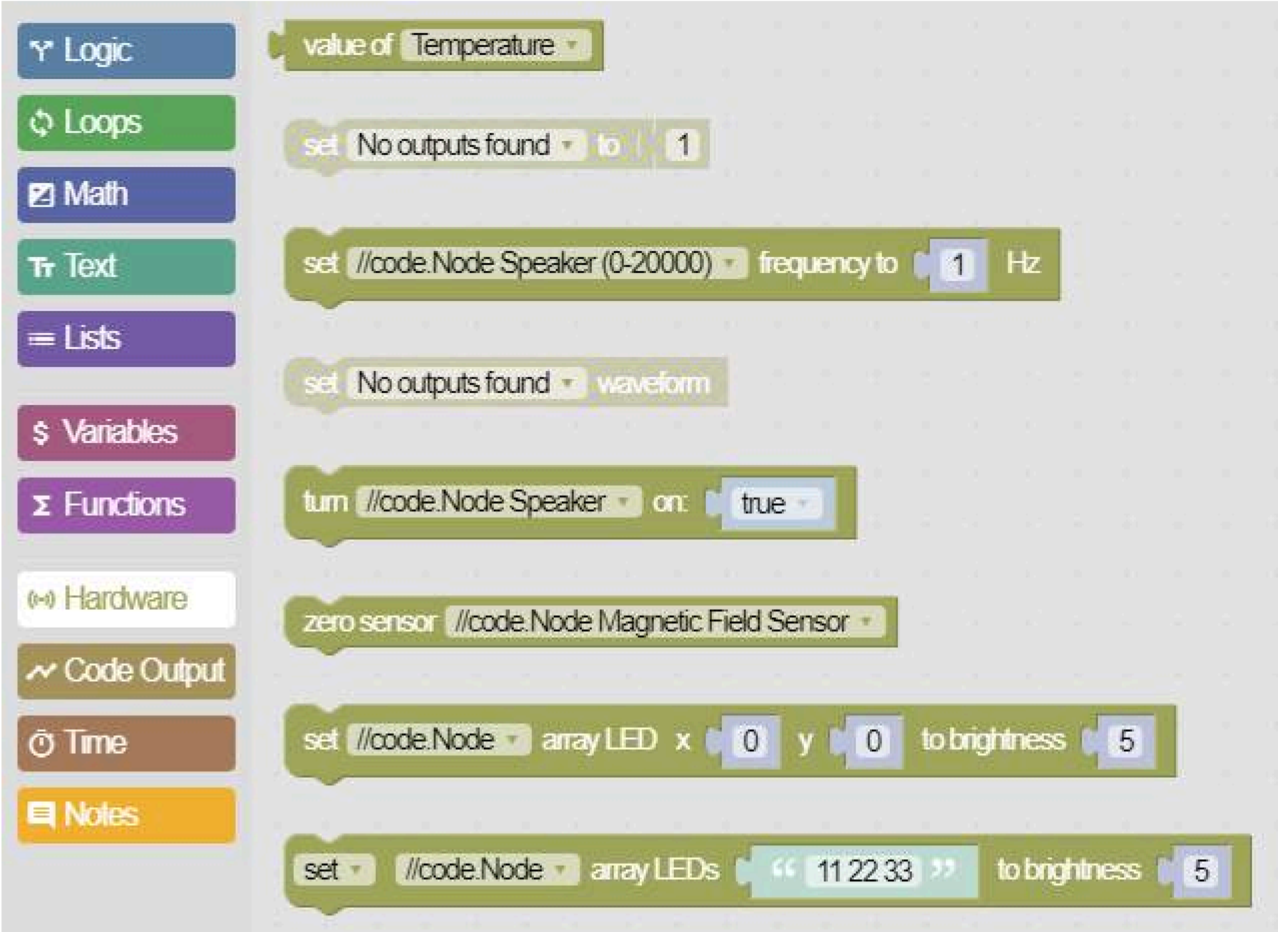
Example 3: Returning values from a function



Functions can also return a value. In this example, a function called `getNoteFrequency` looks up a character `i` from the `notesList` list and returns the note letter's frequency. This value is then set to `noteFreq`, which the code passes to the `playNote` function to play the note.

Hardware

Use the Hardware blocks to interact with PASCO sensors and accessories.



Sensor inputs

Use sensor measurements to control hardware and software outputs.

Read a measurement value

Use the **value of** block to read a measurement from a sensor. In this example, a //code.Node is connected providing the option to select any of its measurements from the list.

Zero a sensor

Use the **zero sensor** block to zero a measurement of a sensor.

//code.Node outputs

Control the output devices on a //code.Node using specialized blocks provided in the Code tool.

Speaker

Two blocks are available to control the speaker on a //code.Node. One block sets the speaker frequency and the other simply turns it on (`true`) or off (`false`). You can also turn off the speaker by setting the frequency to 0.

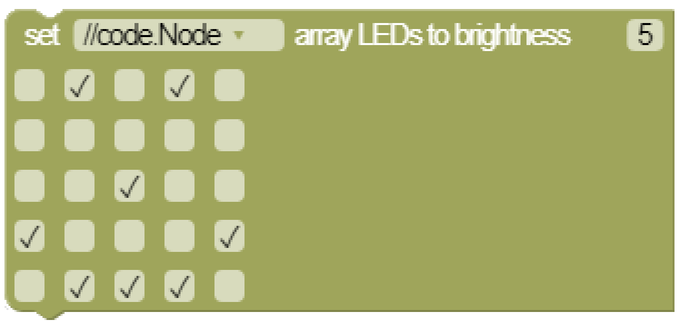
LED Array

Three blocks are available to set the brightness of each LED in of the //code.Node LED array. Set the brightness of each LED on a scale from 0 to 10. Select the LED in the array by its *x* and *y* coordinate. The top left of the array is defined as the origin (0,0). The *x* coordinate corresponds to the horizontal direction and the *y* coordinate corresponds to the vertical direction.

The following block sets the brightness of a single LED on the array.

The next block sets the brightness of a group of LEDs. Define the coordinate of each LED by two consecutive numbers, where the first number is the *x* coordinate and the second number is the *y* coordinate. Separate each LED listed by a space. In the following example, we form a diagonal line in the center of the array from the top left to bottom right.

The third block sets each LED by selecting a box. This example draws a happy face.



RGB LED

Control the color of the //code.Node RGB LED by setting the brightness of the red **R**, green **G**, and blue **B** LEDs.

//control.Node outputs

Control the output devices connected to a //control.Node using specialized blocks.

Speaker

Two blocks are available to control the speaker on a //code.Node. One block sets the speaker frequency and the other simply turns it on (`true`) or off (`false`). You can also turn off the speaker by setting the frequency to 0.

Servo Motors

Use the **set servo** block to control positional or continuous rotation servo motors.

1. ***//control.Node selector***

   If using more than one //control.Node, select which //control.Node you want to control.

2. **Port**

   Select the port the servo is connected to on the //control.Node.

3. **Servo mode**

   Select **angle** for standard servos or **rotate at speed** for continuous rotation servos.

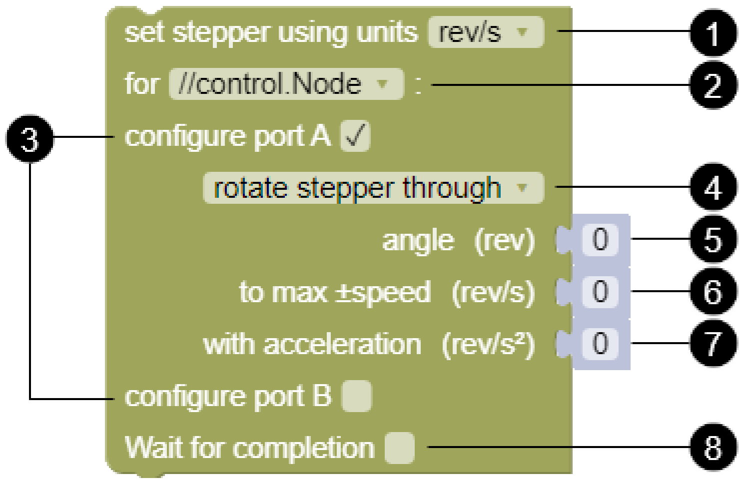
   > **Note:**
   > Use the **turn off power** option if you do not want the servo to consume power. The servo will not hold its position when this option is selected.

