SPARKvue®

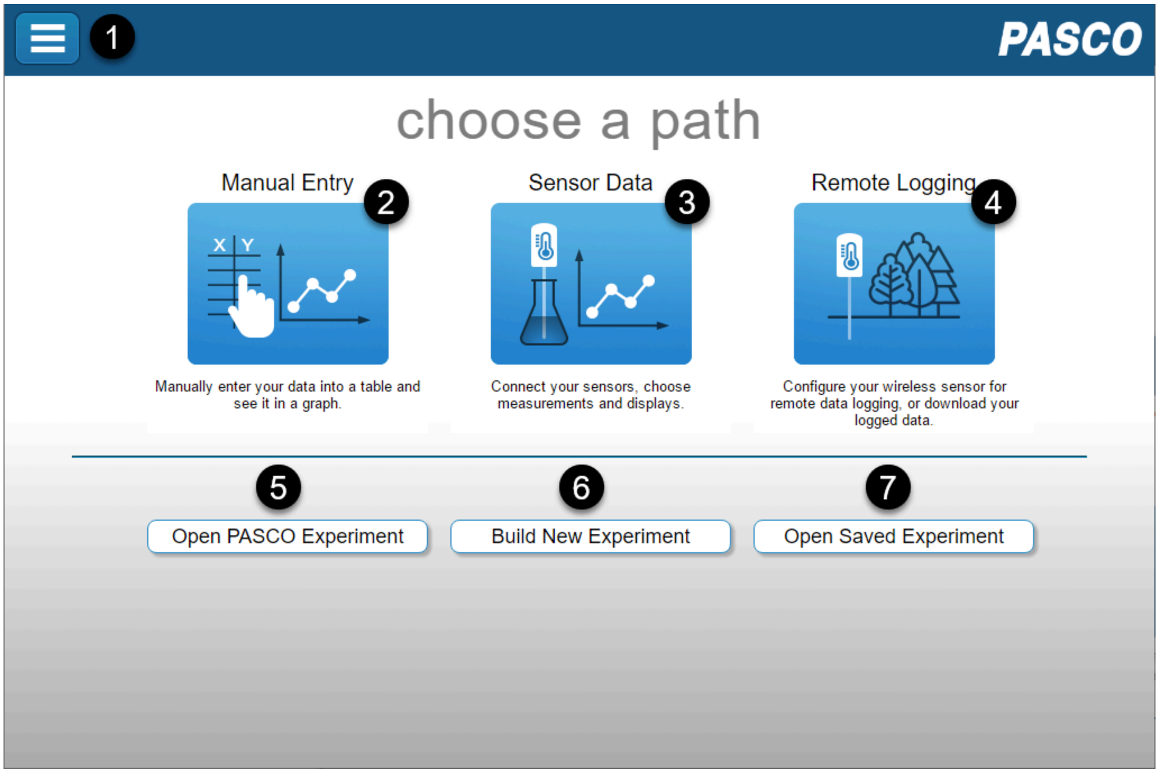
Contents

Getting Started

Start here to get an overview of the interface and use the Welcome Screen options to quickly set up an experiment.

Welcome Screen

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



1 **Main Menu**

Start or join a Shared Session *(on page 80)*, set preferences, and check for updates.

2 **Manual Entry *(on page 14)***

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

3 **Sensor Data *(on page 6)***

Connect sensors and select measurements to display in common layouts.

4 **Remote Logging *(on page 11)***

Collect data on a Wireless Sensor without being connected to a computer or mobile device. After data collection is completed, download the data to SPARKvue for analysis.

5 **Open PASCO Experiment *(on page 15)***

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

6 **Build New Experiment *(on page 8)***

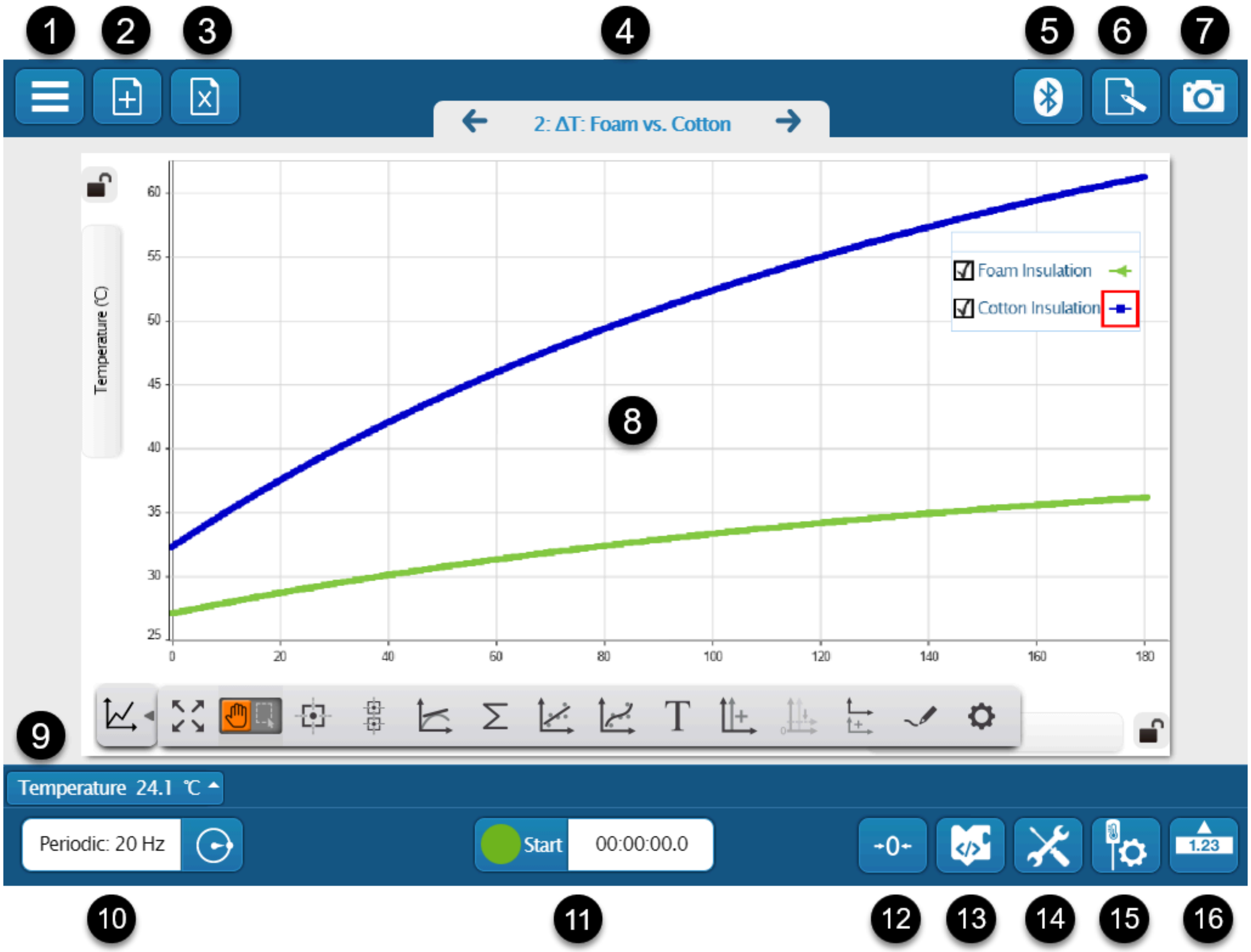
Create a custom experiment layout to display data and control hardware.

7 **Open Saved Experiment *(on page 84)***

Access previously saved files on your computing device.

Experiment Screen

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



1 **Main Menu**

- **Start New Experiment**: Return to the Welcome Screen *(on page 2)*.
- **Experiments**: Open a PASCO Experiment *(on page 15)*.
- **Open**: Open a previously saved experiment *(on page 84)*.
- **Save** or **Save As**: Save the current file to your computing device.
- **Export Data**: Export data *(on page 83)* as a CSV file.
- **Preferences**: Toggle auto-hide for new runs on all experiment pages.
- **Shared Session**: Start or join a Shared Session *(on page 80)*.

2 **New Page**

Build a new experiment *(on page 8)* page after the one currently displayed on the screen.

3 **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 *(on page 62)* and set up remote data logging *(on page 11)*.

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 *(on page 17)* 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 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 *(on page 77)*)
- Sampling rate *(on page 76)*
- Stop Condition *(on page 78)*

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 will change 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 *(on page 86)* 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: Adjust the number format, default units, and displayed color for measurements.

15 **Hardware Setup** 

Configure sensors and their measurements:

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

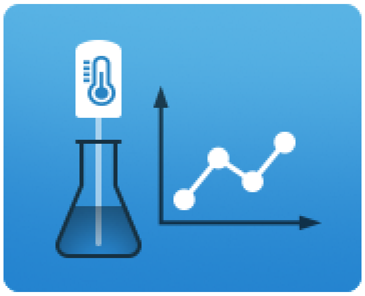
16 **Live Data Bar Visibility** 

Click to show or hide the Live Data Bar.

Set up and display sensor data

Connect sensors and select measurements to display in common layouts.

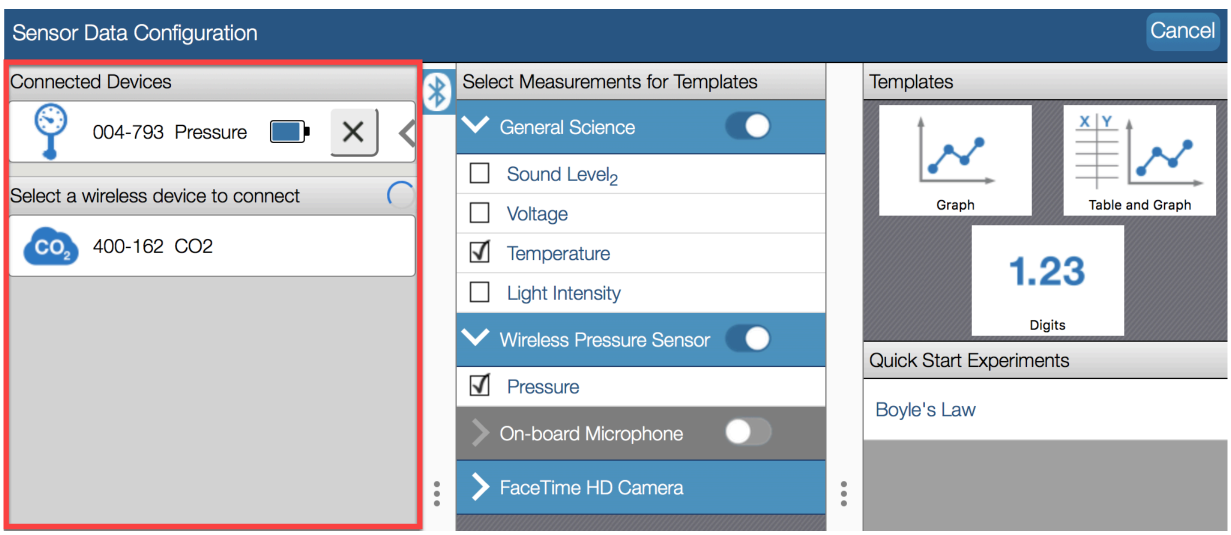
1. Open SPARKvue or click then select **Start New Experiment**.
2. Click **Sensor Data**:



3. Connect a Wireless Sensor:

    a. Turn on the sensor.
    b. Select the sensor that matches its device ID.

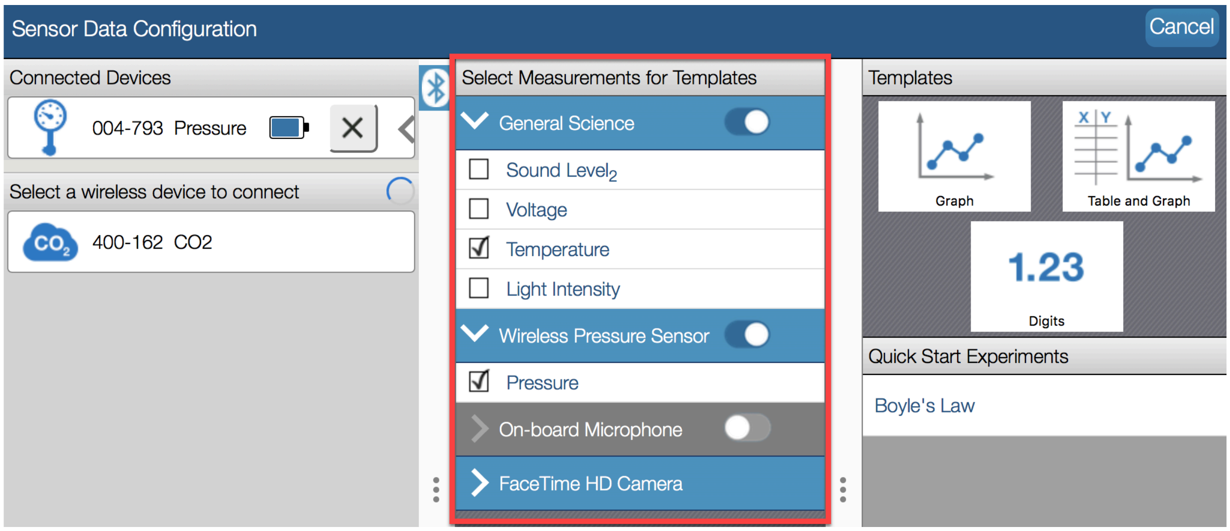
> **Note:**
> For PASPORT and Science Workshop sensors, see Connect a sensor to SPARKvue *(on page 62)*.



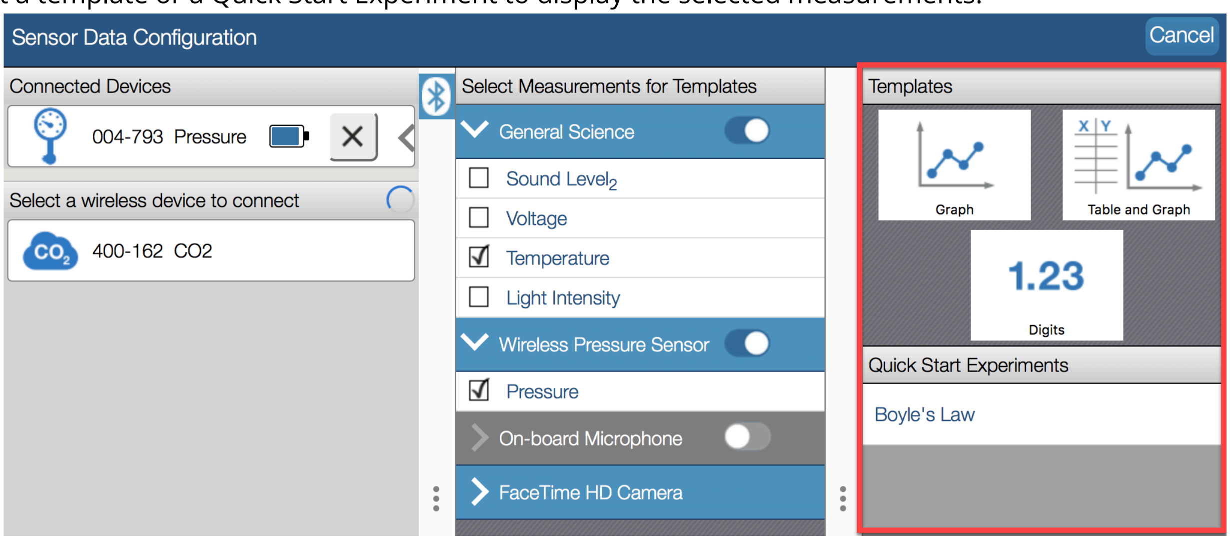
4. Select ☑ up to three measurements to display.

Tip:

- Toggle the switch to disabled to turn off sensors you do not 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 *(on page 3)*, click **Start** to begin collecting data.

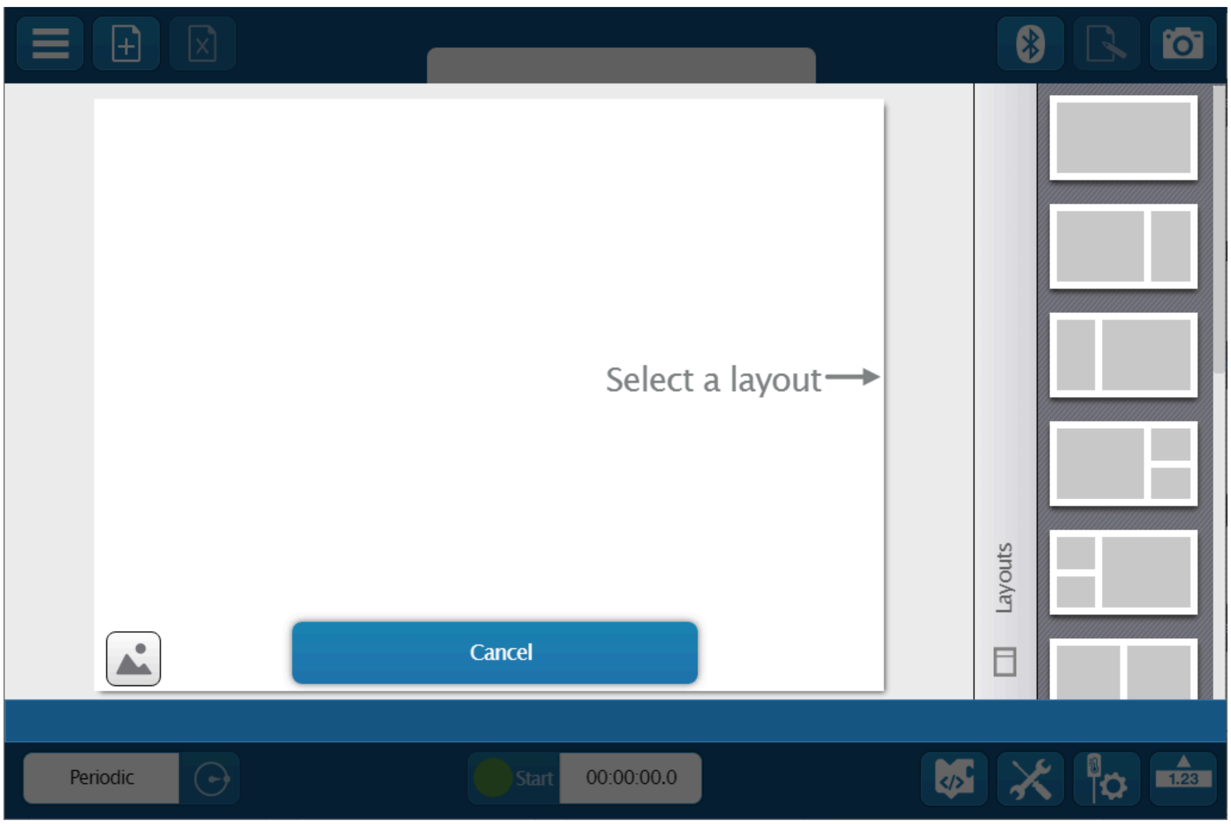
See Displaying and Analyzing Data *(on page 17)* for information on how to analyze data in each display.

Build a custom experiment page

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

1. Start a new page:
    - On the Welcome Screen *(on page 2)*, select **Build New Experiment**.
    - On the Experiment Screen *(on page 3)*, click .
2. Optional: Click  to add a background image to your page.
3. Select a layout containing one or more display elements. Scroll for additional layout choices.

> **Note:**
> Some displays require a minimum size. See Table 1 : Minimum size for displays *(on page 10)*.