4. **Value**

   A positive value rotates the servo counterclockwise and a negative value rotates it clockwise. For **angle**, enter a value between -90 and 90 degrees. For **rotate at speed**, enter a value between -100 and 100%. You can also replace this block with one from the **Math** group.

Stepper Motors

Use the **set stepper** block to control the High Speed or Low Speed Stepper Motor. You can configure the stepper to rotate continuously or for a specified angle.



Note:
The stepper will not rotate if the angle, max speed, or acceleration parameters are set to zero.

1 **Units**

Select which units you want to use for the angle, max speed, and acceleration.

2 **//control.Node selector**

If using more than one //control.Node, select which //control.Node you want to control.

3 **Port**

A single block can be used to configure both Power Out ports simultaneously. Select the box to configure port A, port B, or both.

4 **Rotation mode**

Select one of the following modes:

- **rotate stepper continuously**

  In this mode, the stepper rotates until another block changes the motion. Enter a value for **max speed** to set the rotation speed. Enter a value for **acceleration** to set the rate that the stepper changes its speed from zero to the **max speed**.

- **rotate stepper through**

  In this mode, the stepper rotates for an amount equal to the entered **angle**. Enter a value for **max speed** to set the rotation speed. Enter a value for **acceleration** to set the rate that the stepper accelerates to the **max speed** and slows down to a stop.

- **stop**

  In this mode, the stepper slows down to a stop. Enter a value for the acceleration to set the slow-down rate.

5 **Angle (*through* mode only)**

Enter an angle for the stepper to rotate then stop. If the **Wait for completion** box is selected, the program will not move to the next block until the motor completes its rotation.

6 **Max speed**

Enter the maximum speed the stepper turns. A positive number rotates the counterclockwise and a negative number rotates clockwise.

7 **Acceleration**

Enter the amount the stepper accelerates when it starts rotating. When using **through** mode, this value also sets the deceleration when coming to a stop.

> **Note:**
> High acceleration values may cause jerky movements that make motion less precise. It can also cause the motor to slip and report inaccurate measurements.

8 **Wait for completion (*through* mode only)**

Select this box to allow the stepper to complete its rotation before moving to the next block.

The following table lists the allowed range for each measurement by unit. A value entered less than the minimum value will output the minimum value. A value entered larger than the maximum value will output the maximum value.

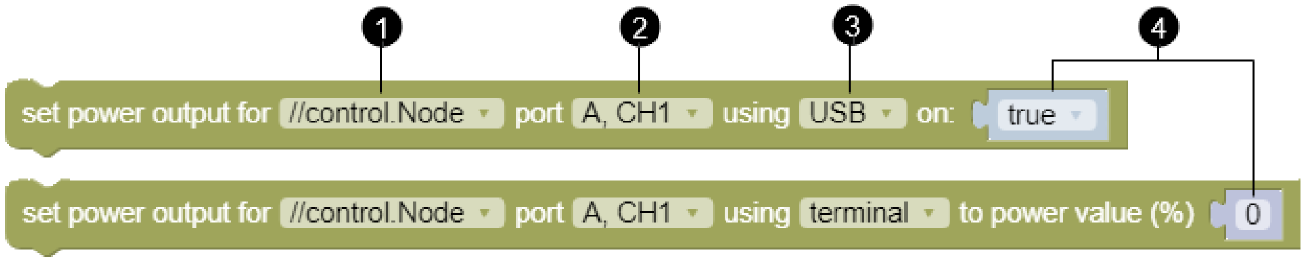
	rev/s		rpm		°/s		rad/s	
	Min	Max	Min	Max	Min	Max	Min	Max
Angle	0	68.26	0	68.26	0	24570	0	428.8
Max speed	-2	2	-120	120	-720	720	-12.6	12.6
Acceleration	0	6.826	0	409.6	0	2457	0	42.88

Grow Light

Use the **set grow light** block to control the brightness of the blue and red LEDs on the Grow Light. Enter a value between 0 and 10 to set the brightness of the red (R) and blue (B) LEDs.

Power Output Board

Use the set power output block to control devices connected to the Power Output Board. The Power Output Board can control any device that can be powered up to 5 volts with 1 amp of current. The board includes two channels that allows you to connect and control two devices simultaneously. You can connect a device to each channel using either the USB port or the terminals. The power output can be varied when using the terminal connection. Power is provided using pulse width modulation (PWM).



1 **//control.Node selector**

If using more than one //control.Node, select which //control.Node you want to control.

2 **Port**

Select the Power Out port the Power Output Board (A or B) is connected to and the Power Output Board channel the device is connected to (CH1 or CH2).

3 **Connection**

Select whether the device is connected to **USB** or to the **terminal**.

> **Important:**
> Make sure that **terminal** isn't selected when a USB device is connected. The USB device might get damaged if a power value is set below 100%.

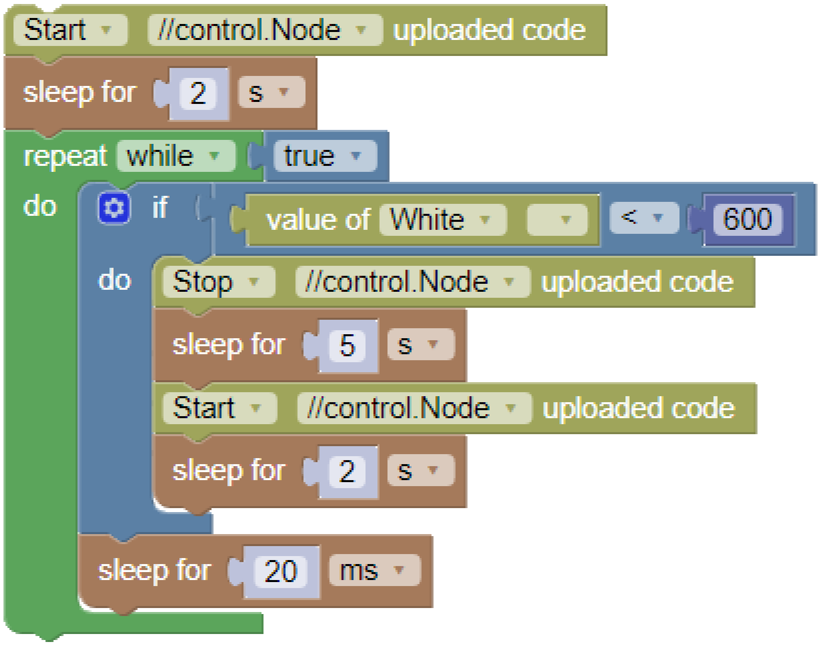
4 **Value**

- **USB**: Select `true` to have the power on or `false` to have the power off. You can also replace this block with one from the **Logic** group.
- **terminal**: Enter a power value between -100 and +100%. The percentage indicates the duty cycle of the PWM signal. A lower percentage corresponds with a lower average voltage. You can also replace this block with one from the **Math** group.

Run uploaded code

The **uploaded code** block can be used to `start` or `stop` code uploaded to the //control.Node. In order to use this block, the code in the Code tool must be different than the code uploaded to the //control.Node.

The following example code uses this block to start and stop uploaded code using a Wireless Light Sensor measurement. The uploaded code starts when you click **Start** . When the sensor detects darkness (set at a value less than `600`), the uploaded code stops running for 5 seconds then starts again. When you click **Stop** , the uploaded code also stops.



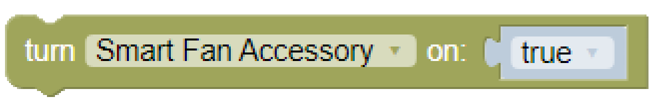
Smart Cart accessory outputs

Create a program that controls the thrust of a Smart Fan Accessory, the speed of a Smart Cart Motor, or launches a projectile from the Smart Ballistic Cart Accessory.

Smart Fan Accessory

Use the **set output** block to set the thrust of the Smart Fan Accessory between -100 and +100. You can either use the default number block to set the thrust or replace this block with a block from the Math group.

Use the **turn on** block to turn the fan on or off. Set the parameter to `true` to turn on the fan or to `false` to turn off the fan. Use a block in the **Logic** group in place of the true/false block to create a logical expression to turn the fan on or off.



Smart Cart Motor

Use the **set output** block to set the power of the Smart Cart Motor between -100 and +100. You can either use the default number block to set the power or replace this block with a block from the Math group.

Use the **turn on** block to turn the motor on or off. Set the parameter to `true` to turn on the motor or to `false` to turn off the motor. Use a block in the **Logic** group in place of the true/false block to create a logical expression to turn the motor on or off.

Smart Ballistic Cart Accessory

Launch the projectile from the Smart Ballistic Cart Accessory using the **release projectile** block.

Signal generator outputs

Control the output voltage, frequency, and waveform of a signal generator using Blockly code.

The blocks described on this page can be used together to control the signal generator on the Wireless AC/DC Module and the 550 Universal Interface.

Set output value

Use this block to set the output voltage of a signal generator. The default number block can be replaced with a block from the **Math** group. In this example, the voltage of the AC/DC Module is set to 1 volt.

Set output frequency

Use this block to set the frequency of a signal generator. The default number block can be replaced with a block from the **Math** group.

Set output waveform

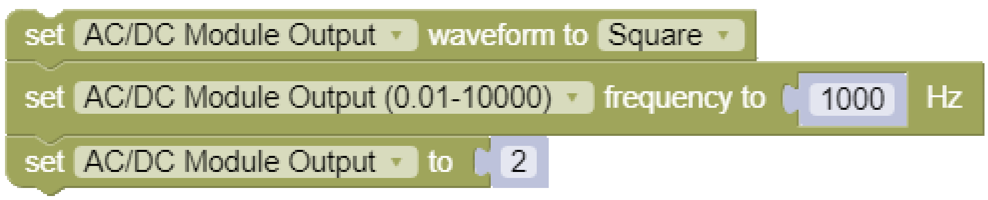
Use this block to set the waveform of a signal generator. In this example, the waveform of the AC/DC Module is set to a square wave.

Turn output on or off

Use this block to turn a signal generator on (`true`) or off (`false`). The default true/false block can be replaced with a block from the **Logic** group.

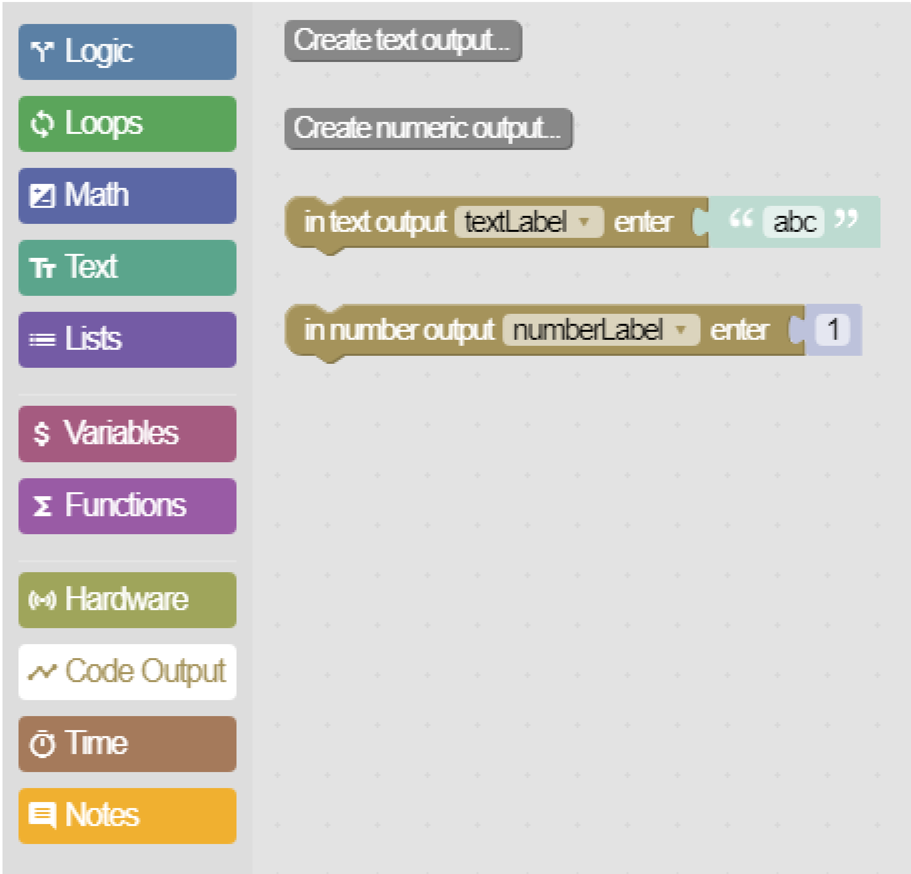
Example: Multiple blocks used together

In this example, the Wireless AC/DC Module outputs a square wave with an amplitude of 2 V and a frequency of 1000 Hz.



Code Output

Use the Code Output blocks to provide text or numeric output in a Digits or Table display.

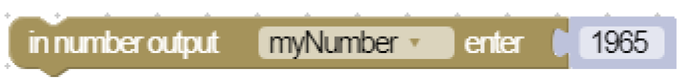


Text Output

This block outputs a string of text.

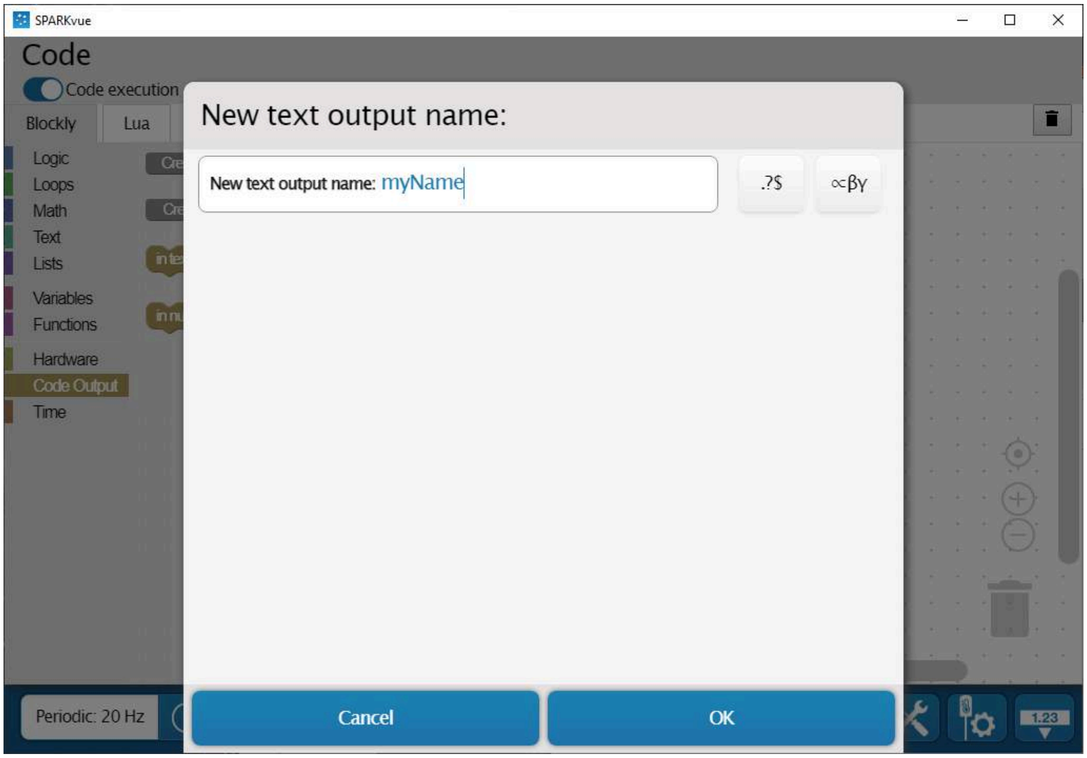
Numeric Output

This block outputs a numeric value, such as a sensor measurement.