4. Connect all sensors *(on page 62)* sensors that will be used with the experiment page.

5. Select a display *(on page 17)* for each area on the experiment page.

> **Note:**
> Some displays will not appear unless a supported sensor or device is connected.

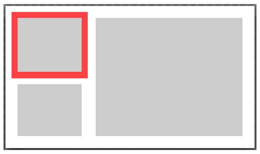
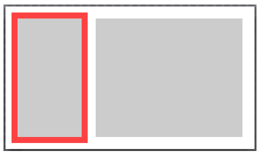
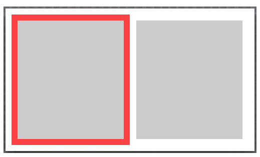
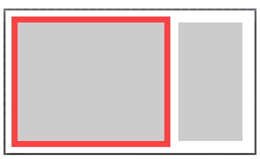
6. For each display, click **Select Measurement**, then select a measurement from the menu.

> **Note:**
> You need to first connect a sensor to SPARKvue before you can select a measurement in a display.

7. Click **Start** to begin collecting data.

To delete an experiment page, click . See the Experiment Screen *(on page 3)* page for information on how to use the tools on the screen. See Displaying and Analyzing Data *(on page 17)* for information on how to analyze data in each display.

Table 1. Minimum size for displays

Minimum Size	Displays
	1.23
	
	
	
	

Set up remote logging

Collect data on a Wireless Sensor without being connected to a computer or mobile device. After data collection is completed, download the data to SPARKvue for analysis.

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

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 do not 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 amount of time that the sensor will be able to log data is indicated at the bottom of the window. 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**.

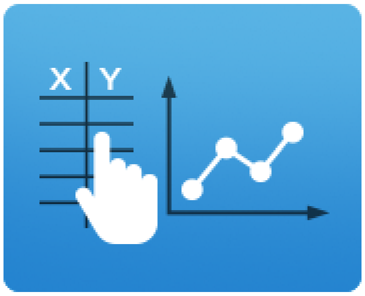
Data logging begins immediately after you click **OK** or press the power button on the sensor (if **Sensor Button Deferred Logging** was selected). The Bluetooth status light will blink yellow and green until data logging begins. When the sensor starts logging data, the Bluetooth status light will blink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

Click **OK** and close SPARKvue. To stop data logging, turn off the sensor or connect it to SPARKvue to download the data *(on page 64)*.

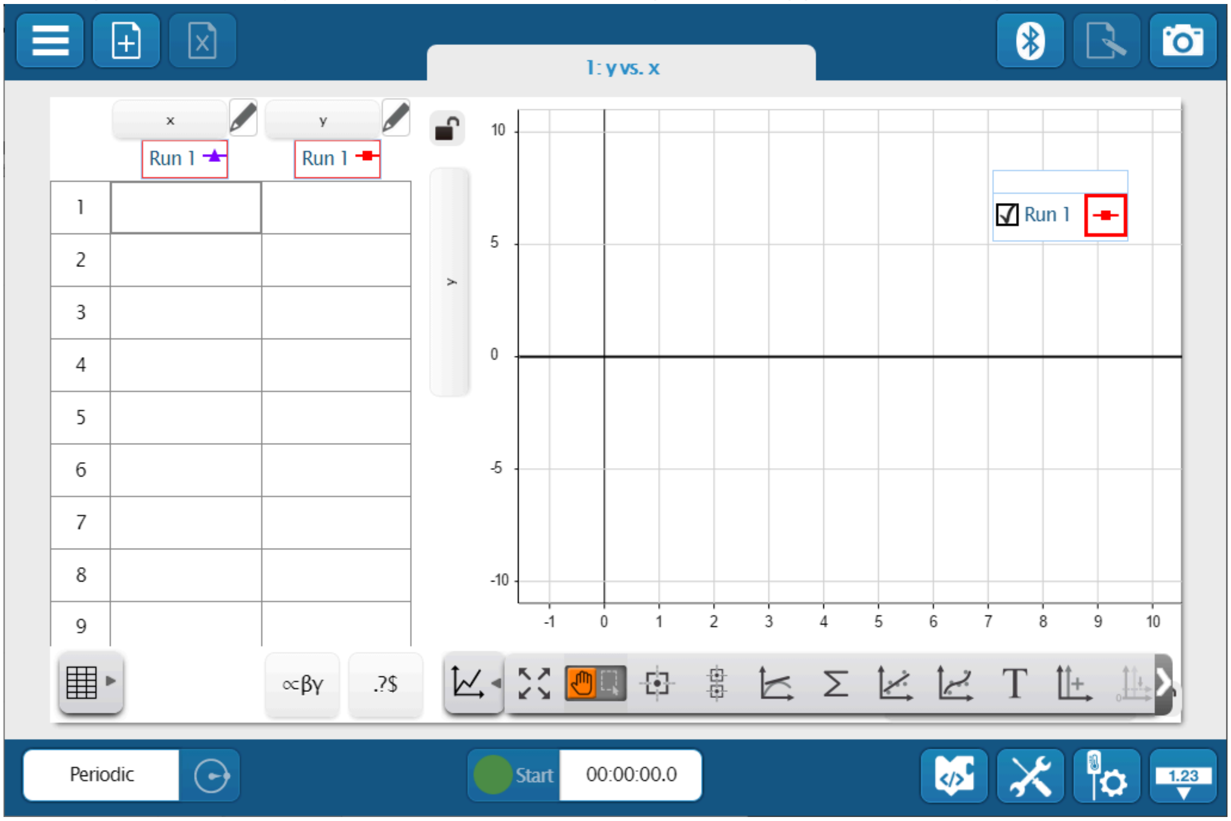
Manually enter data in a graph

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 will appear 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 *(on page 19)* and table *(on page 42)* 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 *(on page 2)*, select **Open PASCO Experiment**.
   - On the Experiment Screen *(on page 3)*, 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. Worksheets for each lab can be downloaded for free from the PASCO Experiment Library.
   - **Essential Physics**

     This directory contains files designed for the Essential Physics lab manual. Worksheets for each lab can be downloaded 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. Relevant files appear when a sensor is connected using the Sensor Data *(on page 6)* path from the Welcome Screen.
   - **SPARKlabs**

     Files contained in the main directory are complete labs containing data collection steps and analysis questions. These files are intended to be used digitally and do not have a corresponding print worksheet.
3. Connect a sensor to SPARKvue. *(on page 62)*
4. Click **Start** to begin collecting data.

See the Experiment Screen *(on page 3)* page for additional information on how to use the tools on the screen.

Displaying and Analyzing 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

Display data in a XY line graph, showing one or more measurements versus time (or another measurement). Modify how the data is displayed 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 *(on page 24)***

Click and drag to scale each axis individually.

4 **Plot Area *(on page 24)***

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 *(on page 24)***

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 mouse cursor or touch.

- 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 *(on page 28)***

Use to determine the coordinates of a single data point. Use the delta tool *(on page 30)* to determine the difference between two points of data.

Multi-Coordinates Tool *(on page 29)*

Use to compare coordinates across multiple runs or measurements.

10 **Slope Tool *(on page 37)***

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

11 **Statistics *(on page 34)***

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

12 **Linear Fit *(on page 31)***

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 *(on page 33)*

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

13 **Annotation Tool *(on page 38)***

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

14 **Add y-axis *(on page 21)***

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 *(on page 21)*

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

Add Plot Area *(on page 23)*

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

15 **Prediction Tool *(on page 40)***

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

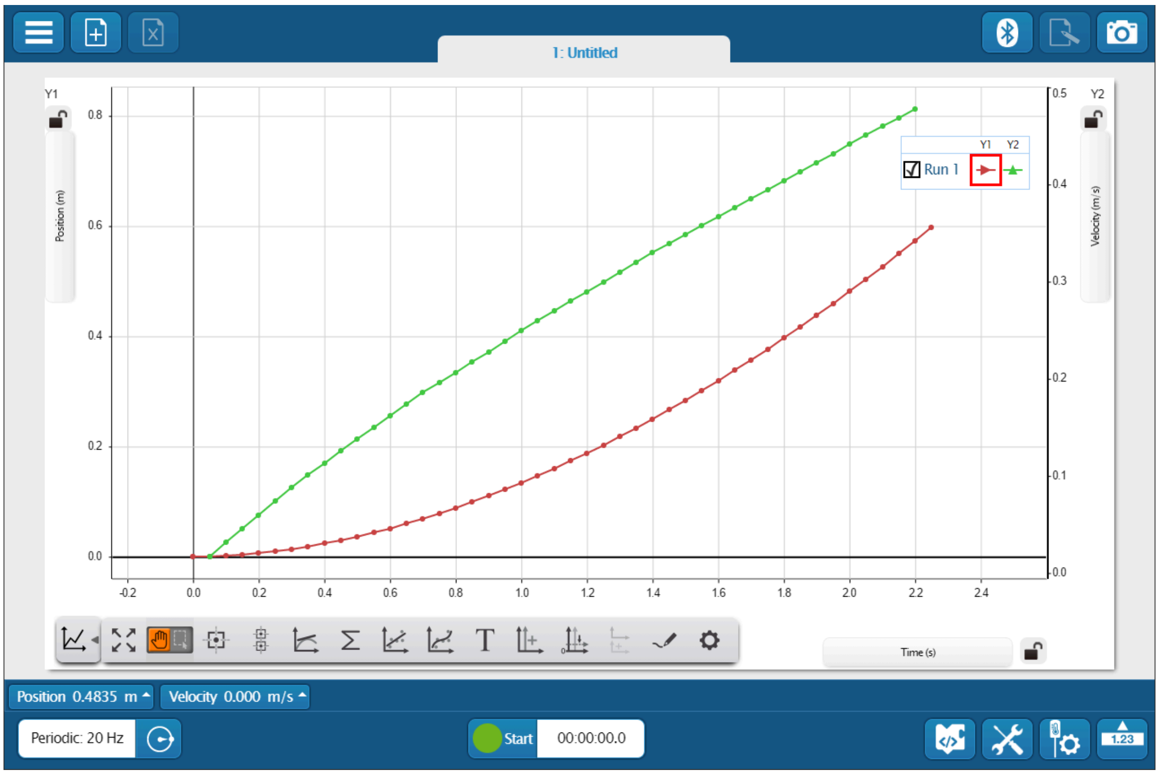
16 **Properties *(on page 27)***

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

Add or remove an additional y-axis to a graph

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 *(on page 23)*.



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 *(on page 24)*.

Remove an axis

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

Note:
It is 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 *(on page 21)*.

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 will scale to show the data of all runs. If you want to scale to a single run, hide the other data runs *(on page 26)* 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:

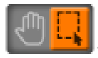
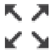
1. Click to lock the scale of one axis.
2. Scale the other axis by clicking and dragging the axis.

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

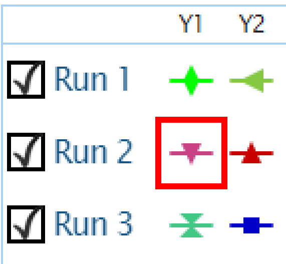
1. Go to  then toggle the cursor tool to .
2. Click and drag over the area you want to zoom in.
3. Click .
4. Optional: Remove the selection highlighter by clicking it then clicking .

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. If you wish to delete data runs, see Delete Runs *(on page 74)*.

Visible runs are displayed in the line graph legend.



Click ☐ to select the runs you want to make visible.

Click ☑ to deselect runs you want to hide.

Show or hide data points and connecting lines in a graph

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 show (On) or hide (Off) connected lines.
3. Click the Data Point Marker menu then select one of the following:

    - **Hidden**: Do not 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.

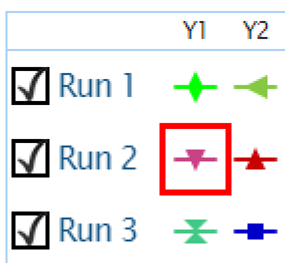
Coordinates of a Single Point

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. The selected run is indicated by a red border.

   Figure 2. Measurement Y1 of Run 2 is selected in the legend.



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 .

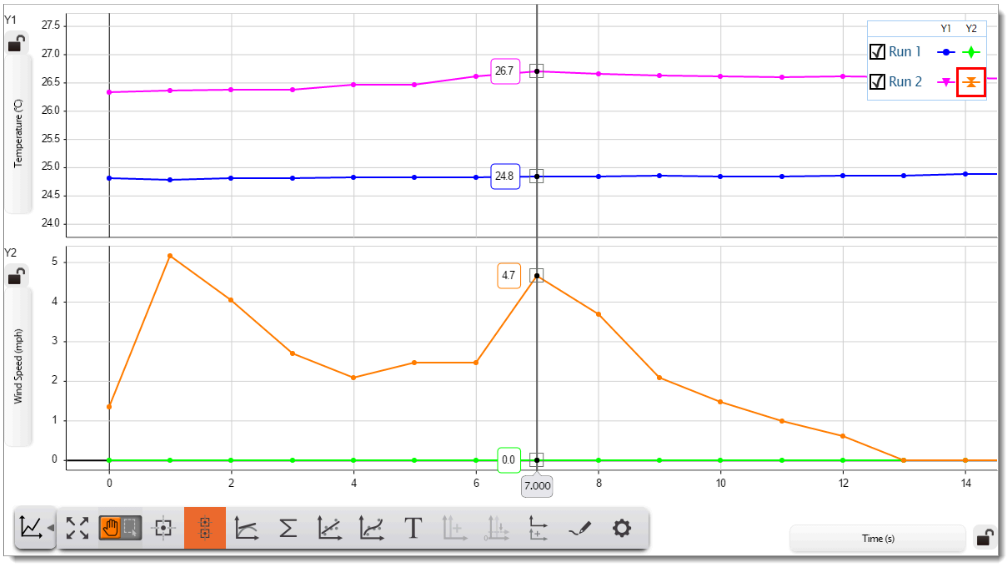
Tip:

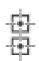
Click  to determine the difference between two data points *(on page 30)*.

Compare Coordinates of Multiple Points

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

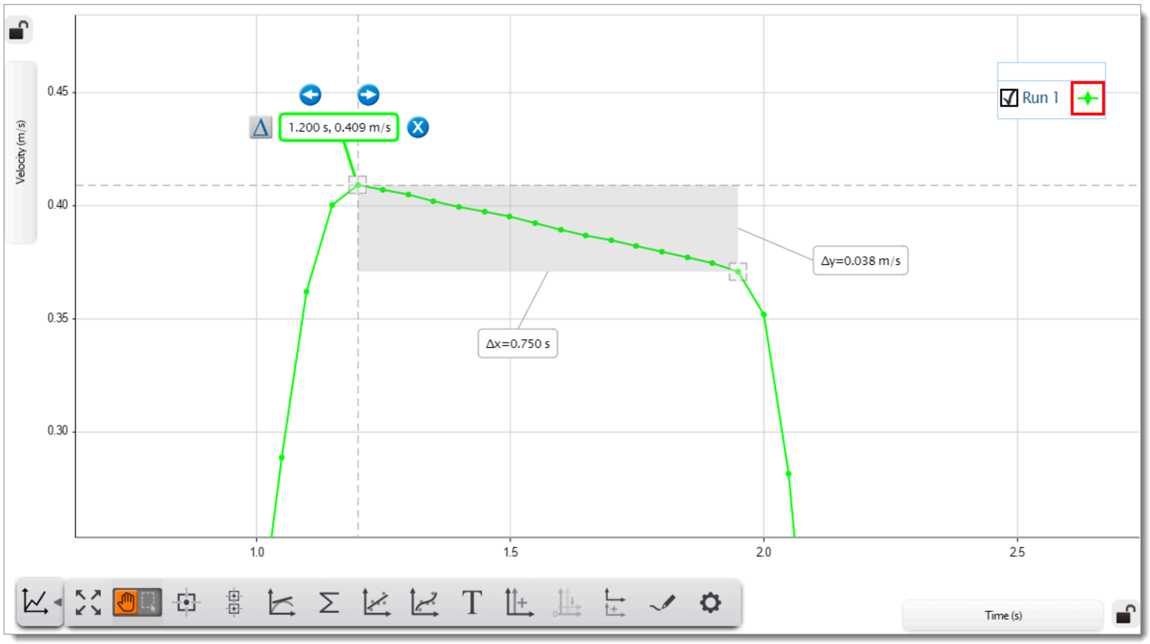
Figure 3. Using the multi-coordinates tool in a line graph.



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

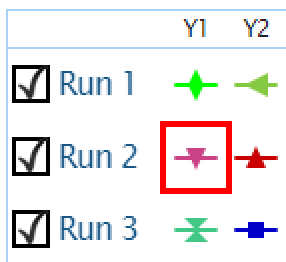
Determine the difference between two data points on a line graph

The coordinates tool includes a delta tool to determine the difference between two data points in a run of data. When the delta tool is activated it will display the difference of the vertical (Δy) and horizontal (Δx) coordinates between the two points.



1. Select the run you want to analyze in the graph legend. The selected run is indicated by a red border.

Figure 4. Measurement Y1 of Run 2 is selected in the legend.



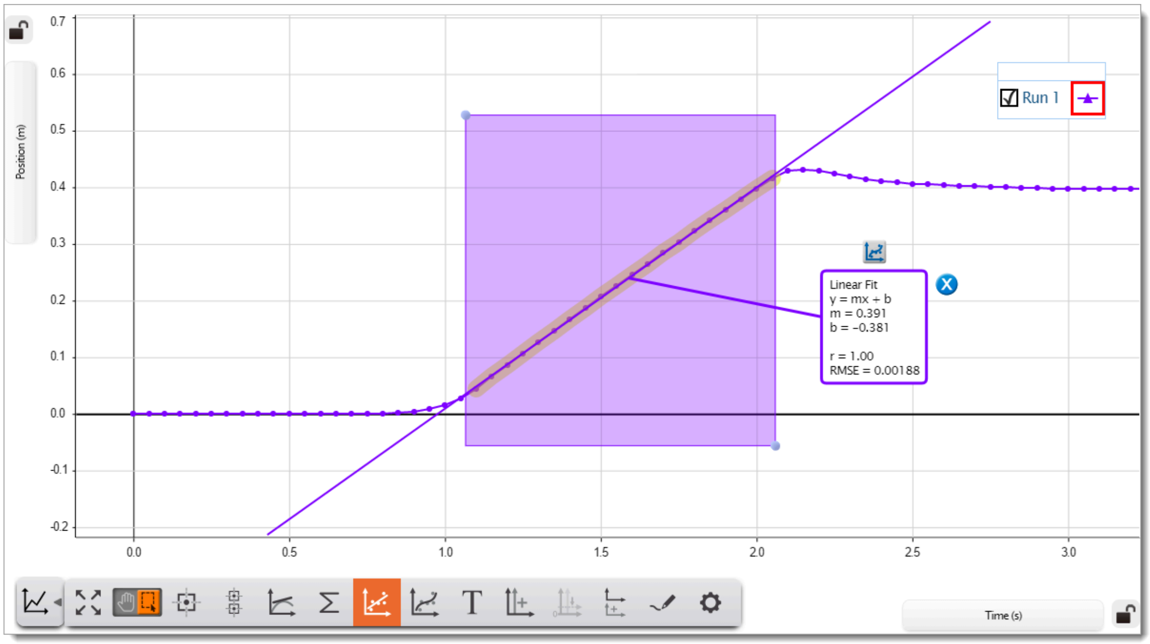
2. Click the point you want to analyze then click .
3. Click to make the delta tool visible.
4. Place the tool on a different data point by dragging the tool or clicking and .