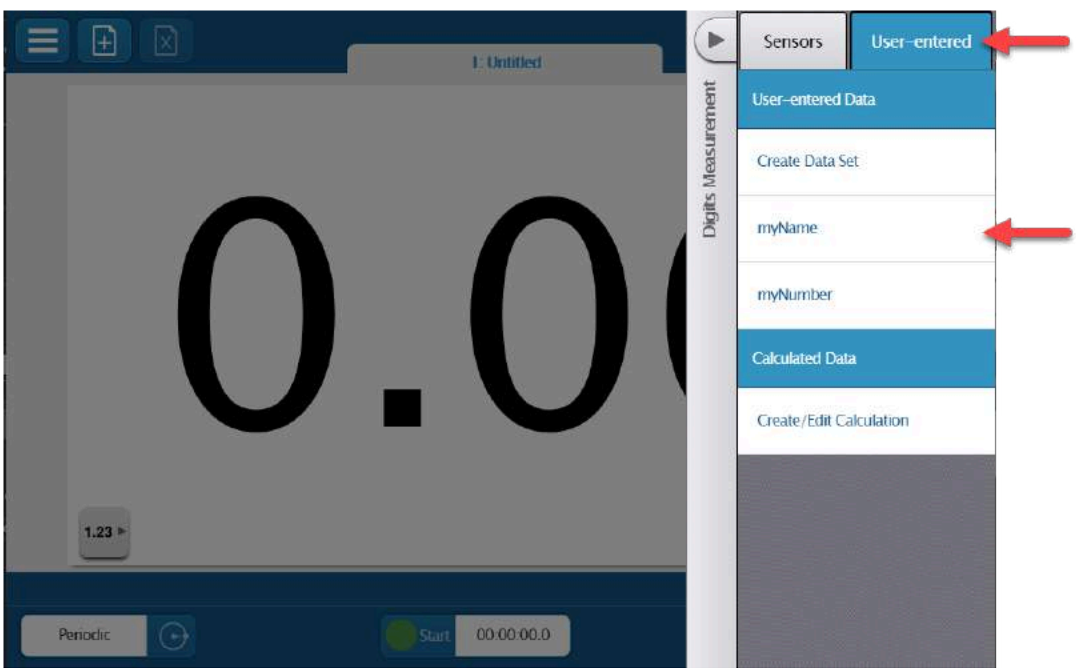


Create a Text Output

1. Click the **Create text output** button, enter a name for the output, and click **OK**.

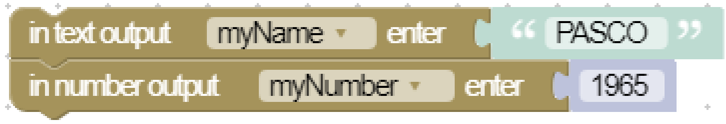


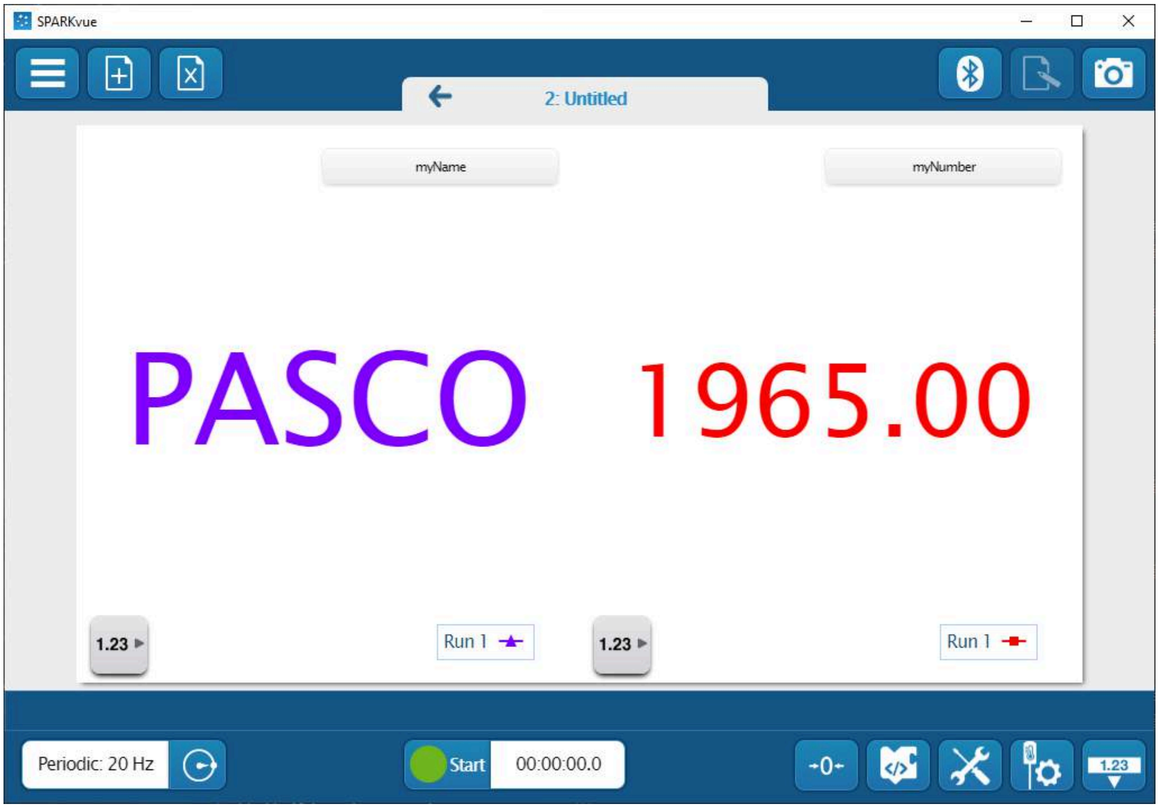
2. Close the Code tool.
3. Click , select a template, then select the Digits display 1.23.
4. Click **Select Measurement**, select the **User-entered** tab, then select the name of the text output.



Example 1: Displaying Text Versus Numbers

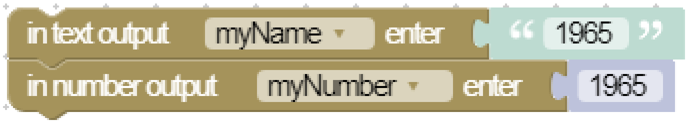
This example contains two Digits displays with one showing text output and the other shown numeric output.

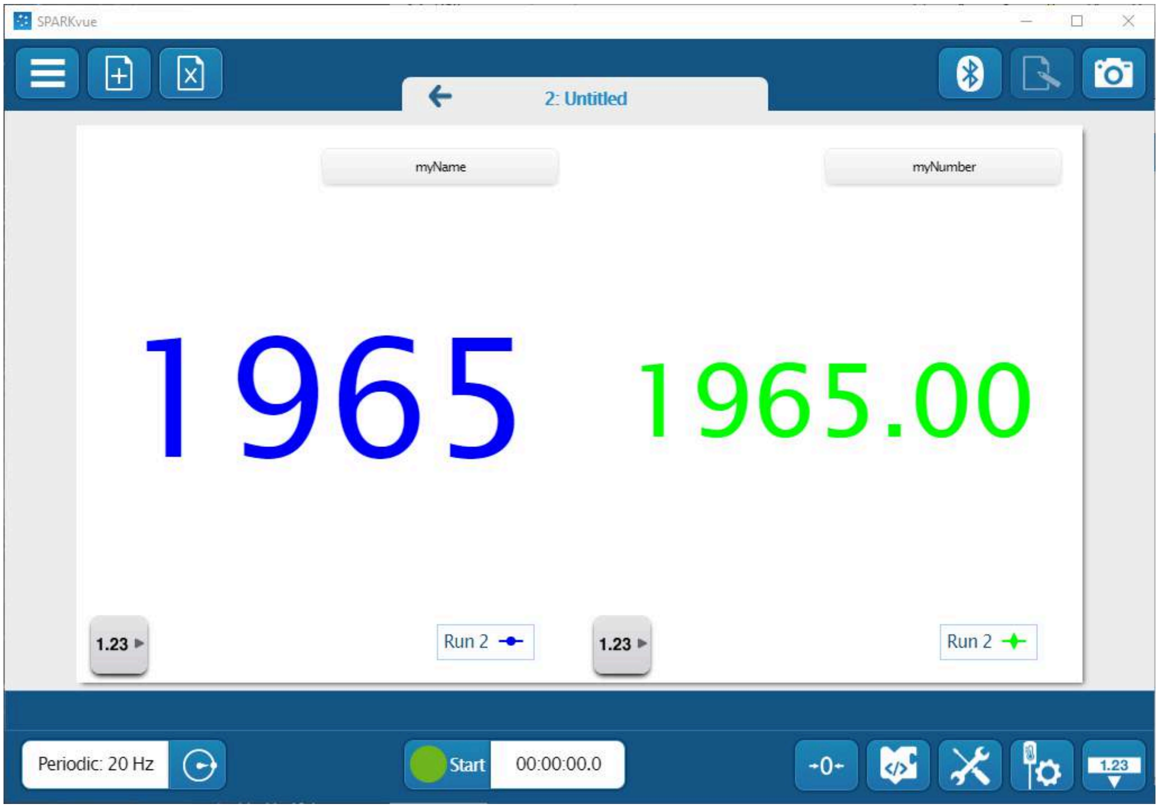




Example 2: Displaying Numbers in Text Output

It's possible to display numbers in text output. However, the code interprets numbers as a string of text instead of a quantity of measurement, which means that the code can't apply mathematical operations on the output. When showing a number, you can see the difference between a text and number display.