Remove the tool by clicking the tool then click .

Fit a line to data in a line graph

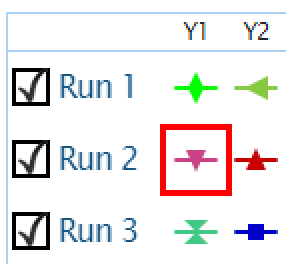
Use the linear fit tool to fit a line to data and determine the slope and y-intercept of the trend line.

> **Tip:**
> Other curve fits are available using the curve fit tool *(on page 33)*.



1. Select the run you want to analyze in the graph legend. The selected run is indicated by a red border.

Figure 5. Measurement Y1 of Run 2 is selected in the legend.



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.

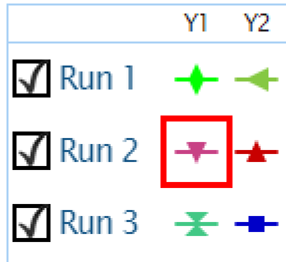
If you wish to try a different fit to your data, click the information box then click .

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. The selected run is indicated by a red border.

   Figure 6. Measurement Y1 of Run 2 is selected in the legend.

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.

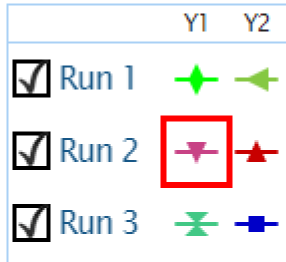
If you wish to try a different fit to your data, click the information box then click .

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. The selected run is indicated by a red border.

Figure 7. Measurement Y1 of Run 2 is selected in the legend.



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.

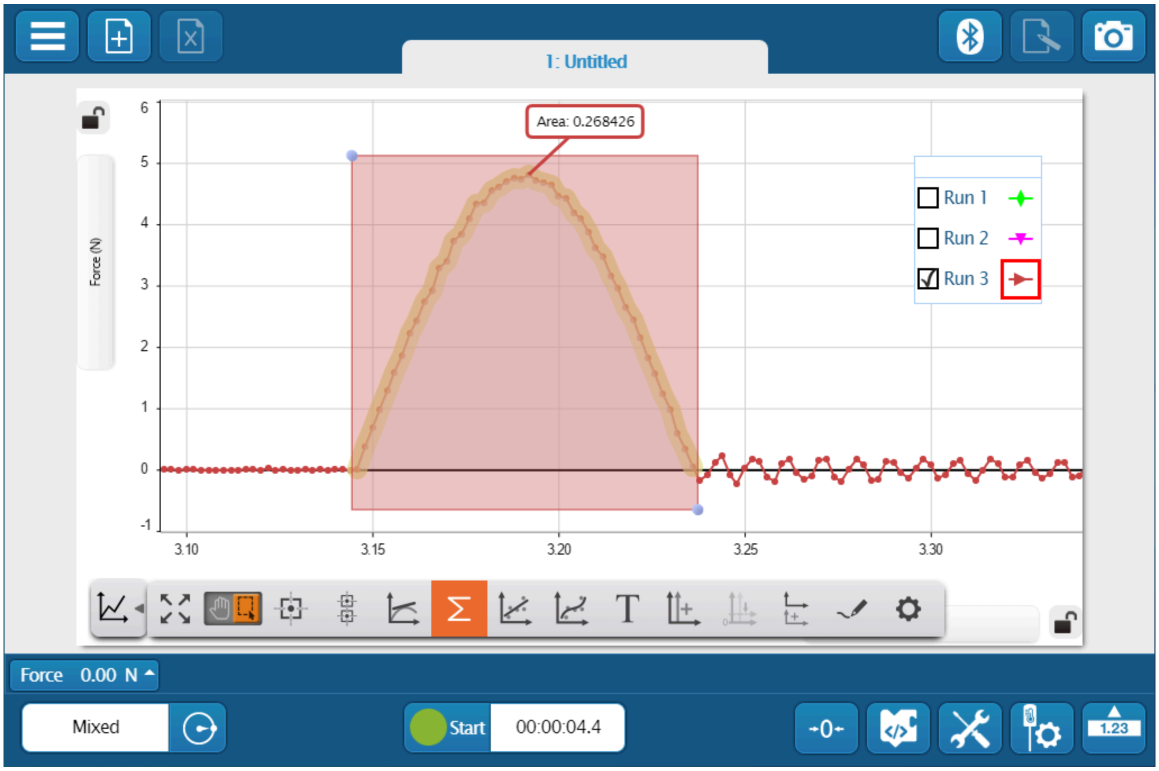
4. Click .
5. Select one or more statistics that you want to display then click OK.

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 in a line graph

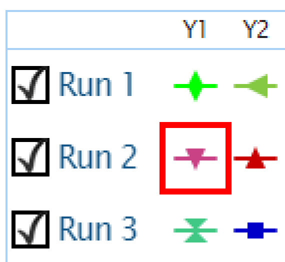
Use the area tool to find the area under a curve.

Figure 8. Using the area tool to find the area under a curve.



1. Select the run you want to analyze in the graph legend. The selected run is indicated by a red border.

Figure 9. Measurement Y1 of Run 2 is selected in the legend.



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 .

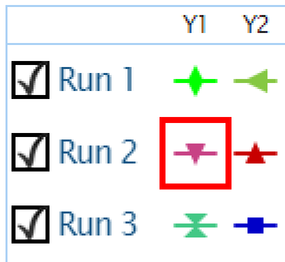
Determine the slope of the tangent to a curve

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

> **Tip:**
> If you are interested in finding the slope of a range of data, see Linear Fit *(on page 31)*.

1. Select the run you want to analyze in the graph legend. The selected run is indicated by a red border.

   Figure 10. Measurement Y1 of Run 2 is selected in the legend.

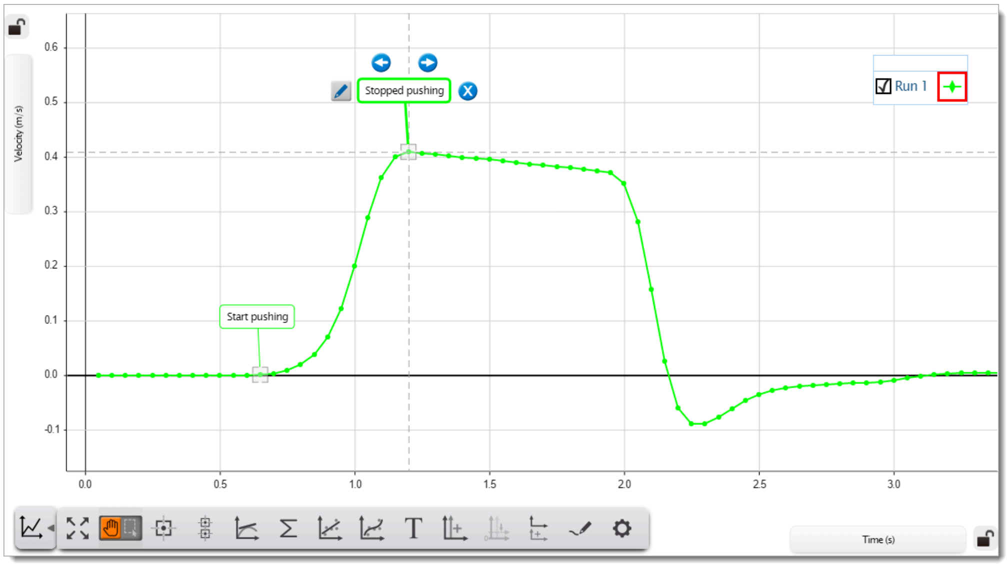
2. Click the point you want to analyze then click .
3. Place the tool on a different data point by clicking and dragging the tool or clicking or .

To remove the tool, click the tool then click .

Label a data point in a line graph

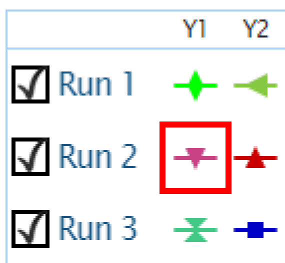
Use the annotation tool to label a data point in a graph with a brief note.

Figure 11. Two points labeled on a line graph.

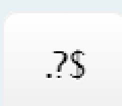


1. Select the run you want to analyze in the graph legend. The selected run is indicated by a red border.

Figure 12. Measurement Y1 of Run 2 is selected in the legend.



2. Click the point you want to analyze then click .
3. Click the Enter Note field and type your note.

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

4. Click OK.

The following modifications can be made to the annotation tool:

- Place the tool on a different data point by clicking and dragging the tool or clicking  or .
- Edit the note by clicking the tool then clicking .
- Remove the note by clicking the tool then clicking .

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.

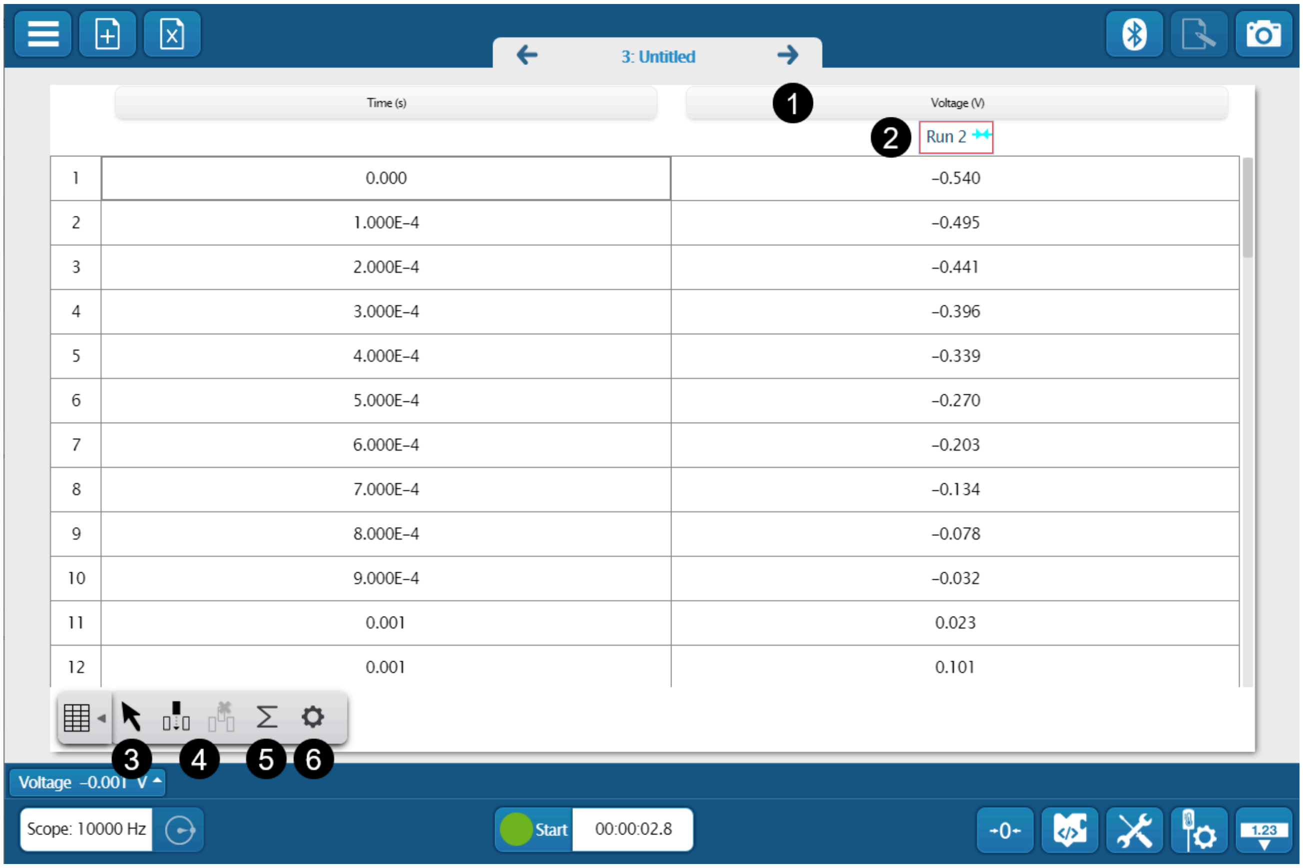
To remove the prediction from the graph, click to deselect the tool.

Table Display

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

Table Display Overview

Get an overview of the Table display tools.



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 or remove columns in a table

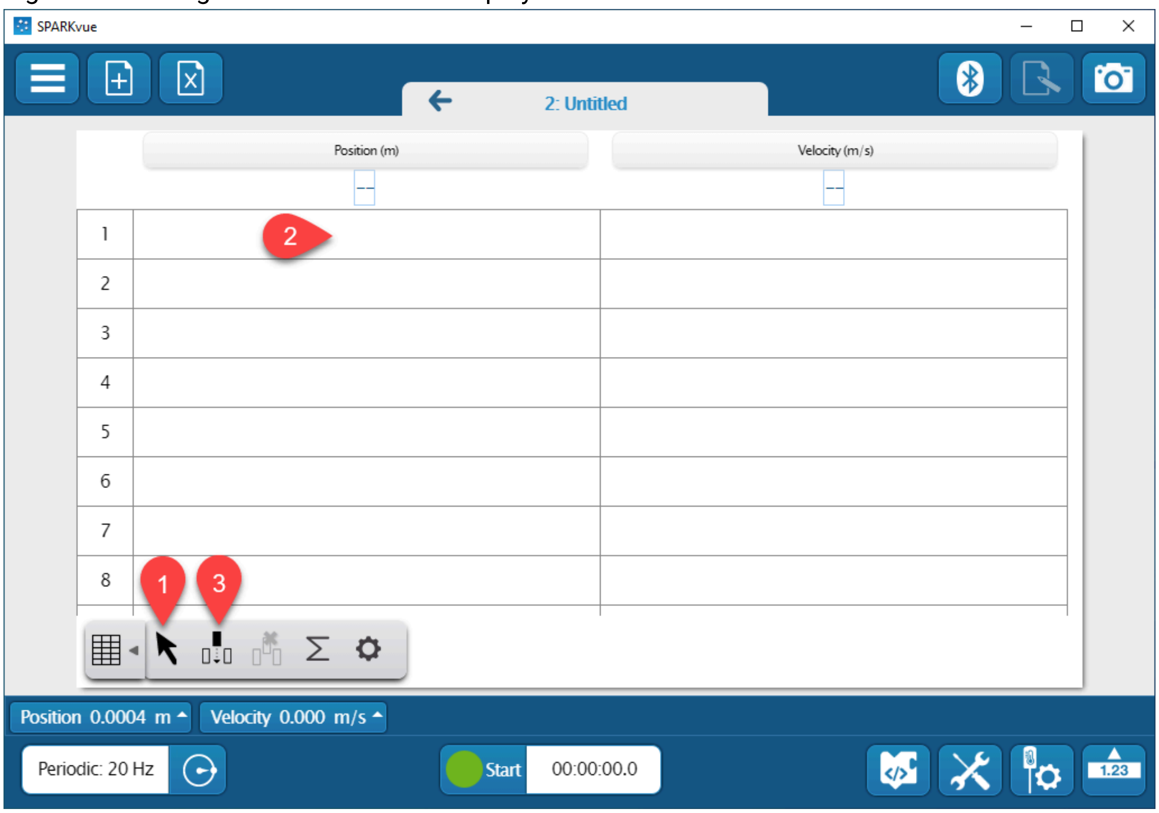
Insert up to eight columns of data and remove unwanted columns in a table display.

Add a column

Go to then click .

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

Figure 13. Adding a column to a table display.



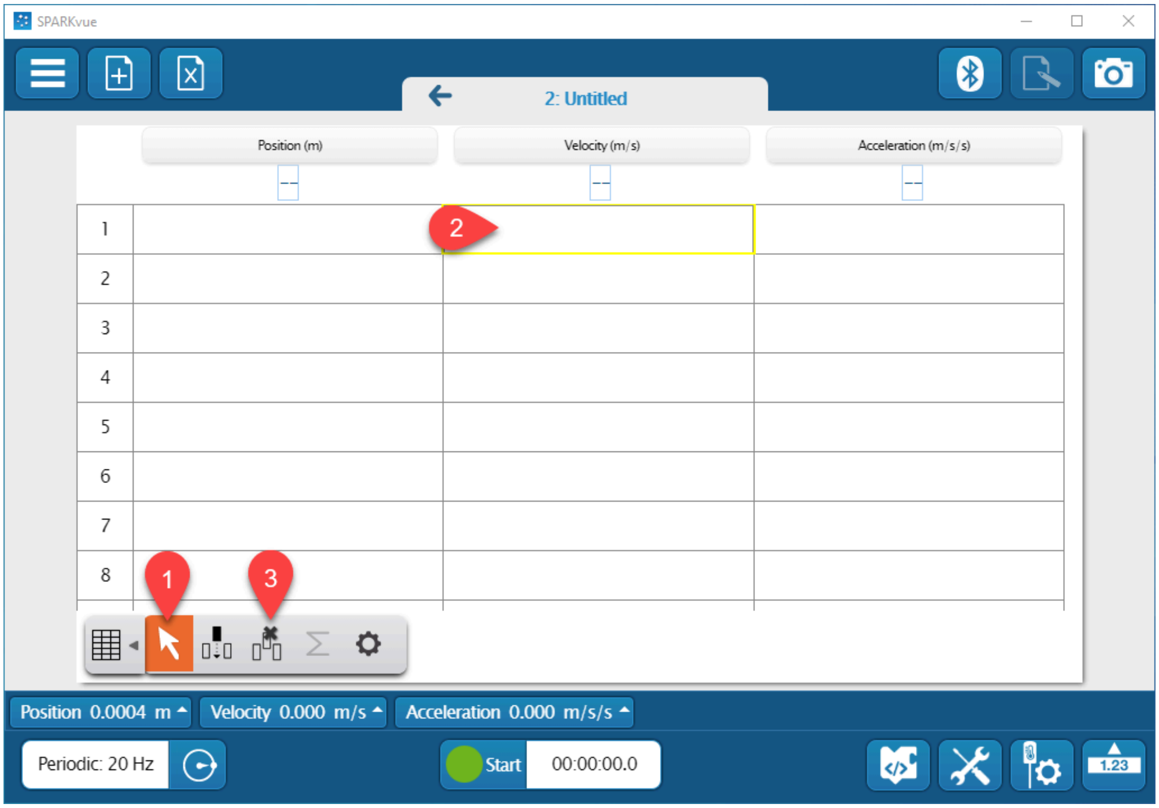
1. Go to then click .
2. Select a cell in the column to the right of where you want to insert a new column.
3. Click .

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 *(on page 74)*.

Figure 14. Removing a column from a table display.



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 in a table

Modify the name and units of user-entered data in a table display.