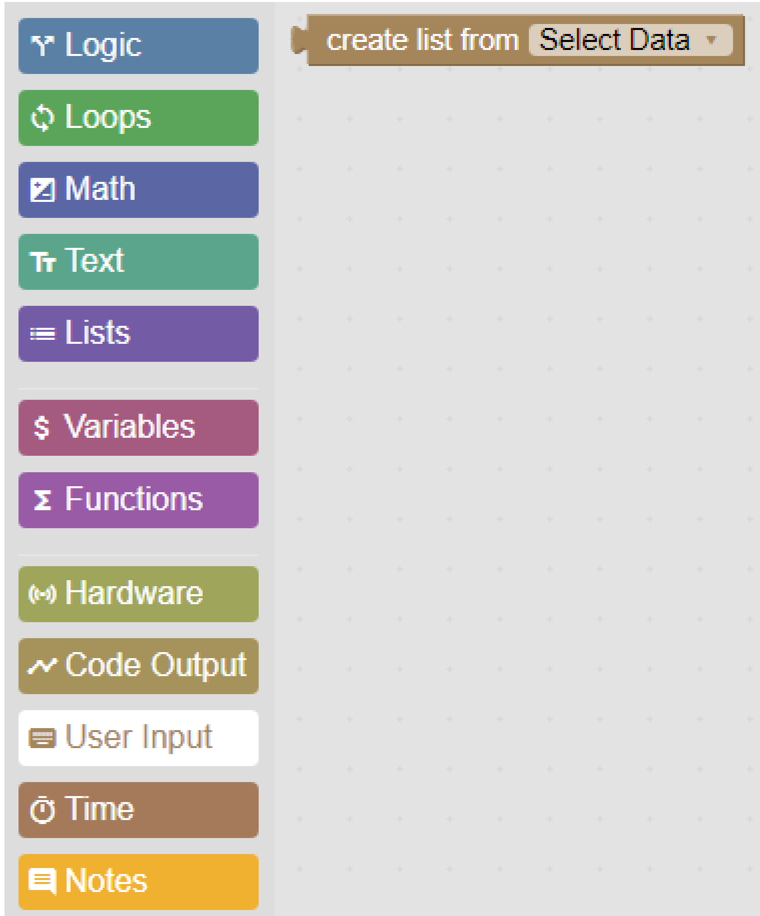




User Input blocks

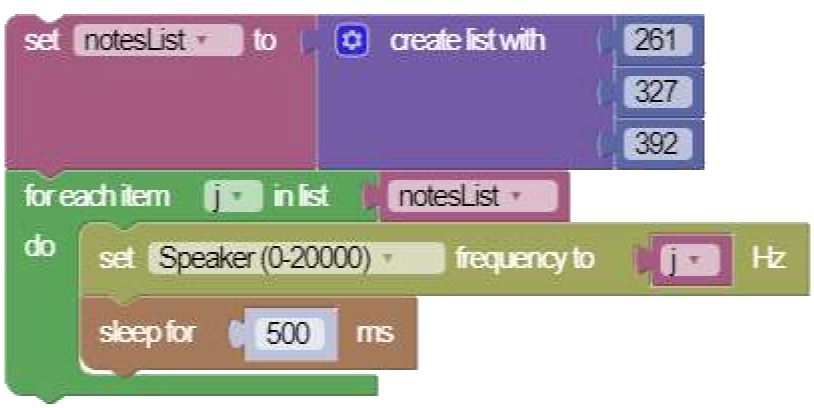
Use the User Input blocks to use user entered data in a Blockly program.

Note:
This feature is not supported for code that is uploaded to a //control.Node. The code must be run from SPARKvue.

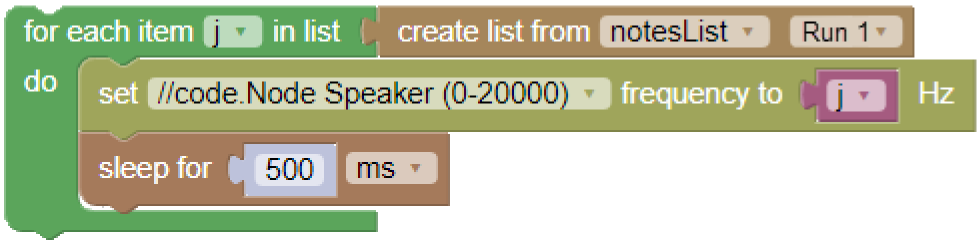


Create a list from a Table display

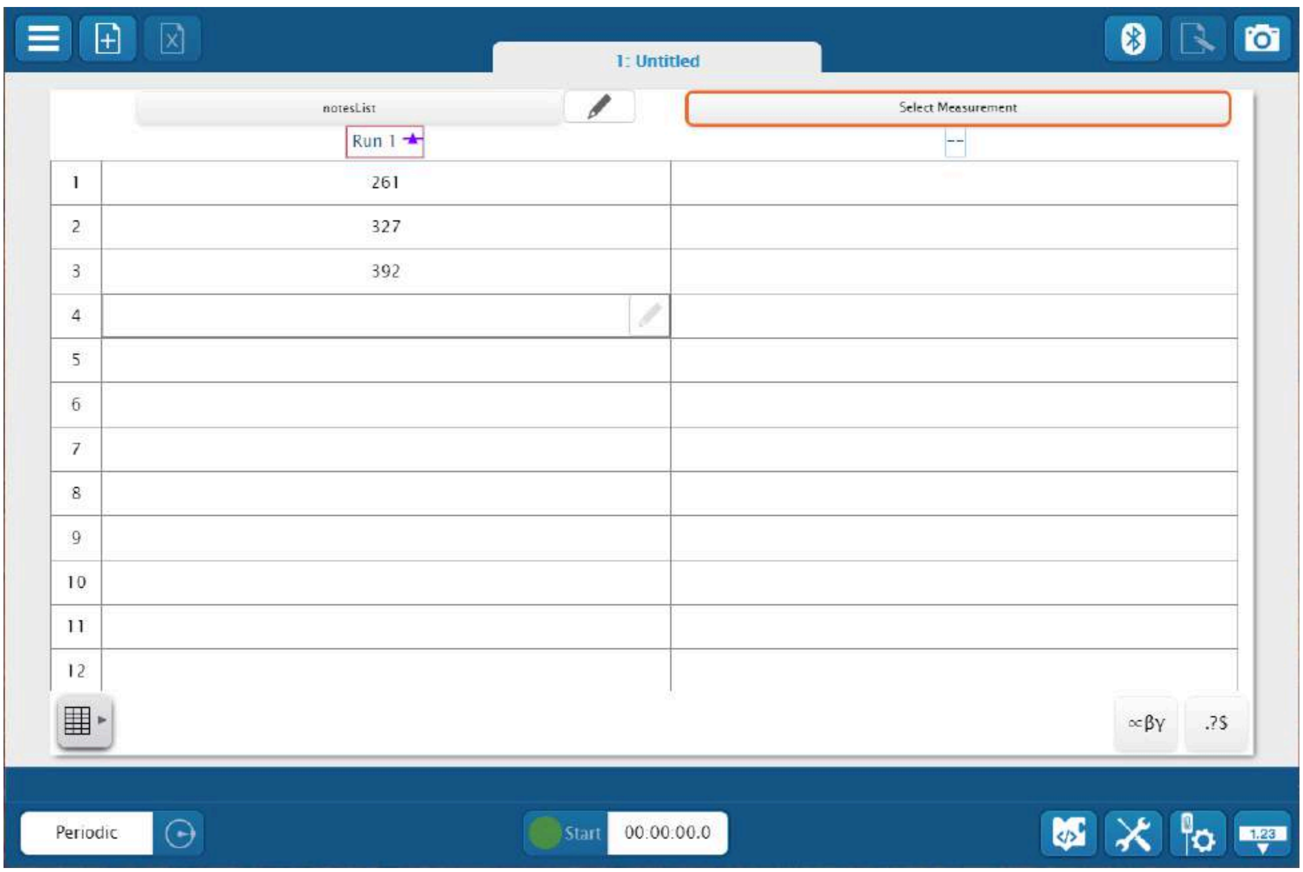
Instead of creating a List of values using a block from the **Lists** category, you can create a list by entering values in a **Table** display. Using a table is convenient since you don't need to modify code to add or remove items from a list. Consider the following example:



In this example, the program sets the speaker to output each frequency in the list block for 500 ms. You can accomplish the same thing by using the **create list** block in the **User Input** category:



Instead of entering the values in a list block, you enter the values in a **Table** display:



To create a user-entered data table:

1. Build a new page and select a Table for the display.
2. Click **Select Measurement**.
3. Select the **User-entered** tab then click **Create Data Set**.
4. Enter a **Measurement Name** then click **OK**.
5. Enter data in the column.