Note:
Changes made to the measurement name and units in the table will also carry 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 the statistics of a measurement in a table

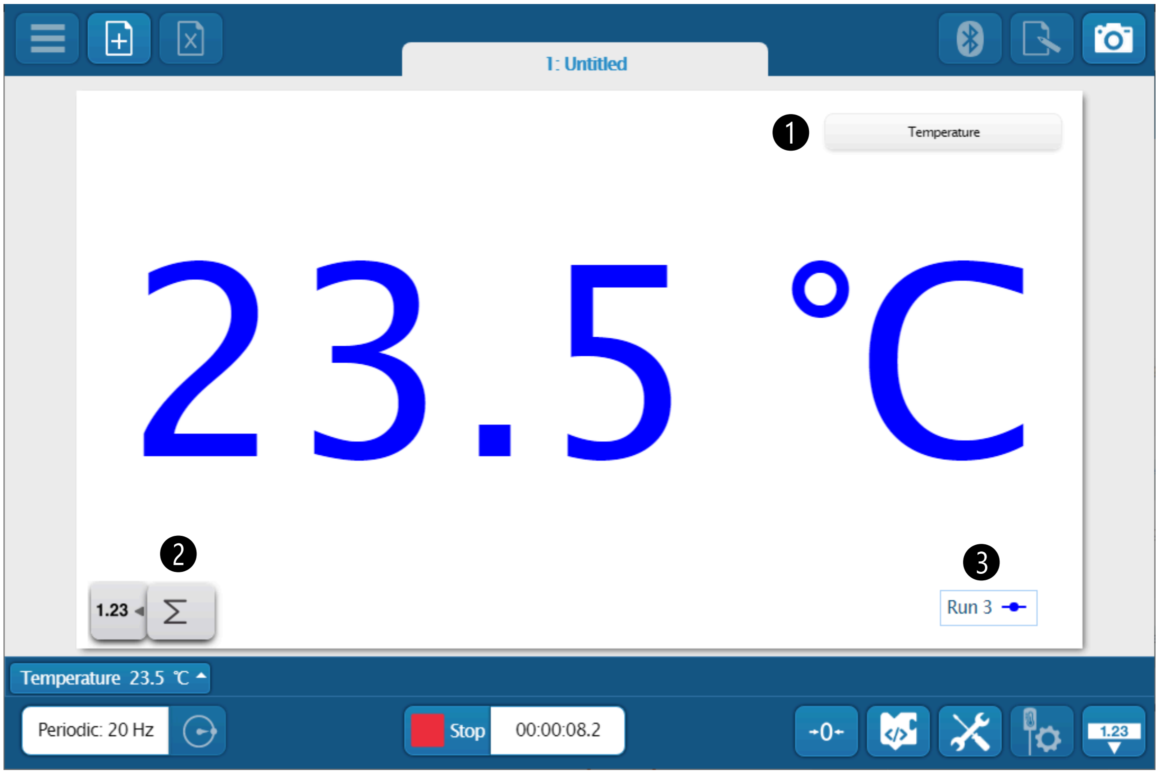
Use the statistics tool to display the minimum, maximum, mean, and standard deviation of data and display it at the bottom of the column.

1. Go to then click Σ .
2. Select one or more statistics you want to display then click OK. The statistics for all the data is displayed 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.

Digits Display

Show the most recently recorded data sample as digits, statistics of a single data run, or text output *(on page 117)* 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 *(on page 86)* that includes text output *(on page 117)*.
2. Click **Select Measurement**, select the **User-entered** tab, then select the name of the text output.

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

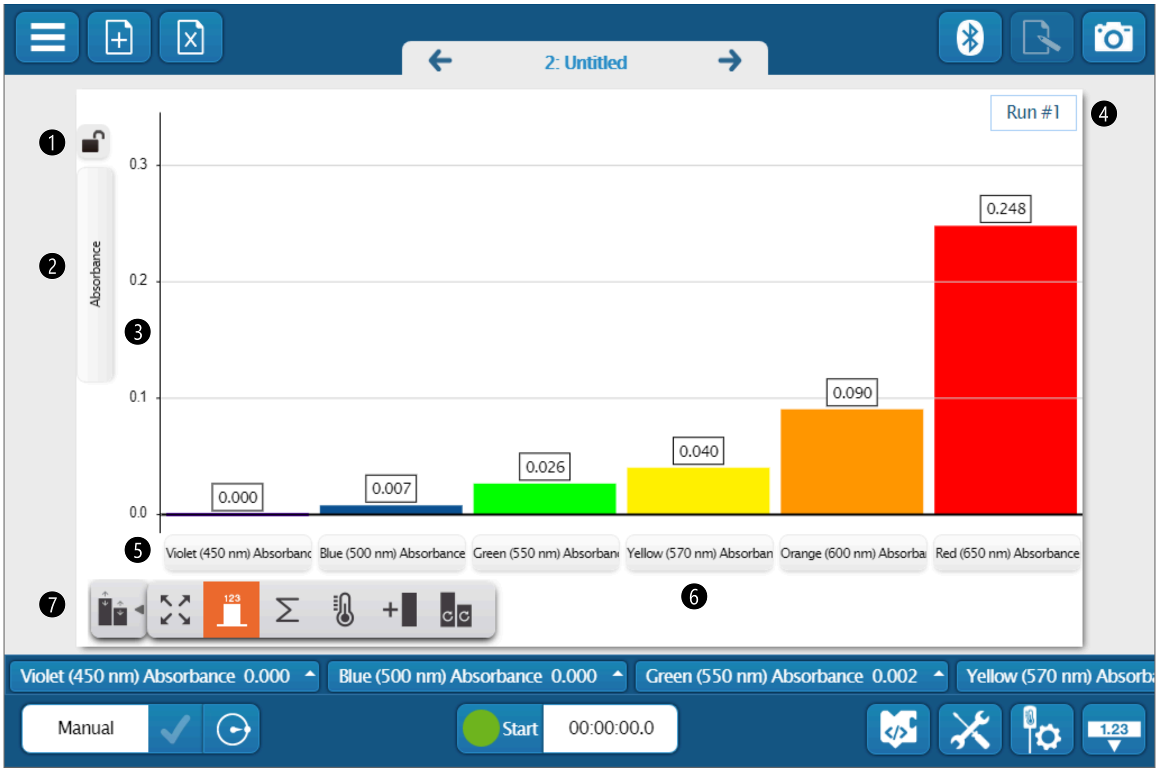
- Click Σ to display the minimum, maximum, mean, or standard deviation for the selected run.
- Click to scale the meter to the best fit.
- Click 1.23 to display the measurement as digits below the meter.
- Click to change the visual appearance of the meter and manually set the minimum and maximum values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

3 Run Selector

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

Bar Meter Display

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



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 is only possible to select a measurement that is 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 Σ

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 *(on page 74)*.

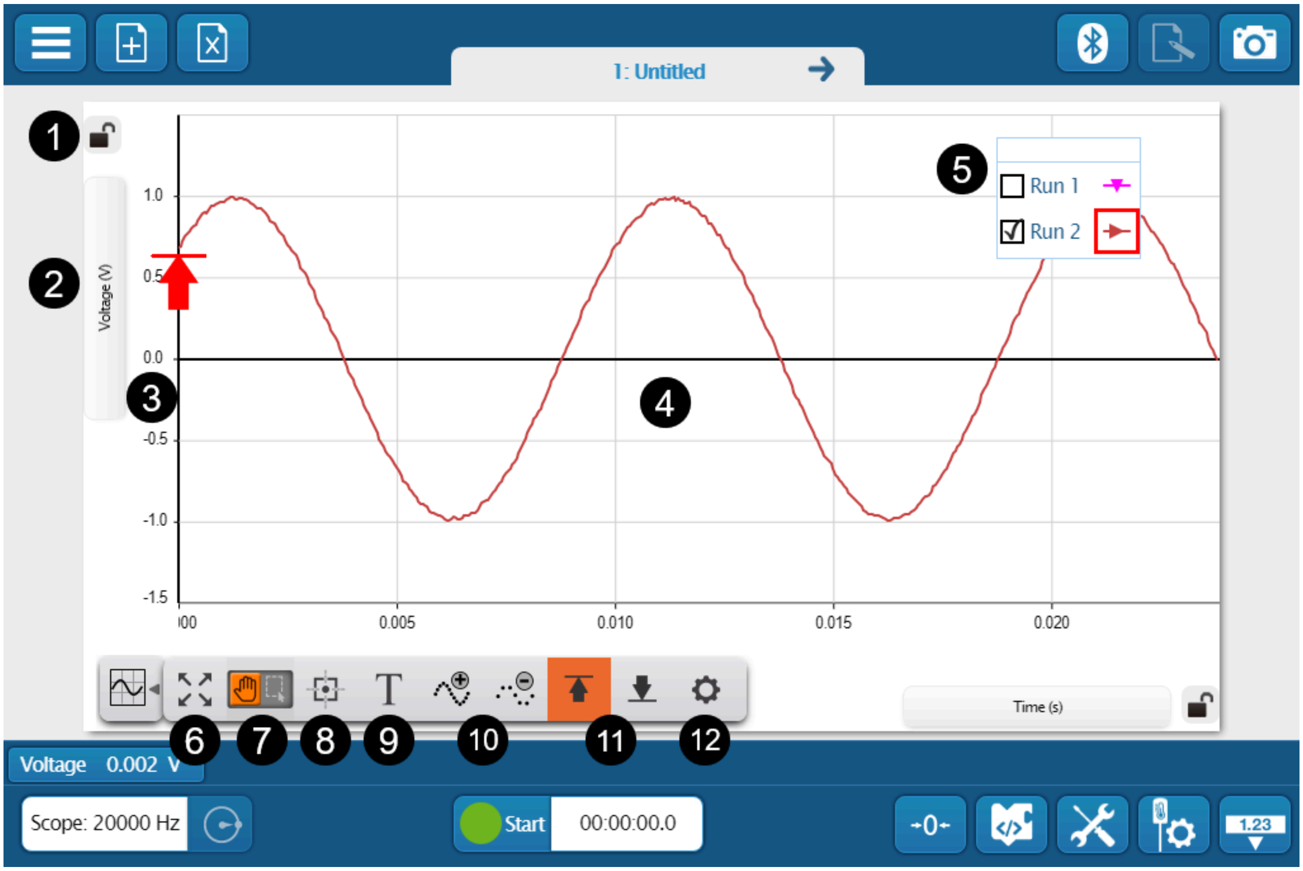
Scope Display

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.

Scope Display Overview

Get an overview of the Scope display tools and how to modify the display.

Note:
The Scope display will only appear as an option when a supported sensor *(on page 55)* is connected.



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 or touch.

- 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 T **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 will make a smoother trace but decrease 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.

Supported Sensors

The Scope display can be used 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

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.

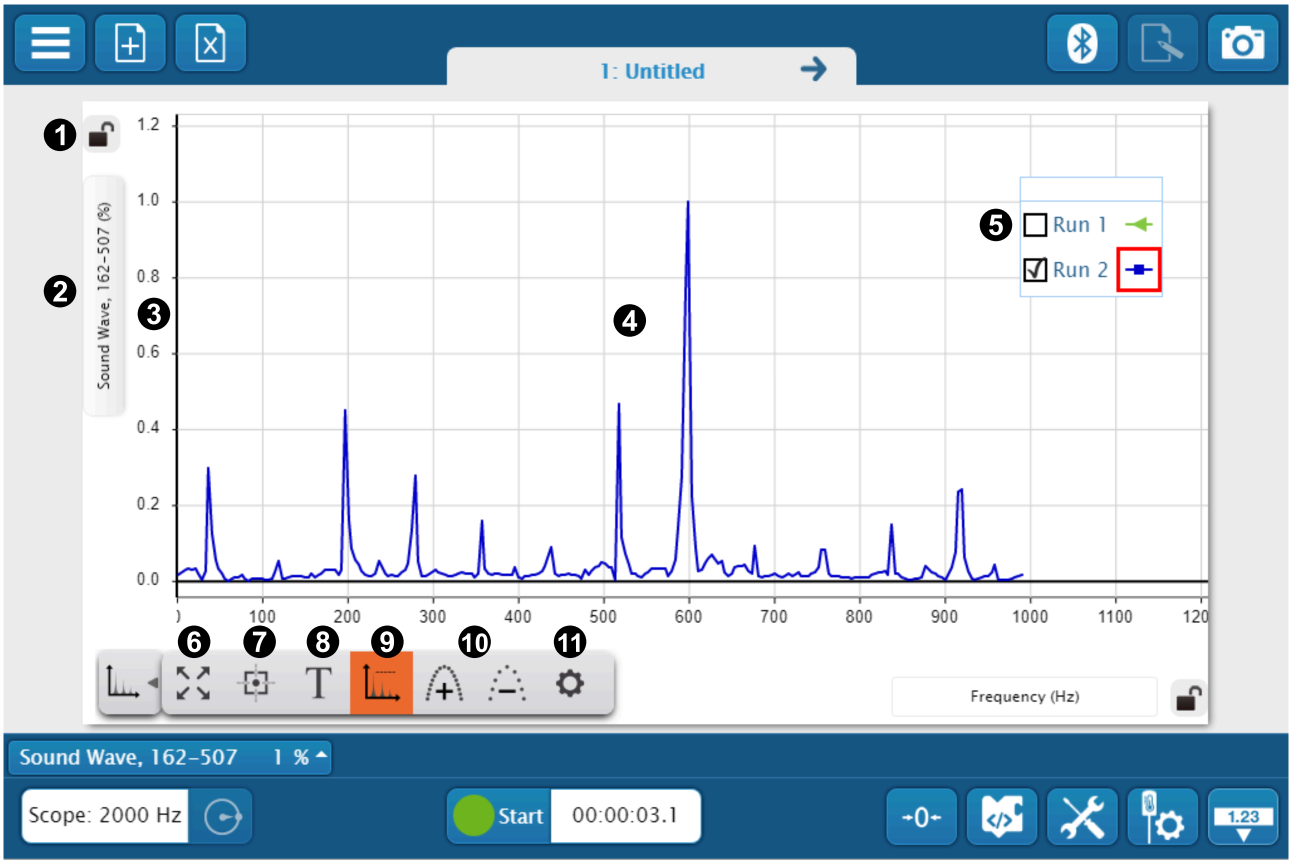
> **Tip:**
> A positive edge trigger will display the trace as the signal is rising at the trigger point. A negative edge trigger will display a trace as the signal is falling at the trigger point.

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

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:
The Scope display will only appear as an option when a supported sensor *(on page 59)* is connected.



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 is 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

The FFT display can be used 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

Sensors and Equipment

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

Connect a sensor to SPARKvue

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

> **Tip:**
> Use the Sensor Data *(on page 6)* path to connect sensors from the Welcome Screen.

Wireless Sensors

Connect Wireless Sensors to SPARKvue using Bluetooth or USB (if available):

- 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**.

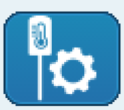
PASPORT Sensors

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

1. Power on the interface (if required).
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 are not compatible with SPARKvue.

1. Power on the interface (if required).
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 Science Workshop port on the interface then:

    - **Analog Sensors**

        a. Click to open **Hardware Setup**.
        b. Click for the port the sensor is connected 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) will work 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**.

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 is not 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 will not appear in the **Wireless Devices** window if the Bluetooth status light is blinking 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 will 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.

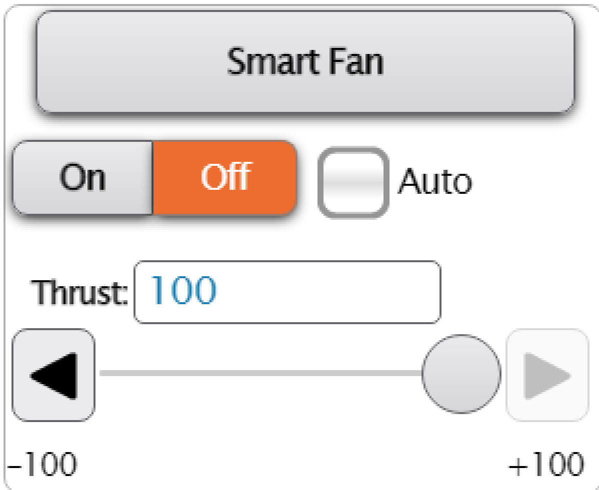
  a. Click Open PASCO Experiment or Open Saved Experiment.
  b. Select a file to open.

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

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 *(on page 8)* (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 will automatically turn on when collecting data and turn 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

A program can be created to control the Smart Fan using the Code *(on page 86)* tool. The following two blocks are provided in the **Hardware** group:

On/Off Block 

Use this block to turn the fan on when its parameter is set to `true` and off when set to `false`. A Logic block can be used 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 the Smart Ballistic Cart Accessory Projectile

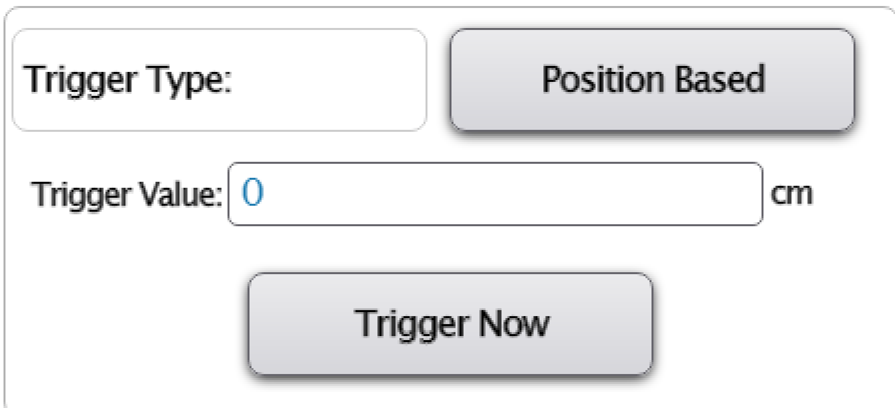
Launch a projectile from the Smart Ballistic Cart Accessory by setting a position or time based trigger.

> **Warning:**
> Wear safety glasses while using the Smart Ballistic Cart Accessory. Do not 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 *(on page 8)*. 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 will trigger 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 will launch when the cart returns to its starting position.

Trigger Now

Click to launch the projectile manually. This button can be used while data is being collected or not.

Create a Program

A program can be created to launch a projectile from the Smart Ballistic Cart Accessory using the Code *(on page 86)* tool. The block shown here is found in the **Hardware** group.


release
Ballistic Cart, Blue
projectile

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 *(on page 8)*. 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 current Vector Displayed being configured when using more then one 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 that will be displayed on the Vector Display. To change, click the current range then select a range to display.

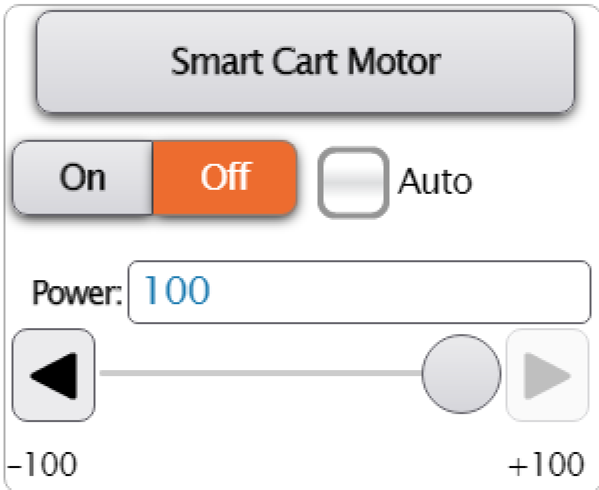
Change Sign

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 *(on page 8)* (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 will automatically turn on when collecting data and turn 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

A program can be created to control the Smart Cart Motor using the Code *(on page 86)* tool. The following two blocks are provided in the **Hardware** group:

On/Off Block 

Use this block to turn the motor on when its parameter is set to `true` and off when set to `false`. A Logic block can be used 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.

Managing 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 Data Run

Rename runs to provide a useful description your data.

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 sample rate of a sensor

Change how many data points are collected per unit of time.

The default sample rate will work fine 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), the sample rate should be lowered to conserve memory and prevent the data file from getting too large.