Once you have created a user-entered data measurement, you can then add a **create list** block to the Blockly workspace.

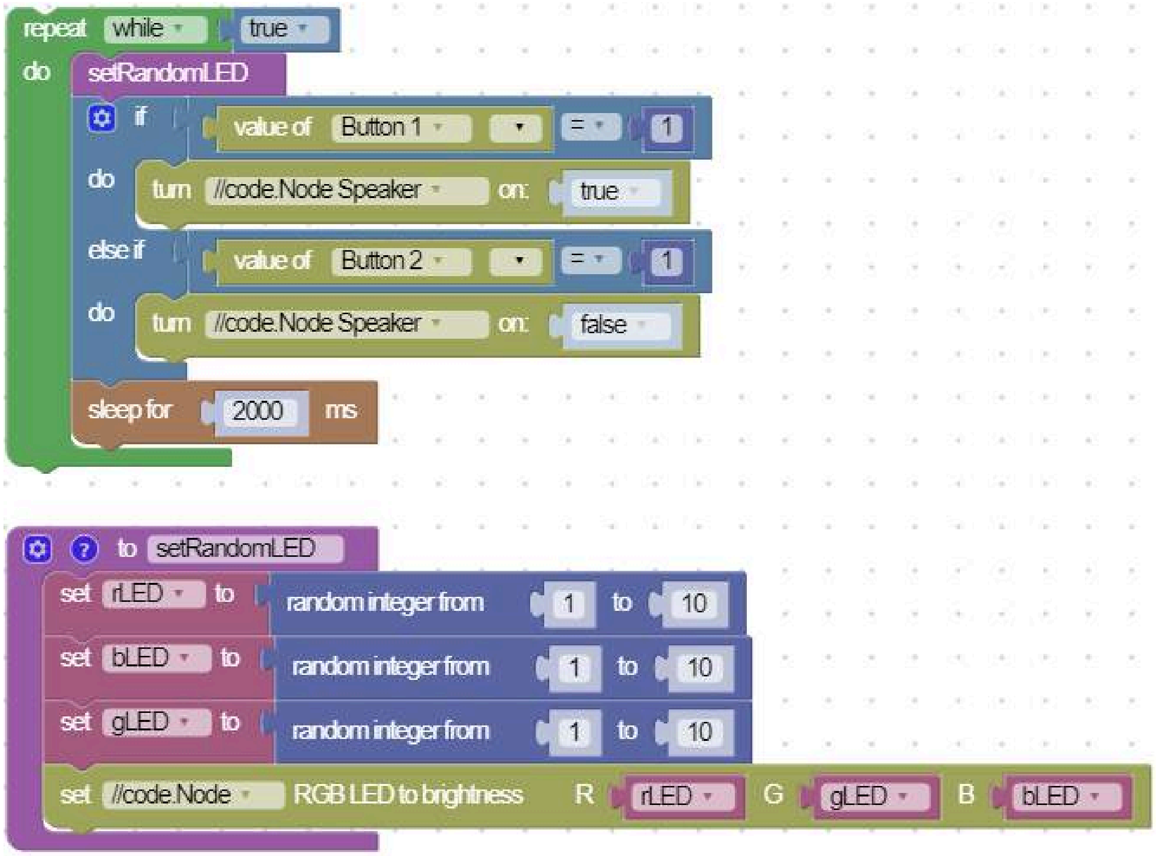
Time

Use the Time blocks to pause code or get a timestamp from when the program was started.

Example 1: Using Sleep Blocks

Sleep blocks pause the code for a designated amount of time. In this example, the code turns on the red LED on for one second, then turns on the blue LED for a half second, and finally turns on the green LED for two seconds. Note that when a program is sleeping it's not performing any other task.

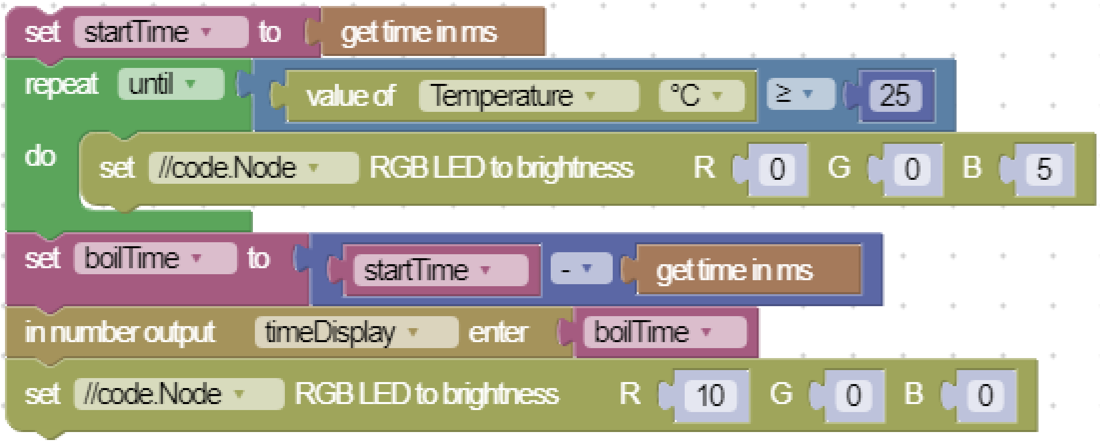
Example 2: Sleep Block Causing an Error



Say you want to change the //code.Node RGB LED color every two seconds and also turn the //code.Node speaker on or off by using the //code.Node buttons. In this example, if you press Button 1, the speaker may or may not turn on. This is because the code only checks the button state *after* every two seconds. If the button isn't pressed down during that check then the code doesn't turn on the speaker. To get the speaker to turn on, you must press and *hold* Button 1 so that when the LED changes the state of Button 1 state is `true`. To turn the speaker off, press and hold Button 2 so that it's held down the same time the LED changes.

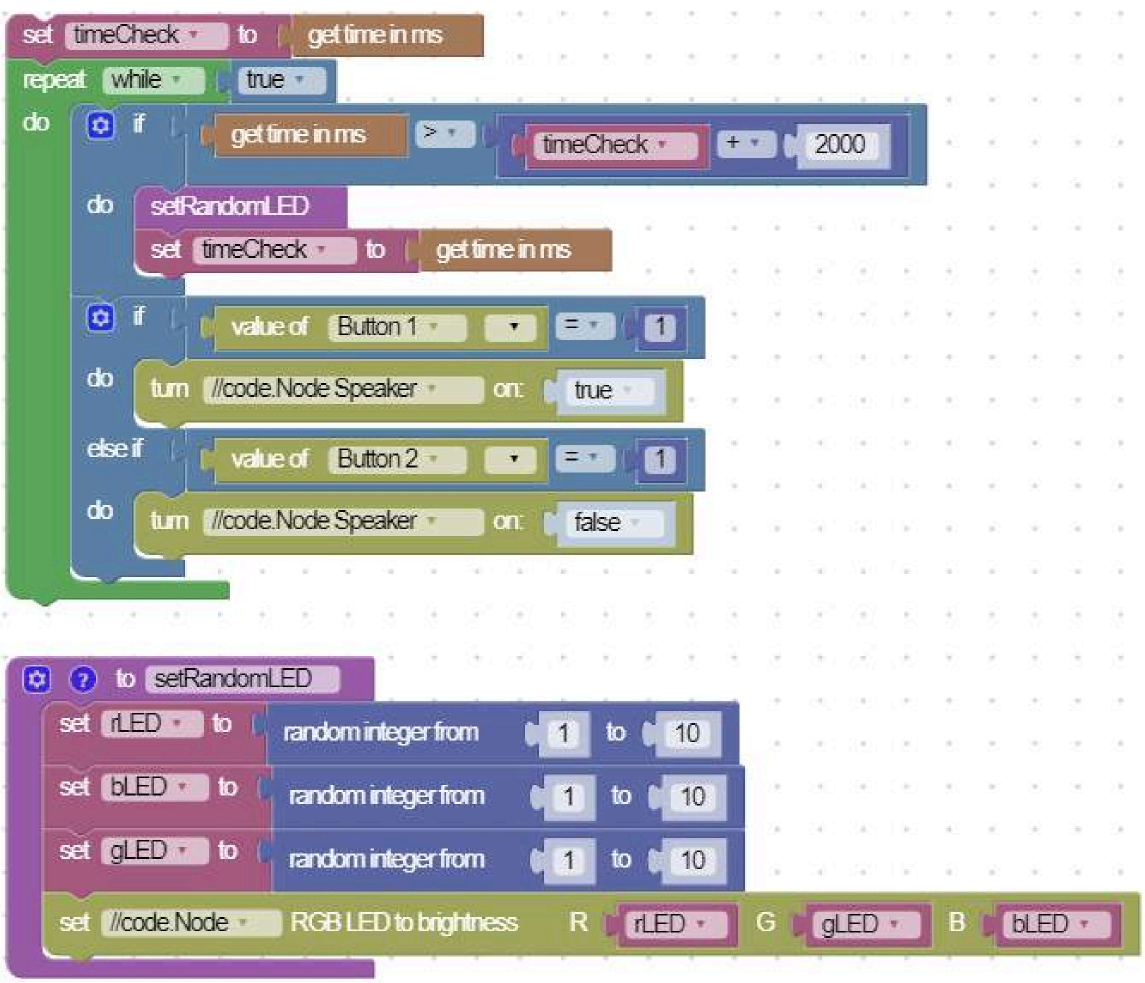
> **Tip:**
> See Example 4 for a method to fix this problem.