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:**
> Higher sampling rates may be limited 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.

The current sample rate is displayed at the bottom of the Experiment Screen. If different sample rates are set for connected sensors, the sample rate will display "Mixed."

Set Up and Use Manual Sampling

Manually select when data points are recorded instead of continuously recording data.

While in Manual Sampling mode, data is continually *monitored* but not *recorded* until the Keep Sample button is clicked. This mode is used when recording a sensor measurement versus a non-sensor measurement, such as in a Boyle's Law or Beer's Law experiment. It is 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 *(on page 6)* or Build New Experiment *(on page 8)* 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 will also appear in the table after clicking Start.

5. Click Start to begin data collection. The current measurement is displayed but not recorded.
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 will automatically stop after the specified time unless Stop is clicked.

Sharing and Saving 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.

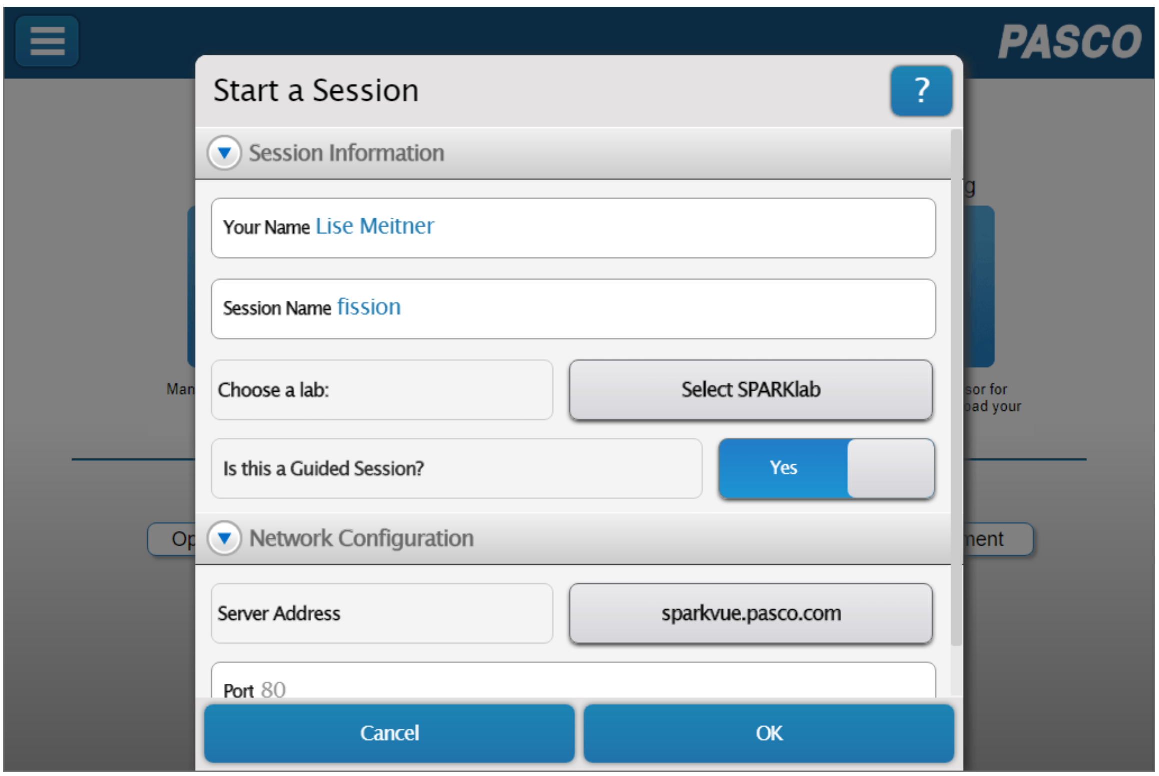
Use a Shared Session to:

- 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

Shared Sessions can be initiated from the Welcome Screen *(on page 2)* or the Experiment Screen *(on page 3)*.

- 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. Others will use this Session Name 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** *(on page 81)* 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 user hosting the session.
5. Click **OK**.

You will be connected to the Shared Session once you have been accepted by the user who is hosting the session.

Manage a Shared Session

After starting a Shared Session *(on page 80)* the Connected Users panel appears. Use the Connect Users panel to manage the Shared Session.

Approve and Reject Users

Users will appear in the Connected Users panel when try 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 will remain 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 comma separated values (CSV) file which can be opened with a spreadsheet application such as Excel or Google Sheets.

1. Click the Main Menu then select Export Data...
2. In your computing device's file explorer, set the file type as CSV and save the file in a easy to find location.
3. Open the CSV file in your preferred spreadsheet application.

Open a saved experiment

Access previously saved files on your computing device.

1. Open your computing device's file explorer:
   - On the Welcome Screen *(on page 2)*, select **Open Saved Experiment**.
   - On the Experiment Screen *(on page 3)*, click then select **Open**.
2. Use your computing device's file explorer to find and open the file.
3. Connect a sensor to SPARKvue. *(on page 62)*
4. Click Start

   to begin collecting data.

See the Experiment Screen *(on page 3)* page for additional information on how to use the tools on the screen.

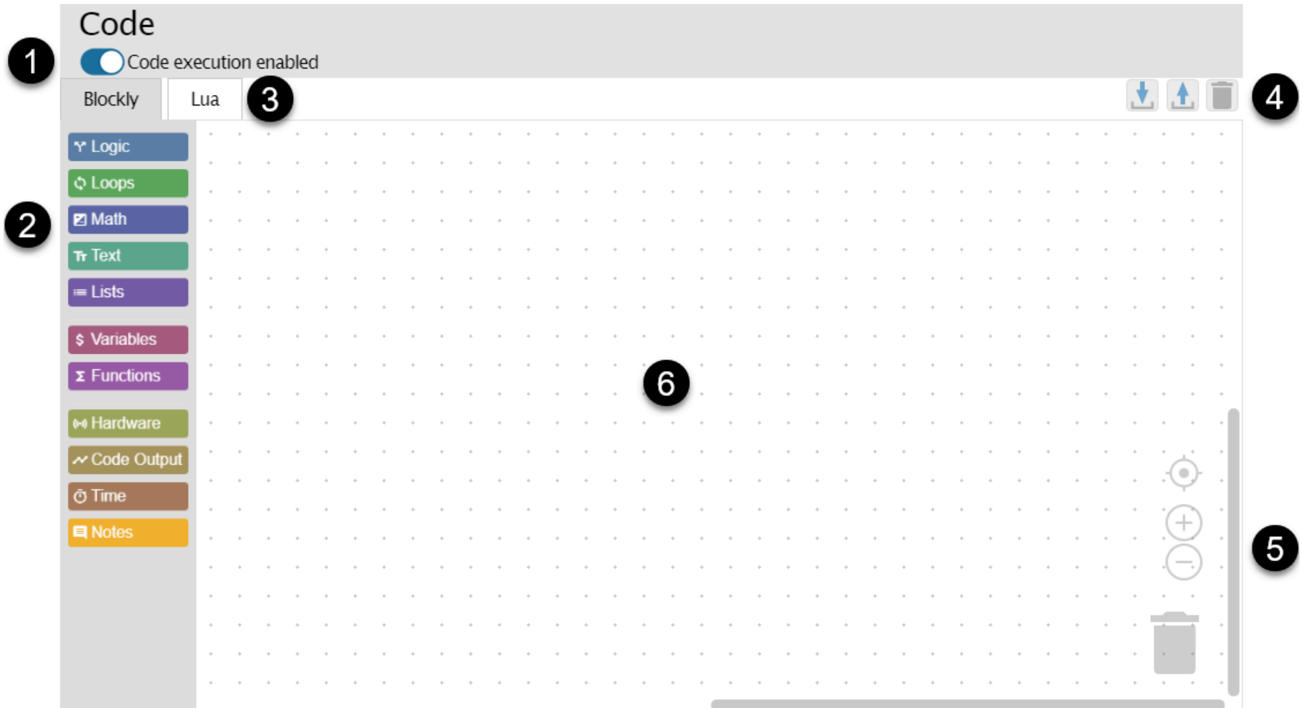
Blockly Coding

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

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 *(on page 3)*.



1 **Code Execution Switch**

Toggle to enable or disable code to run a program *(on page 129)* during data collection.

2 **Blockly Toolbox**

Click a category to add a block to the workspace to build a program *(on page 127)*.

Logic *(on page 90)*

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

Loops *(on page 93)*

Use the loop blocks to control the number of times lines of code are executed.

Math *(on page 96)*

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

Text *(on page 100)*

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

Lists *(on page 104)*

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

Variables *(on page 107)*

Use the Variable blocks to store a value that can be referred to later.

Functions *(on page 110)*

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

Hardware *(on page 114)*

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

Code Output *(on page 117)*

Use the Code Output blocks to provide text or numeric output in a Digits *(on page 47)* or Table *(on page 42)* display.

Time *(on page 121)*

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

Notes *(on page 125)*

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

3 **Lua Code Viewer**

The editor provides a graphical interface for creating programs using the Lua programming language. Click the Lua tab to view (but not edit) what the program looks like in text-based code to help with debugging.

4 Code Tools

- Import *(on page 132)* code from a .pcbx file.
- Export *(on page 131)* all code as a .pcbx file for importing into other SPARKvue or Capstone files.
- Delete all code on the workspace.

5 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 deleted blocks.

6 Workspace

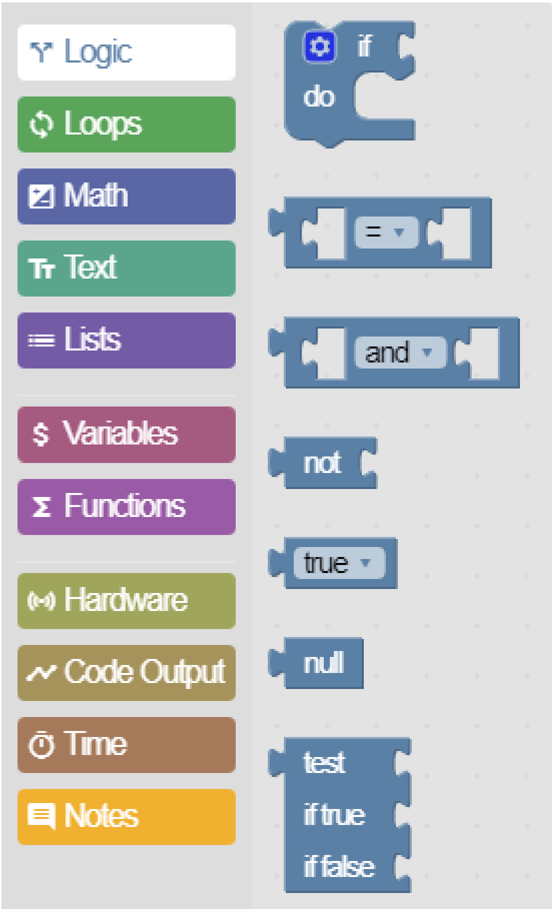
Drag blocks to the workspace to create a program.

Blockly Toolbox

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

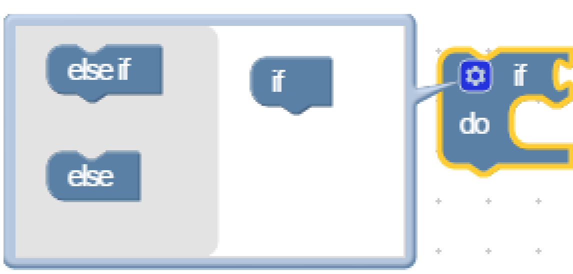
Logic

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



If Statement Block

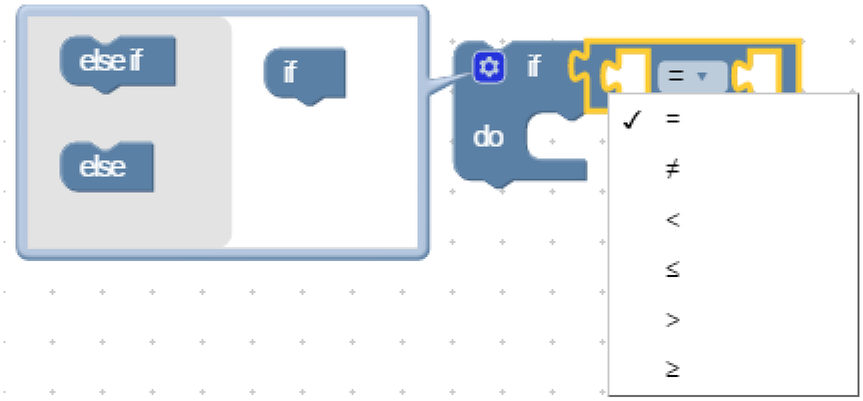
An **if** statement will do some statements if a value is true. An **else if** and **else** statement can be added to the **if** block by clicking .



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

If Statement with the Comparator Block

The comparator block returns true or false based on the comparison being done.



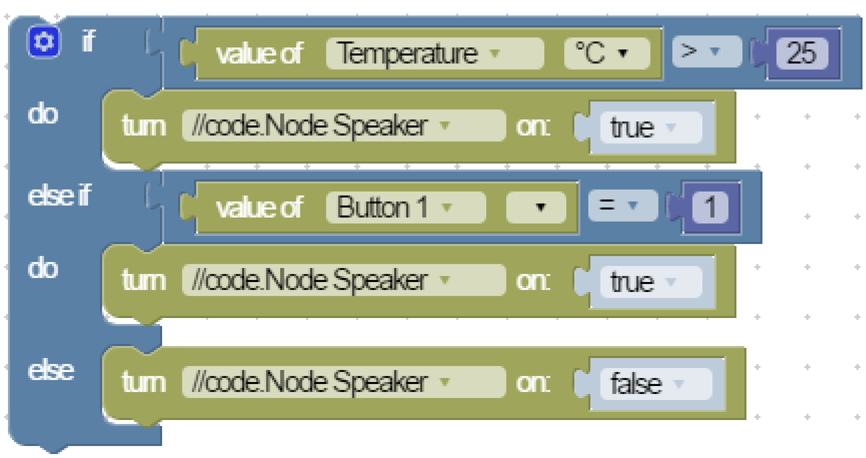
Possible comparisons include:

- 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 will turn on.

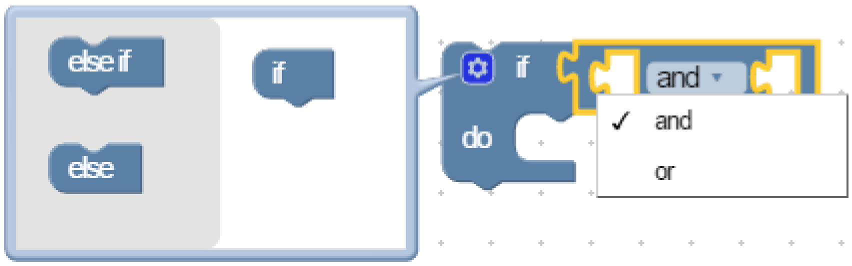
Example 2



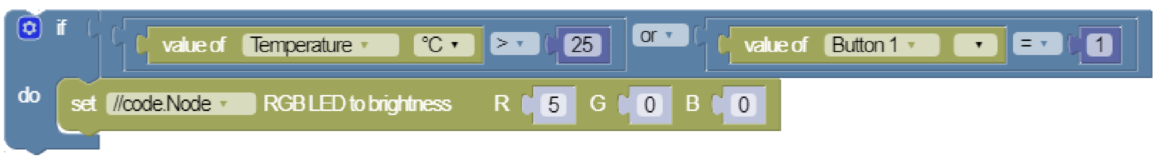
This example uses **else if** and **else** statements. If the temperature is less than 25 °C but Button 1 is pressed, the speaker will still turn on. If neither scenario is true then the speaker will be turned off.

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



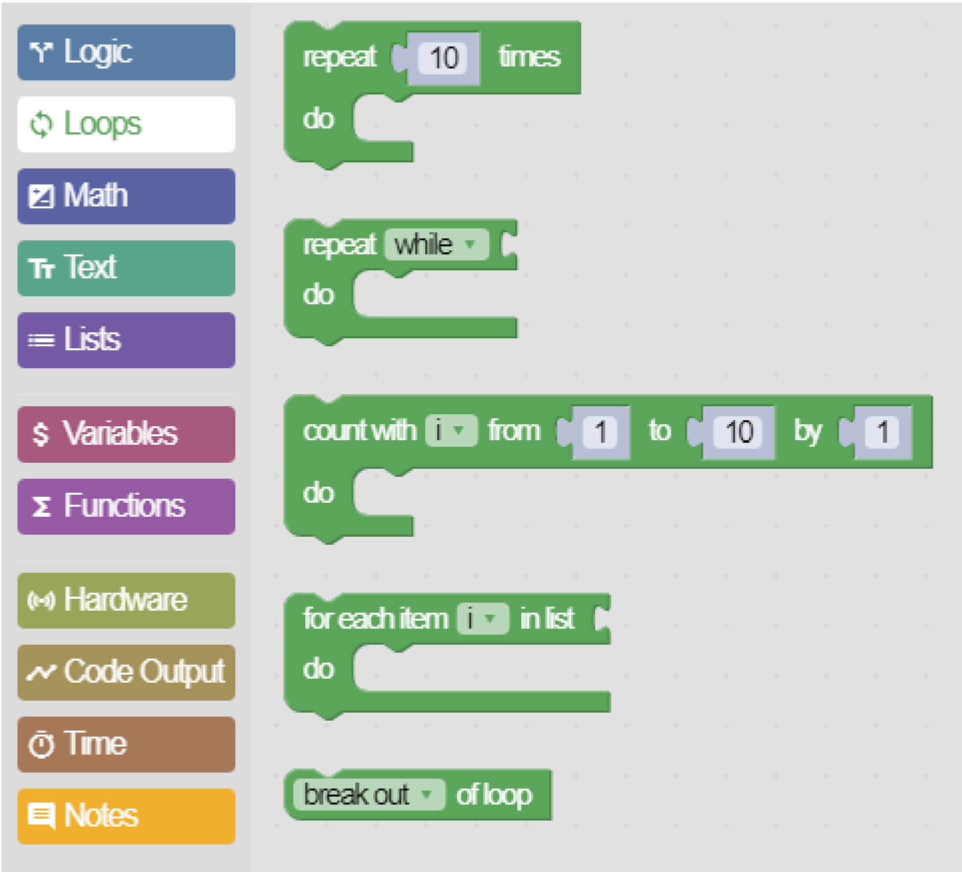
In this example, if the temperature of a sensor is greater than 25 °C *or* Button 1 is pressed, the red (R) LED will turn on.

Example 4

Adding to Example 3, if the temperature is less than 20 °C *or* Button 2 is pressed, the blue (B) LED will turn on. Otherwise, all the LEDs will turn off.

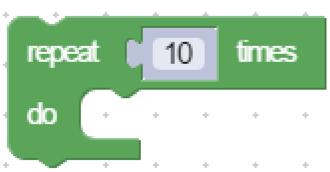
Loops

Use the loop blocks to control the number of times lines of code are executed.

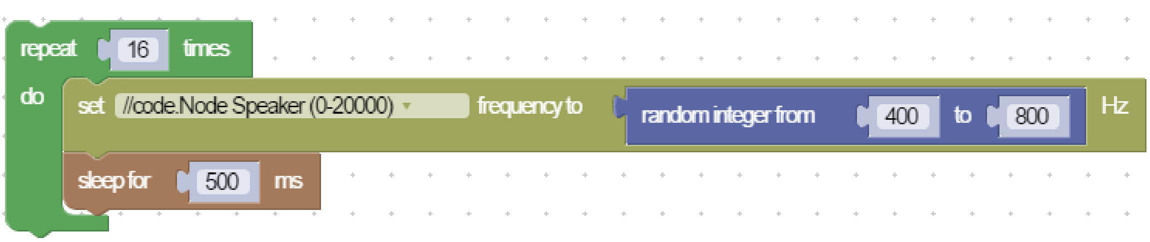


Repeat [number of] times

This block repeats code a specified number of times.



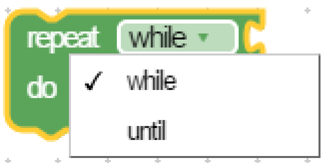
Example 1



In this example, the //code.Node speaker is set to play 16 random frequencies for 0.5 seconds each.

Conditional Repeat

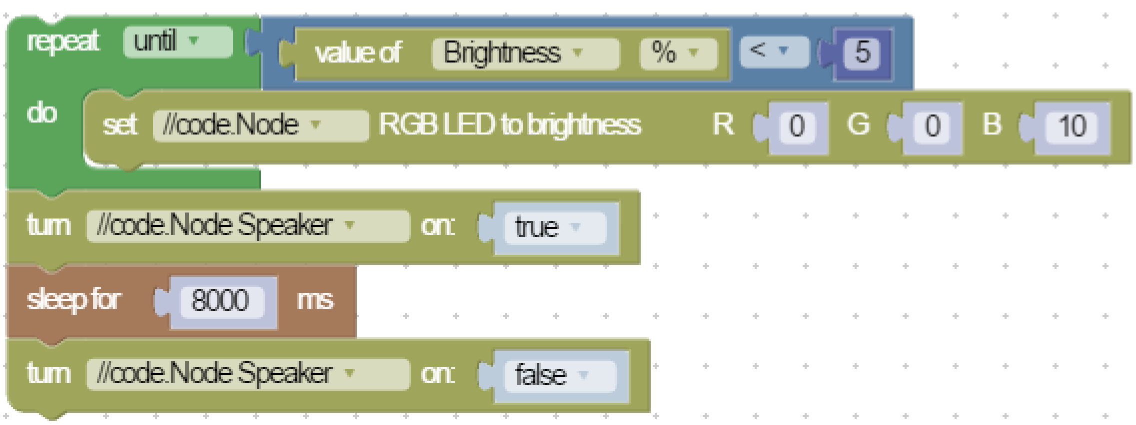
The block repeats a task *while* a condition is valid or *until* a condition is met.



Example 2: ***While*** Loop

A while loop continues as long as a condition is valid. In this example we will keep the LED set to cyan until the program is stopped.

Example 3: ***Until*** Loop

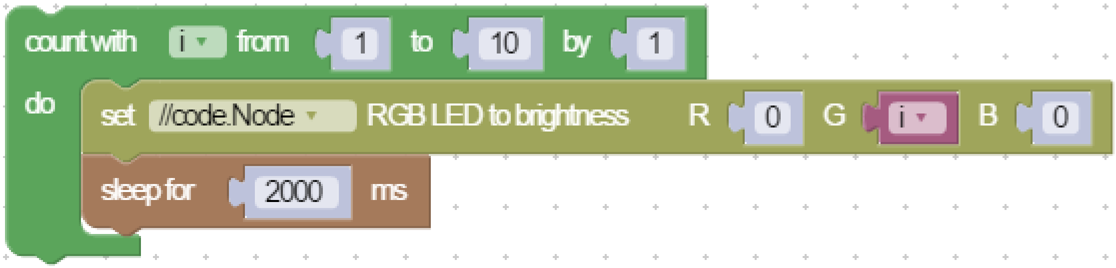


The until statement keeps looping until a condition is met. In this example the loop is keeping the blue LED on. If the light sensor detects darkness, the loop will stop and turn the speaker on for 8 seconds.

Counting Loop

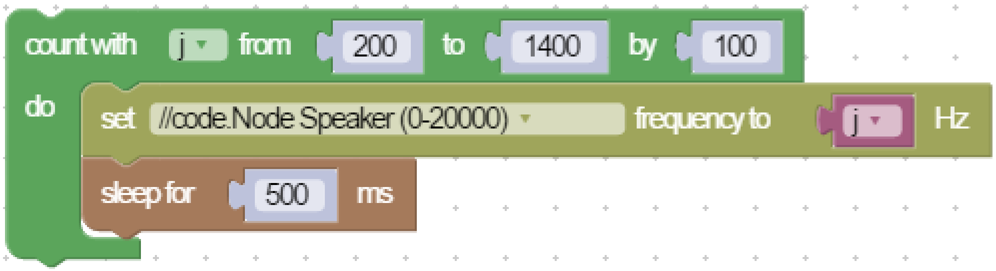
This block counts up at a specified integer until the range limit is reached.

Example 4



This loop will initially set the green LED's brightness to 1 and increment by 1 until 10 is reached. The LED will stay at each brightness level for 2 seconds.

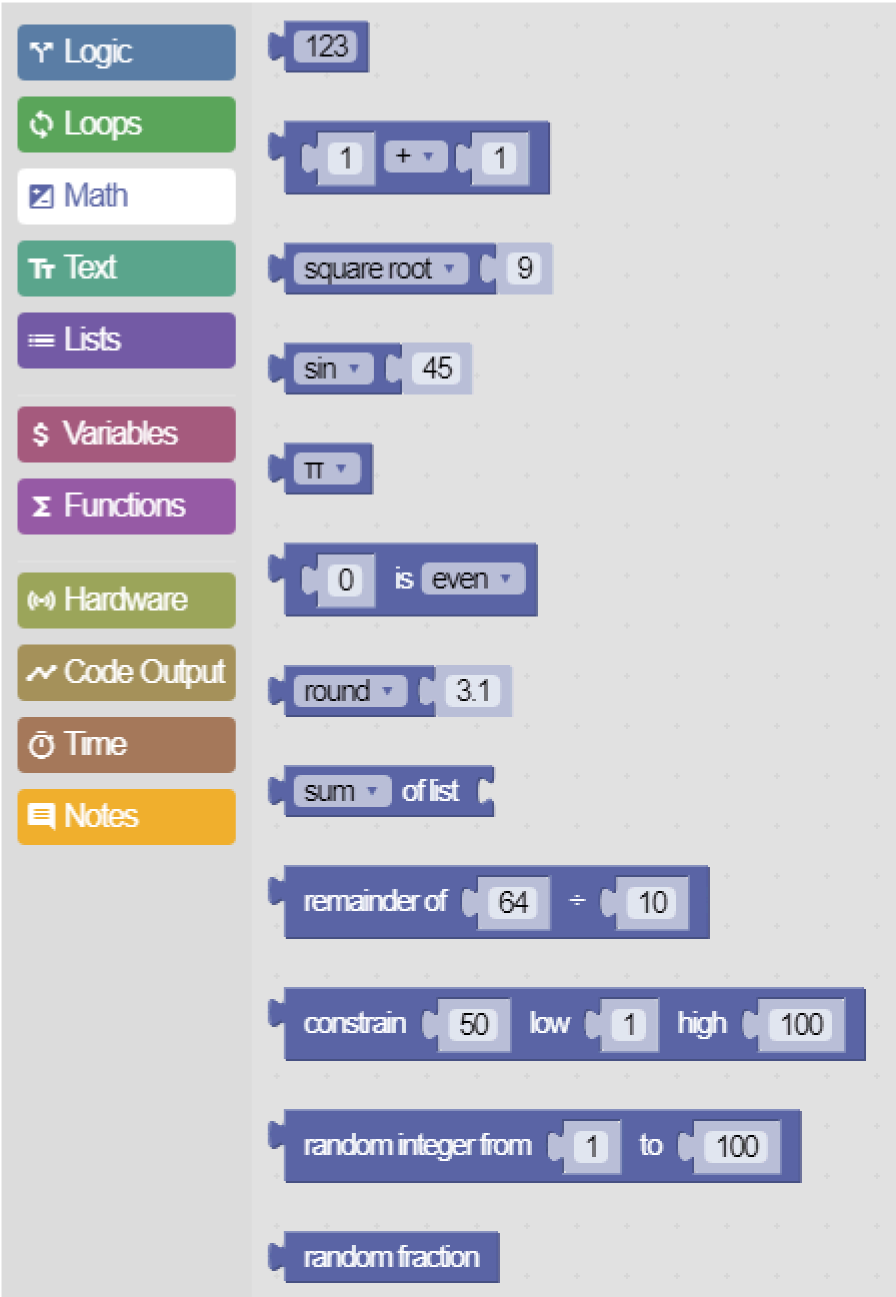
Example 5



This loop will start the speaker at a frequency of 200 Hz and in increase by 100 Hz until the loop reaches 1400 Hz. Each frequency will be played for 0.5 seconds.

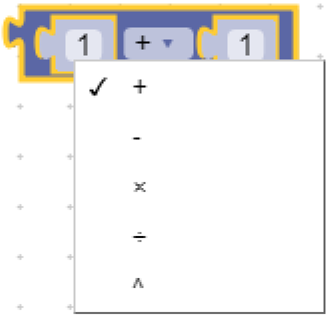
Math

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



Basic Math

Basic Operators

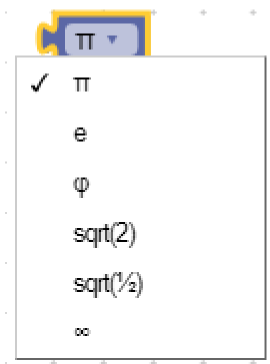


Use this block to perform basic mathematical operations on two numbers and return the result. Available operations include addition, subtraction, multiplication, division, and exponents.

Remainder

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

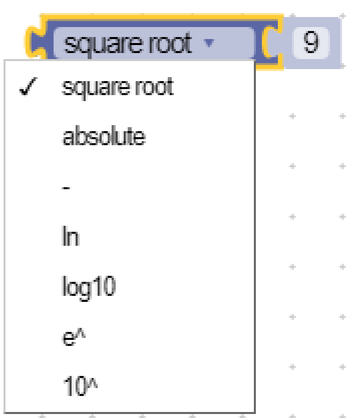
Common Constants



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.

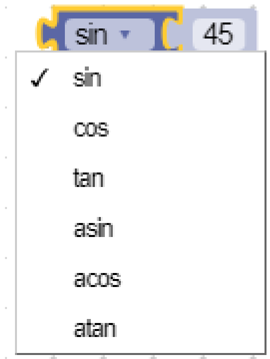
Functions

Scientific Functions



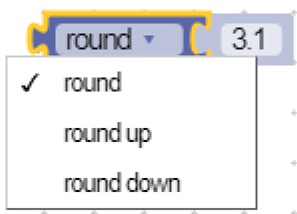
Use this block to performs a function on a single number and return the result. Available function include square root, absolute value, inverse, natural logarithm, base 10 logarithm, exponential, and power of 10.

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.

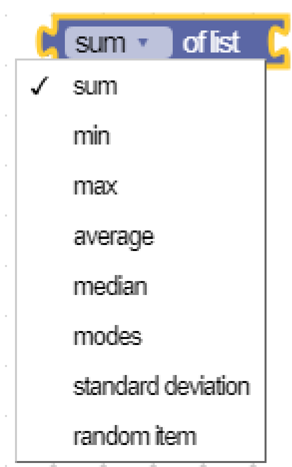
Rounding Functions



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 *(on page 104)* 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 value(s) that appears 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

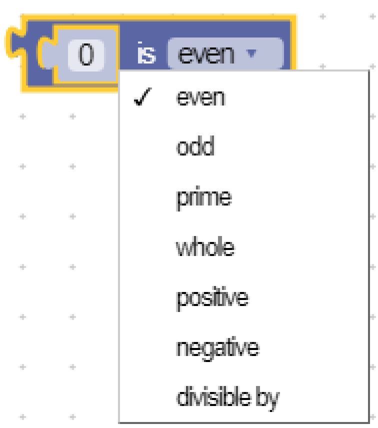
Random Integer

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



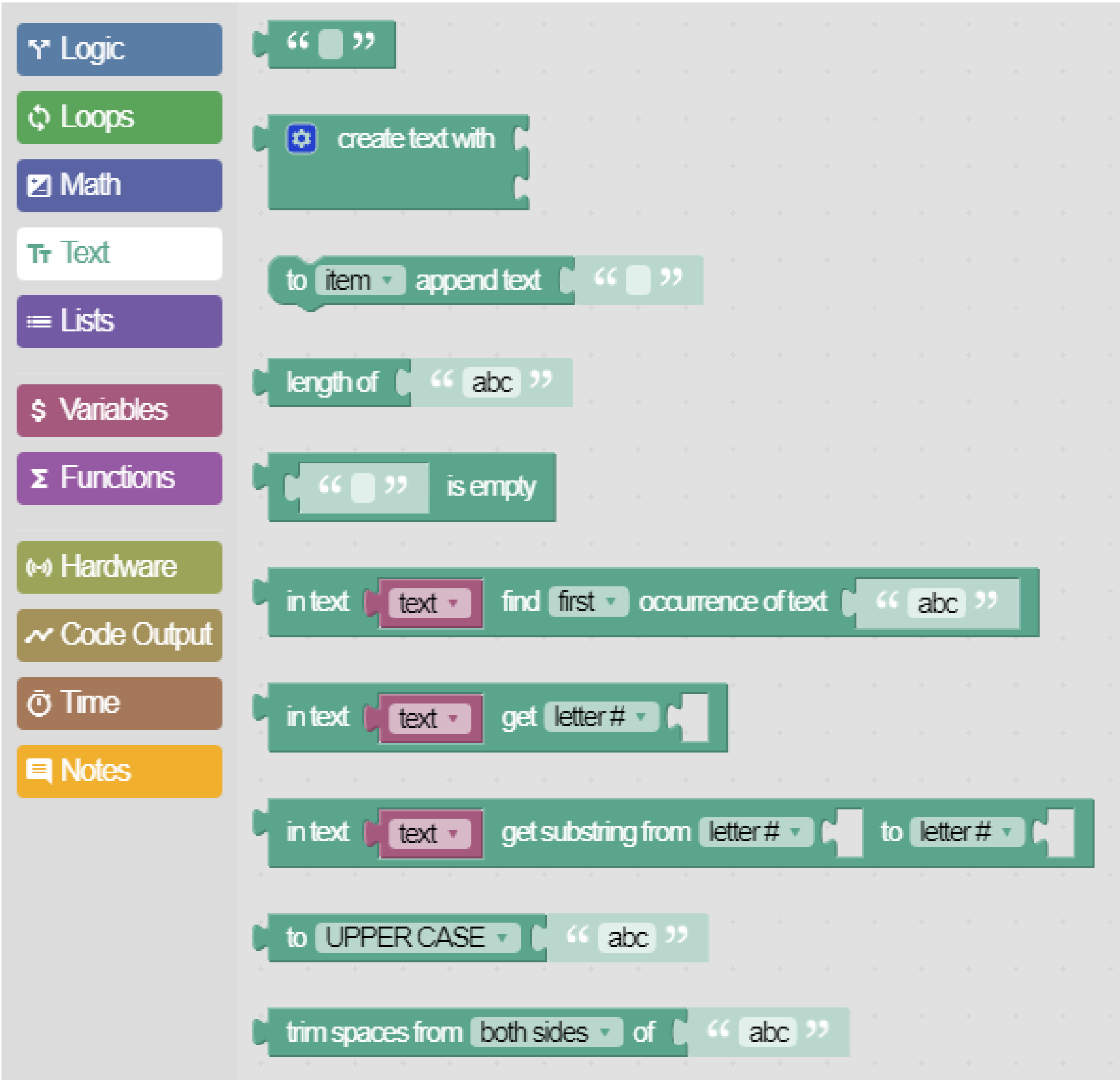
Use this block to check the type of number and return a value of `true` or `false`. The block can check if a number is even, odd, prime, whole, positive, negative, or divisible by a given number.

Constrain a Number

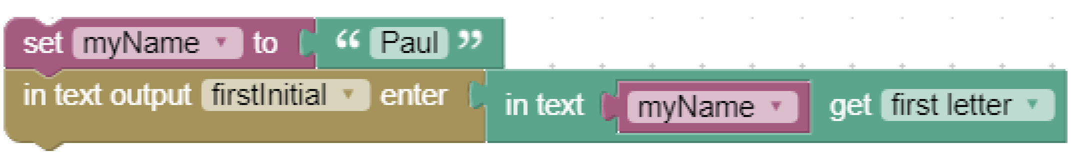
This block returns a number that is constrained between two specified values. If the number is higher than the specified high number, the block will return the high number. If the number is lower than the specified low number, the block will return the low number.

Text

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

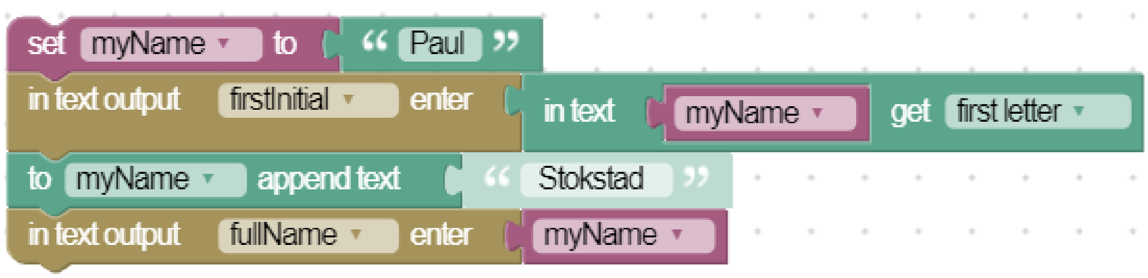


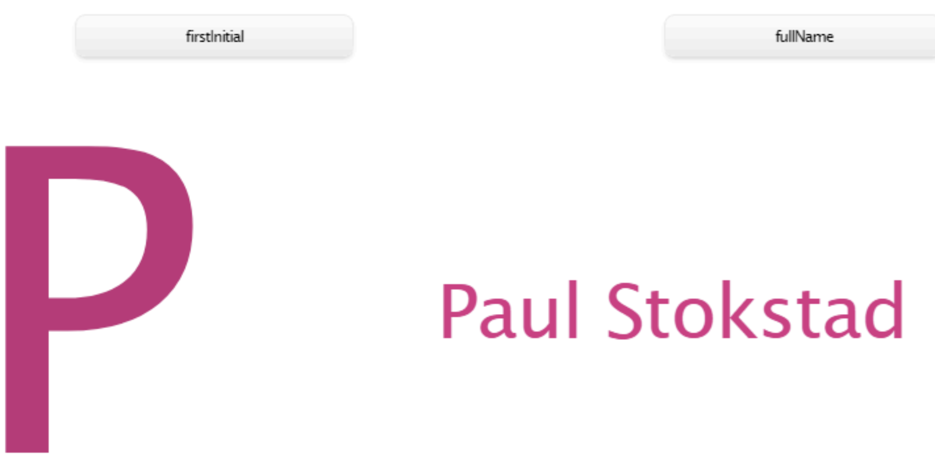
Example 1



In this example, the first line sets the variable *(on page 107)* `myName` to the string "Paul". The second line displays a text output *(on page 117)* as the first letter of `myName` in a digits display.