Example 3: Getting the Instantaneous Time



Use the **get time in ms** block to get a timestamp of when an event happens. In this example, the code uses this block to determine how long it takes for the temperature to reach 25 °C. Use the **number output** block from the Code Output group to display the time.

Example 4: Perform Synchronous Functions

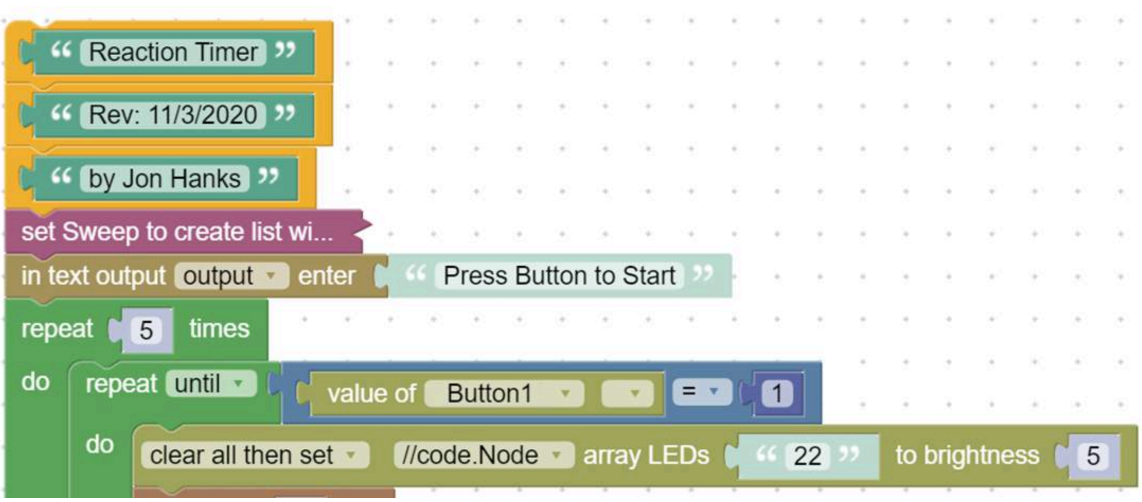


Use the **get time in ms** block if you want tasks to happen every *x* amount of seconds. In this example, set a variable `timeCheck` to the time stamp at the start of the program. Then use an if statement with a condition that sets `timeCheck` to the current time stamp every two seconds and runs the `setRandomLED` function. At the same time, use a separate conditional statement to make the //code.Node speaker turn on by pressing Button 1 or turn off by pressing Button 2. The loop doesn't have to stop before turning the speaker on or off. This doesn't work with the sleep blocks as see in Example 2.

Notes

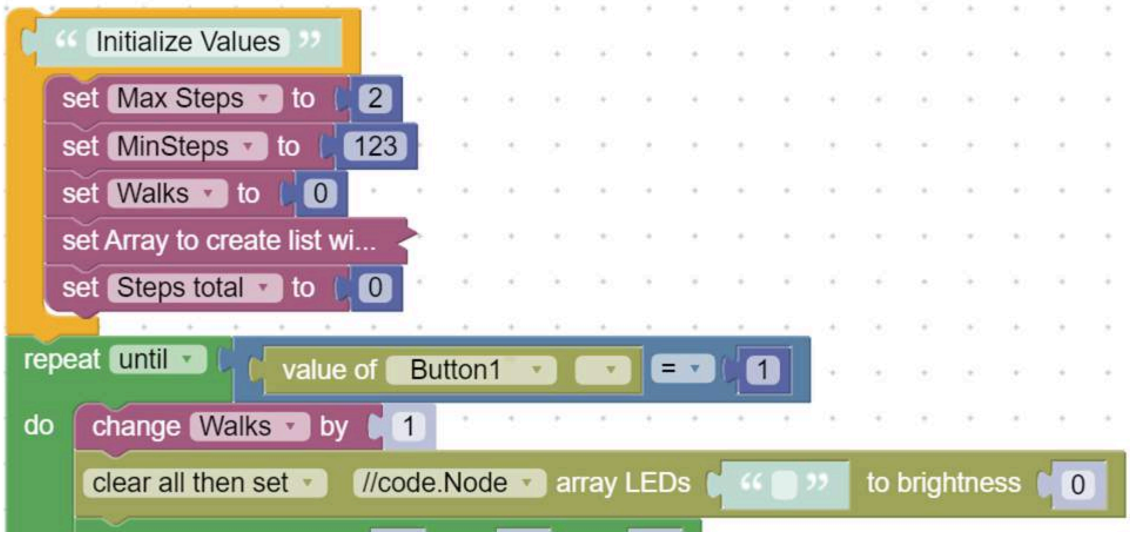
Use the Notes blocks to add comments to your code that the program ignores. Insert a single note between blocks or associate a note with a group of blocks.

Single Line Notes



Single line notes are useful for adding comments to your code to help others understand how the code works. These blocks are also useful for adding metadata to your code such as the author name and revision date. You can add a single line note preceding or following a block.

Block Group Notes



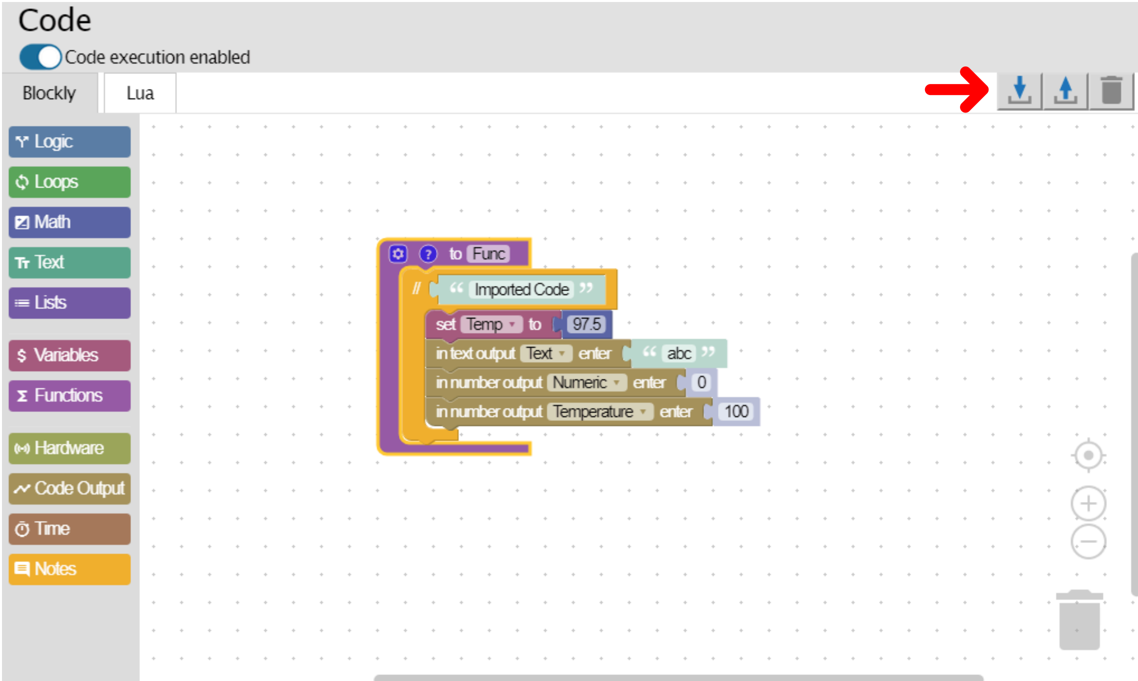
Block Group notes work like single line notes, except that you can associate multiple blocks with a note. This is useful for moving a group of blocks by dragging the note block. Additionally, you can disable a group of blocks by disabling the note block.