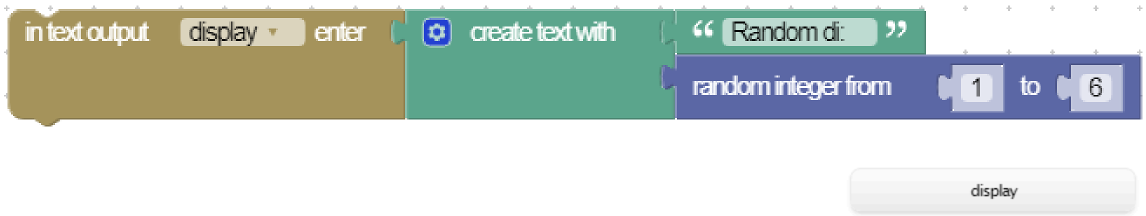
Example 2





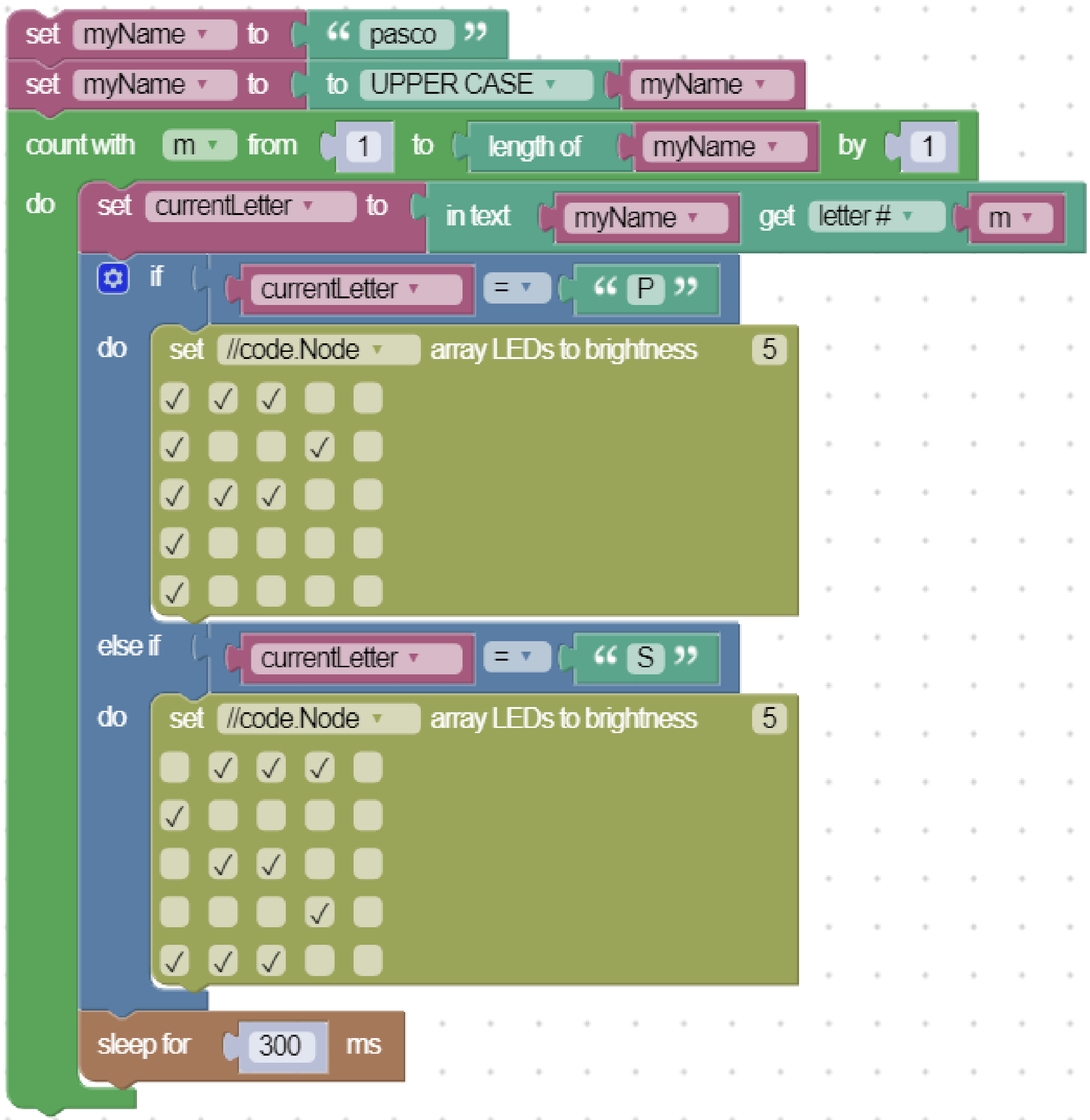
This example uses an append block that adds text onto an existing text variable. We set our initial variable `myName` to "Paul" and later appended " Stokstad" (note the space added at the beginning of the string). We then display the full name into our `fullName` Text Output block.

Example 3



The **create text with** block joins together text from two or more blocks. In this example, the string "Random di: " is joined with the **random integer** math *(on page 96)* block which displays a random number between 1 and 6.

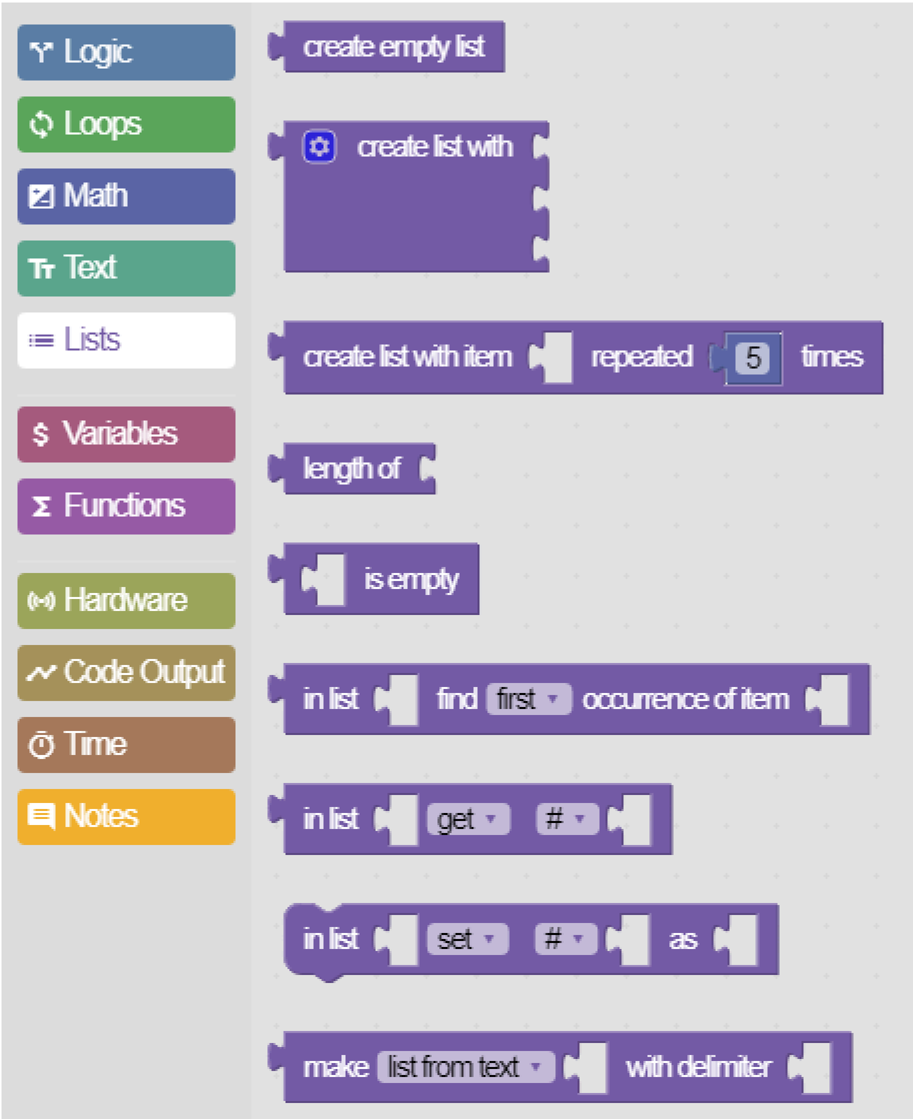
Example 4



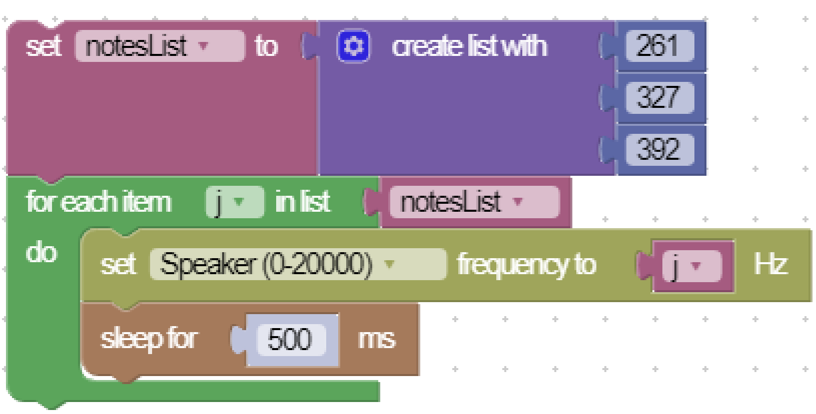
In this example the string "pasco" is changed to uppercase and displays the letters "P" and "S" on the //code.Node LED array.

Lists

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

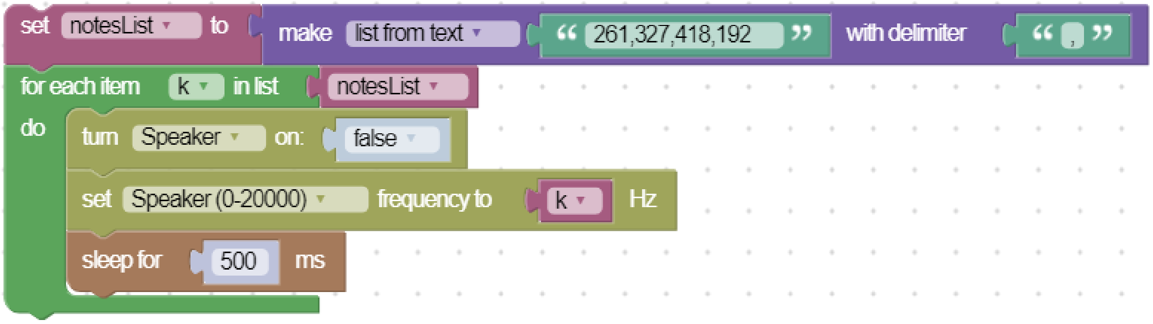


Example 1



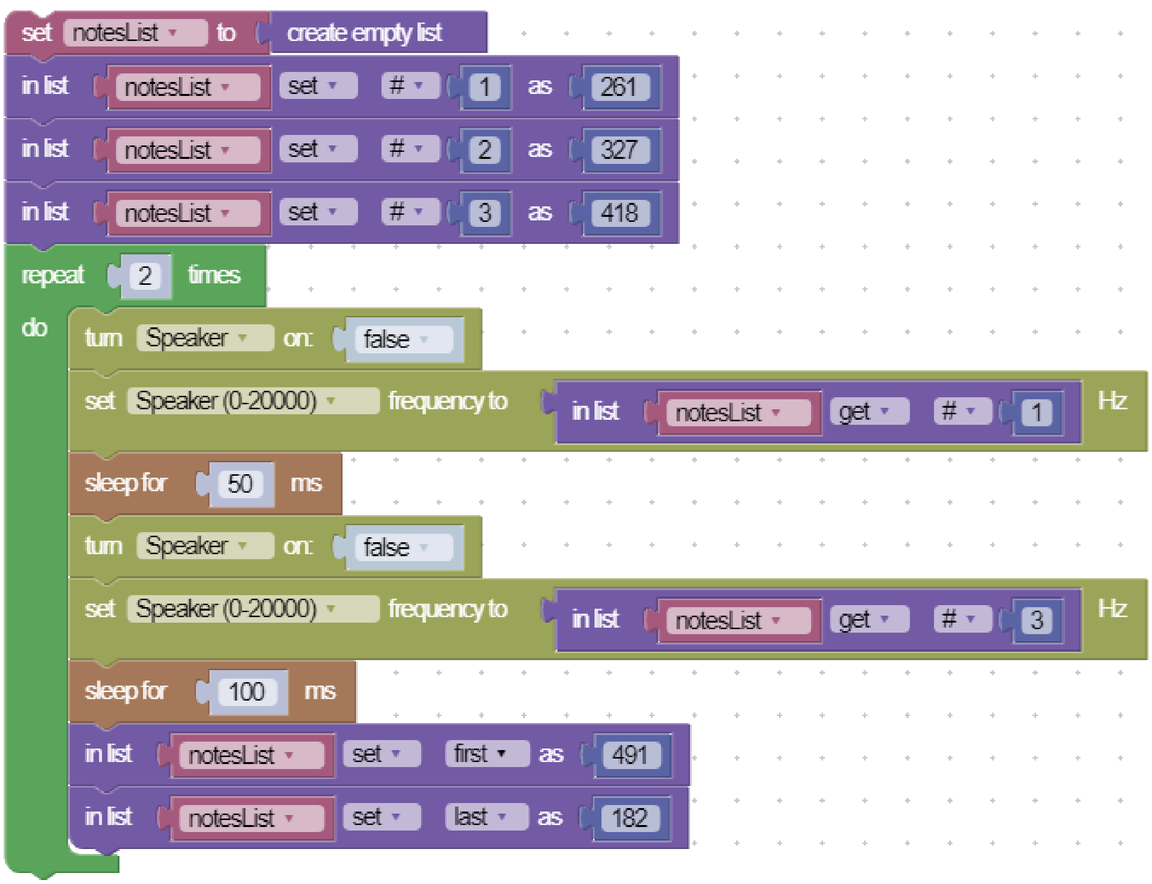
A variable called `notesList` is set to a list of three different numbers. These numbers are then used to set the speaker frequency of the //code.Node where they are played one at a time every 0.5 seconds.

Example 2



Lists can be created from a string. In this example, a list of numbers is created by separating them with a comma, which acts as the delimiter. The end result is that the //code.Node speaker outputs the frequencies of 261, 327, 418, and 192.

Example 3

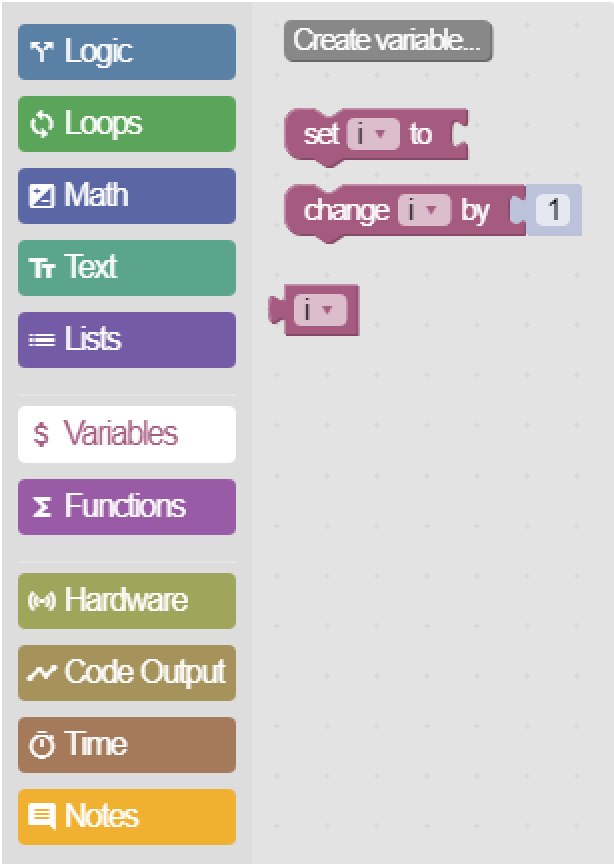


A variable can be set from a list of numbers one value at a time. In this example, a variable is set to a list containing three different integers. In the loop, the //code.Node speaker is set to play the frequencies of the #1 and #3 values in the list, change the first and last values of the list (which happen to be #1 and 3), and play those values again. The speaker output will sound different than the first round because we changed the frequencies after going through the loop one time.

The end result is that the //code.Node speaker outputs the frequencies 261, 418, 491, and 182, each for 0.1 seconds.

Variables

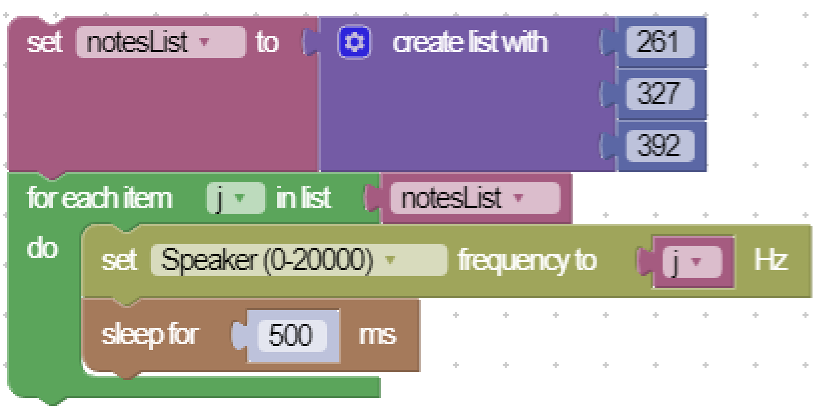
Use the Variable blocks to store a value that can be referred to later.



Example 1

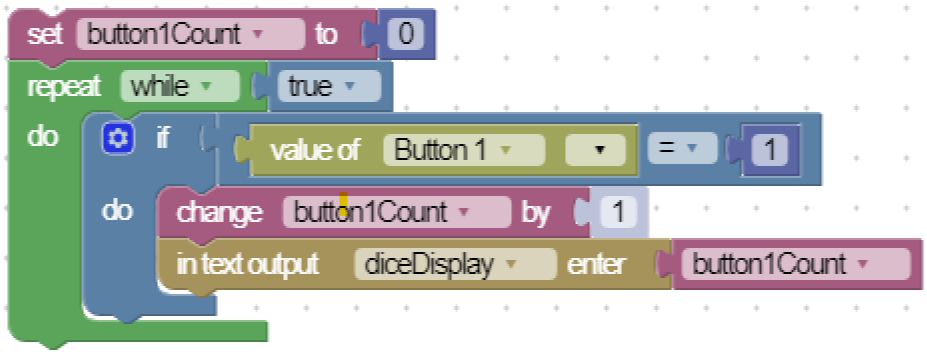
In this example, the variable called `diceValue` is created and is set to a random number between 1 and 6. The value of `diceValue` is then displayed in a text output called `diceDisplay`. The text output is viewed using a Digits display.

Example 2



This example contains two variables. The first variable is called `notesList` which stores a list *(on page 104)* of numbers. The second variable is called `j` which is defined as the item in each list. The first time the loop circles through, `j` is 261, the second time it is 327 and the third it is 392.