Import and Export Code

Import and export Blockly code to share between other SPARKvue or Capstone files.

Share a program created in the Code tool with another file by using the import and export tools at the top of the workspace. The Blockly code exports as a .pcbx file which you can import into a SPARKvue or Capstone file.

> **Note:**
> This feature isn't supported on iOS or Android devices.



Export Code

1. Click the export tool at the top of the Blockly workspace.
2. Select the location where you want to save the file then click **Save**.

Note:
it's not possible to export the following types of measurements:

- User-Entered Data
- Calculation Data
- Timer Data from Photogates

Import Code

You can import code to a new file or to a file which already contains existing code. If the file contains existing code, the imported code is added to the workspace as a separate group of code.

1. Click the import tool at the top of the Blockly workspace.
2. Navigate to the location of the .pcbx file you want to import and click **Open**.

Note:
If you are importing code that contains a block with a name identical to a block in the existing file, the imported block is renamed.

PASCO Code Library

Import a program that contains code you can edit and build upon.

The PASCO Code Library contains sample programs that work with the //code.Node and //control.Node. The PASCO Code Library button is visible only when you connect a device.

Import a program

1. Click the PASCO Code Library at the top of the Blockly workspace.
2. Select a category from the menu.
3. Select a program from the list. A description of the program is provided at the bottom of the window.
4. Click **OK** to import the program.

When you open the program, the functions and other blocks might be collapsed to avoid cluttering the workspace. If you want to explore how the program works or make modifications, you can expand the blocks to see the details.

//code.Node programs

Import programs designed for use with the //code.Node.

Scroll Text

This program contains a function that can scroll text in the //code.Node's 5x5 LED array. The function takes two parameters:

- **Text**: Enter characters in the text block to scroll across the array. You can enter any letter or number and the following symbols: `.`, `/`, `+`, `-`, `=`, and `!`.
- **Number of Times**: Enter a value for the number of times the text scrolls across the array.

Play Music

This program contains a function that plays a song on the //code.Node's speaker. The function takes two parameters:

- **Tempo**: Enter a number to control how fast the song plays. A higher number corresponds with a faster tempo.
- **Song**: This contains a list of text blocks to enter the song notes. Each text block can represent a measure in a song.

To create a song, enter notes as letters in the text blocks of the Song list. Rests are represented by forward slashes `/`. You can modify the notes and rests in the following ways:

- **Sharps and flats**

  Enter a `#` or `b` following the note. For example, `A#` represents A-sharp and `Ab` represents A-flat.

- **Octaves**

  Enter a number before the note to set the note octave. For example, `4A` plays the note A_4, which is typically 440 Hz. This also sets all following notes to the same octave until you set a new octave.

- **Duration**

  Enter a number after a note or rest to set the duration it's played. The number represents the inverse of the duration. A `2` represents half, a `4` represents a quarter, and so forth. For example, `A4` plays an A as a quarter note. This also sets all following notes and rests at the same duration until you set a new duration.

- **Dotted notes and rests**

  A dot increases the duration by half. Enter `*` after the note or rest, such as `A*`.

- **Ties**

  Tie a note between measures by entering a `~` after the note. In the following example, a G note is tied between two measures to play for a duration equal to a whole note plus a half note. If the `~` was missing, this code would play two separate G notes.

//code.Node Invaders

This program provides code for an arcade game displayed on the LED array. The game's objective is to eliminate alien invaders before they reach the bottom of the array. Use the accelerometer to move the cannon by tilting left and right, and press button 2 to fire the laser. When the game is over, the array displays the number of targets eliminated.

Repeat Pattern

This program provides code for a memory game displayed on the LED array. The program displays a pattern by flashing lines on the left and right sides of the array. The objective is to repeat the pattern by pressing buttons 1 and 2, corresponding to the array's left and right sides. The game increases in difficulty with each successful match by adding one additional item to the pattern. When the game is over, the array displays the total number of correct patterns.

PASCObot programs

Import programs designed for use with the PASCObot.

> **Note:**
> For the PASCObot functions to work correctly, connect the stepper motor on the right side of the bot to Power Out port A on the //control.Node.

Move a Distance

This program includes a function that moves the PASCObot a specified distance. The function includes a parameter to enter the distance in centimeters. Enter a positive number to move the bot forward or a negative number to move the bot backward.

Move with Velocity

This program includes a function that moves the PASCObot continuously at a specified velocity. The function includes a parameter to enter the velocity in cm/s. Enter a positive number to move the bot forward or a negative number to move the bot backward.

Turn Left or Right

This program includes two functions that turn the PASCObot 90° left or 90° right. To use the functions, just include the function block in your code.

Rotate

This program includes a function that turns the PASCObot at a specified angle. The function includes a parameter to enter the angle in degrees. Enter a positive number to turn the bot right or a negative number to turn the bot left.

Follow Line

This program includes a function that makes the PASCObot follow a black or white line when used with the Line Follower module. The function includes two parameters:

- **Line intensity (%)**

  This parameter indicates the approximate maximum intensity measured by the Line Follower module when placed over the line.

- **Surface intensity (%)**

  This parameter indicates the approximate maximum intensity measured by the Line Follower module when placed over the surface it's moving on, such as a floor or table.

Before using the function, record some sample measurements of the line and surface intensity to determine what values to enter in the parameters. You should disable code execution so that the program doesn't run when measuring the intensities.

Maintain Range

This program includes a function that uses the Range Finder module. Use this function to move the bot within a specified distance of an object in front of it and maintain that distance. For example, if you place an object in front of the bot, the bot will move towards the object until the distance is equal to the value entered in the function's parameter. If you move the object closer to the bot, the bot will move backwards to maintain the distance specified in the parameter.

Control Gripper

This program includes four functions that use the PASCObot Gripper accessory with two Servo Motors. Use the functions to lower, raise, open, and close the Gripper arms. Each function includes a variable that sets the servo angle. If needed, you can change the angles to values that work best with your setup.

Upload code to the //control.Node

Upload code to the //control.Node so that it can run independently from a computing device.

Uploading code is sometimes necessary for programs that require smooth operation since Bluetooth can experience small delays when sending code instructions.

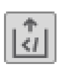
> **Note:**
> It is not possible to download uploaded code from the //control.Node. Be sure to save all programs in a SPARKvue or Capstone file to prevent losing your code.

Upload toolbar

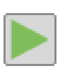
Use the toolbar located at the top right of the **Code** tool to upload, run, and delete code from the //control.Node:

1. **Select //control.Node**

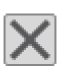
   Use this tool to select which //control.Node you want to manage when more than one //control.Node is connected.

2. **Upload code** 

   Click to upload code to the //control.Node. The //control.Node beeps twice to confirm that the code has been uploaded.

3. **Run code** 

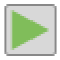
   Click to run the code uploaded to the //control.Node. Click again to stop the code from running.

4. **Delete code** 

   Click to delete the code previously uploaded to the //control.Node. The //control.Node beeps twice to confirm that the code has been deleted.

Running uploaded code

There are three ways to start and stop uploaded code:

- Click Run Code  in the **Code** tool.
- Press the //control.Node power button.
- Use the **uploaded code** block in a program:

Reading sensor data

While the //control.Node is running uploaded code, it can send sensor data back to the software. You can also disconnect and connect the //control.Node to the software while it is running uploaded code without interruption. This can be useful if you want to record data from the //control.Node on multiple devices.

> **Note:**
> Before recording data, make sure to disable code execution in the **Code** tool to avoid code in the software interfering with the uploaded code.