Example 3

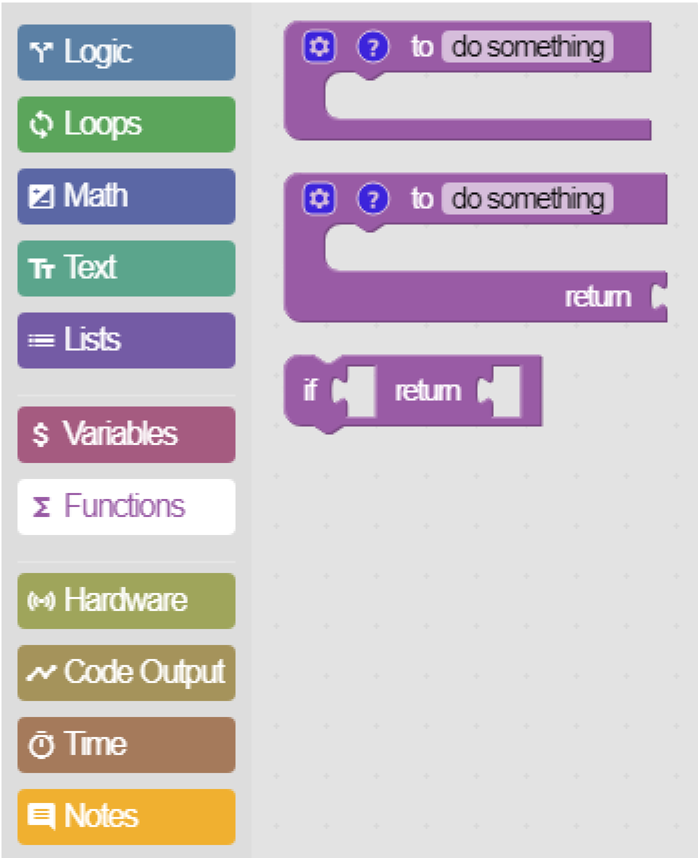


If a variable is a number, the **change by** block can be used to increase or decrease the value of the variable. In this example, the variable `button1Count` increases by 1 each time Button 1 on the //code.Node is pressed.

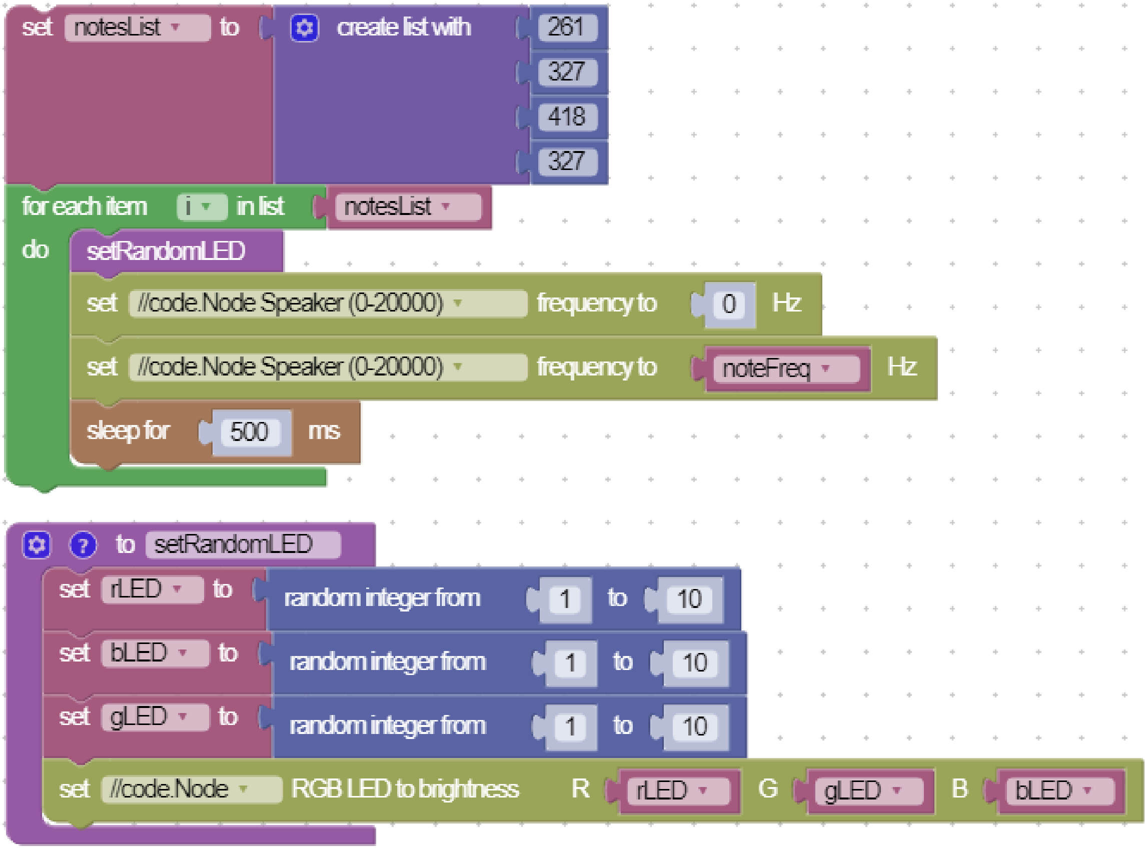
> **Note:**
> The value of `button1Count` is set to 0 at the beginning of the code. Blockly does not assume that a variable starts at 0.

Functions

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

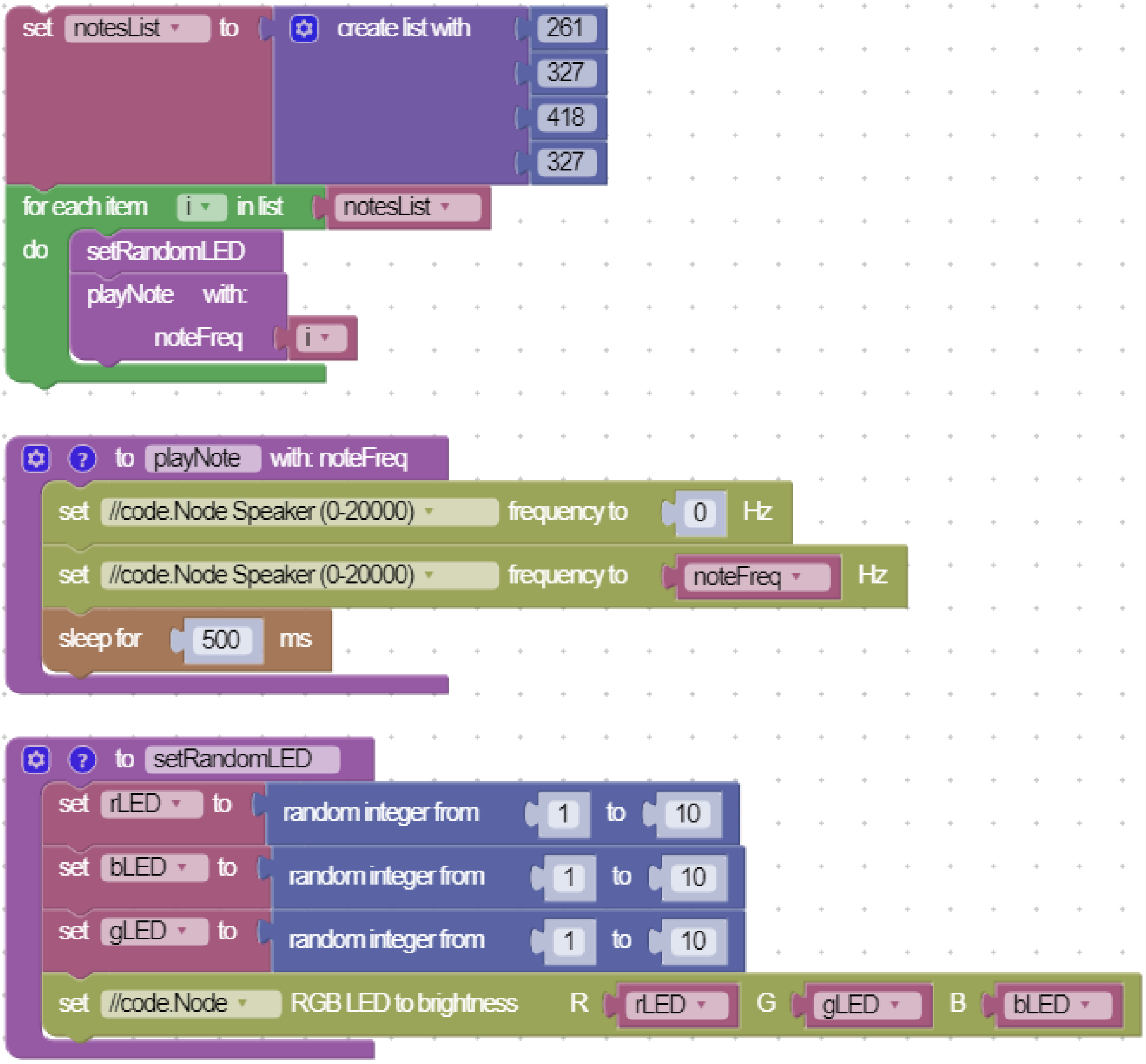


Example 1: Simple function



Functions are used to store repeatable tasks and make code easier to read. In this example, we have a function named `setRandomLED` that sets the //code.Node RGB LED to a random color. By putting this code in a separate function our main code block is easier to read.

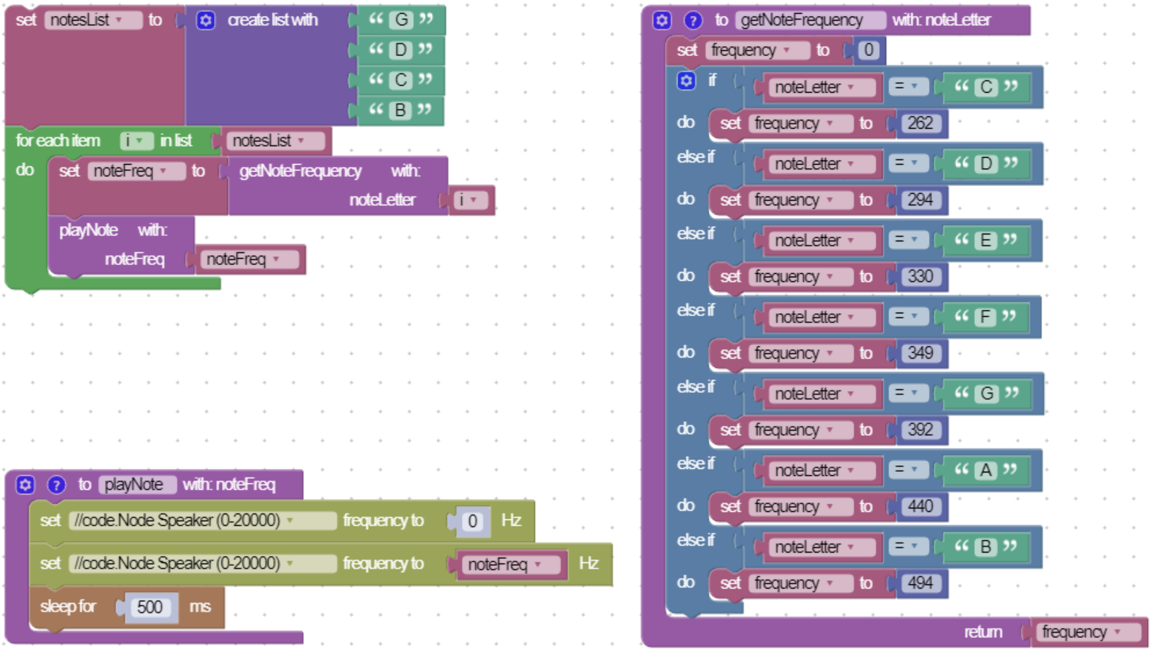
Example 2: Passing variables into a function



Example 1 can be taken a step further by also creating a function to play a musical note. In order to play a note, a value needs to be placed in the function so the program knows which note to play. This is accomplished by selecting properties  on the function block and adding the **input name** block to the function.

In the loop, the note that is to be played is stored in the `i` variable which gets its value from the `notesList` variable. A function called `playNote` is created which takes an input called `noteFreq`. The `i` variable is passed into this function. The function turns off the speaker (if it is running) and plays each note in the list for 0.5 seconds.

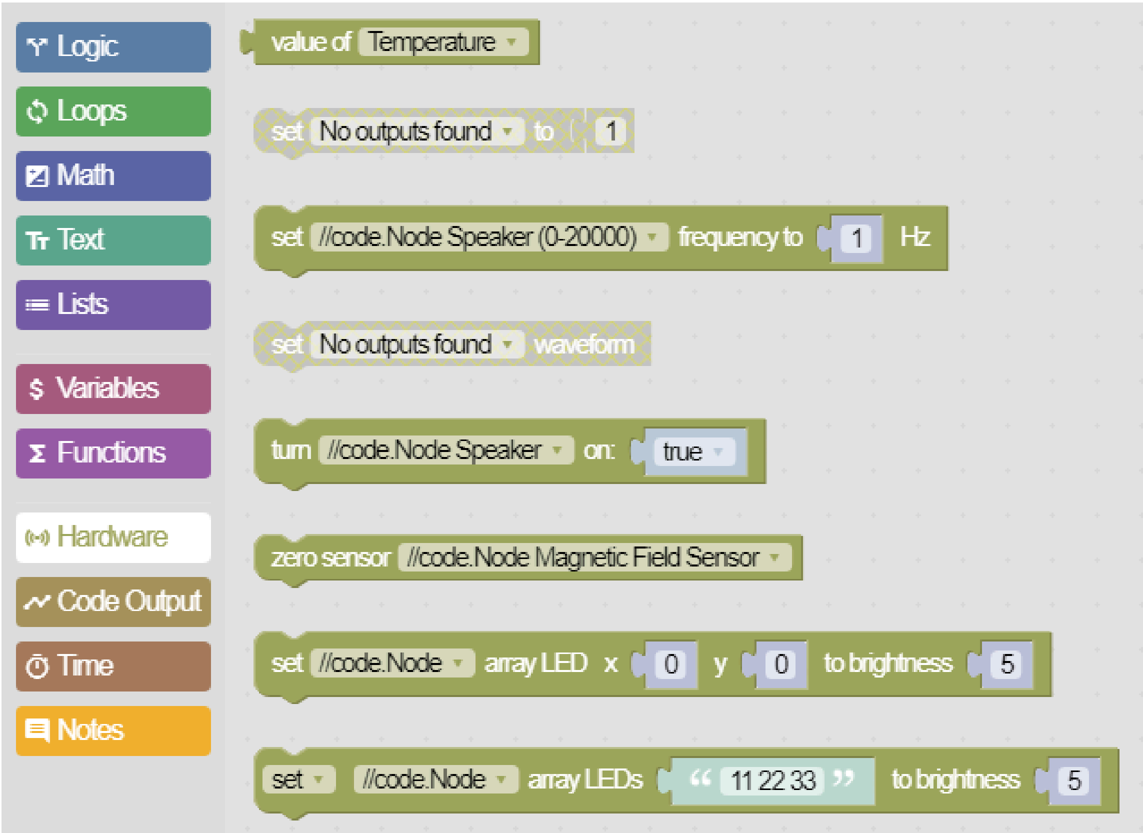
Example 3: Returning values from a function



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 is passed into the `playNote` function to play the note.

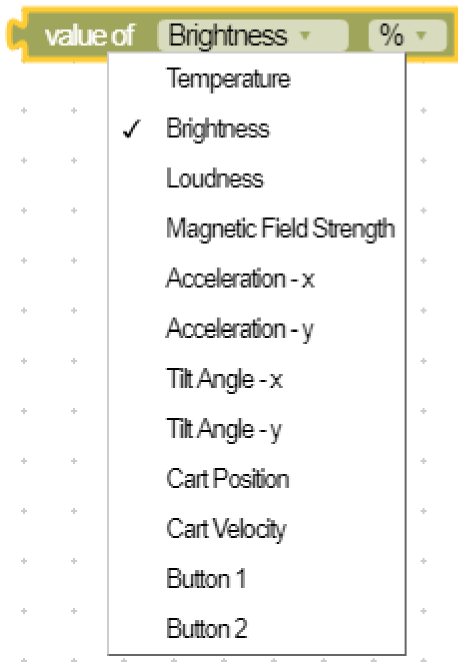
Hardware

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



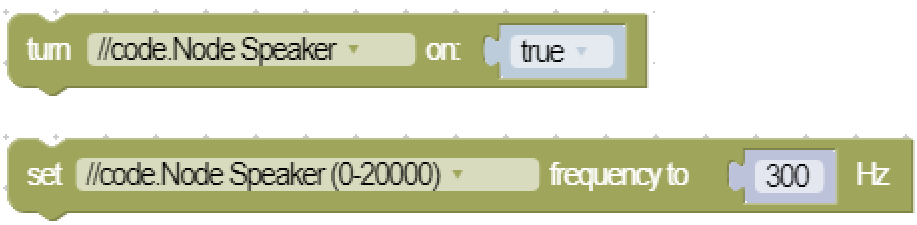
Reading Sensor Values

The **value of** block is used 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.



//code.Node Speaker

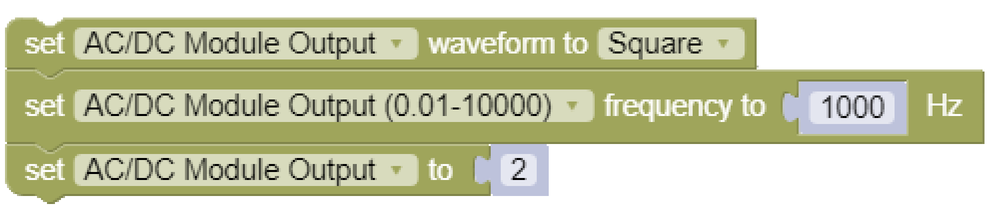
Two blocks are available to control the speaker on a //code.Node. One block sets the speaker frequency and the other simply turns it on (`true`) or off (`false`). You can also turn off the speaker by setting the frequency to 0.



Signal Generator

The same blocks used to control the //code.Node speaker can be used to control the signal generator on the 850 Universal Interface, 550 Universal Interface, and the Wireless AC/DC Module. Two additional blocks allow you to set the signal generator waveform and voltage.

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



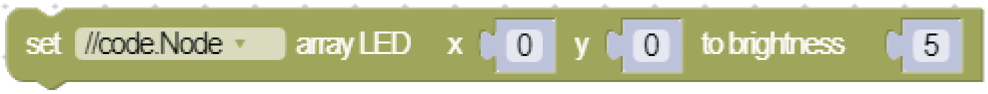
Zero Measurements

The **zero sensor** block is used to zero a measurement of a sensor.

//code.Node LED Array

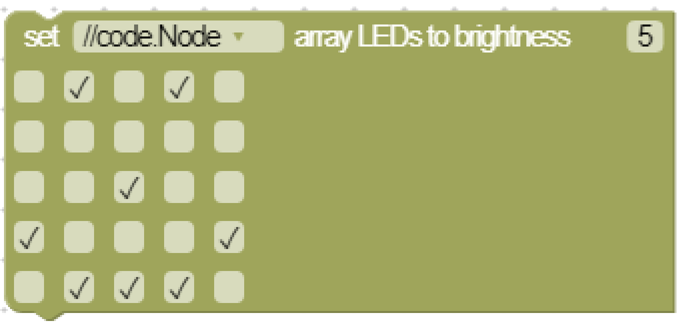
Three blocks are available to set the brightness of each LED in of the //code.Node LED array. The brightness of each LED set on a scale from 0 to 10. An LED in the array is identified by its *x* and *y* coordinate. The top left of the array is defined as the origin (0,0). The *x* coordinate corresponds 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. The coordinate of each LED is defined by two consecutive numbers, where the first number is the *x* coordinate and the second number is the *y* coordinate. Each LED listed is separated by a space. In the following example, a diagonal line is formed 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.



//code.Node RGB LED

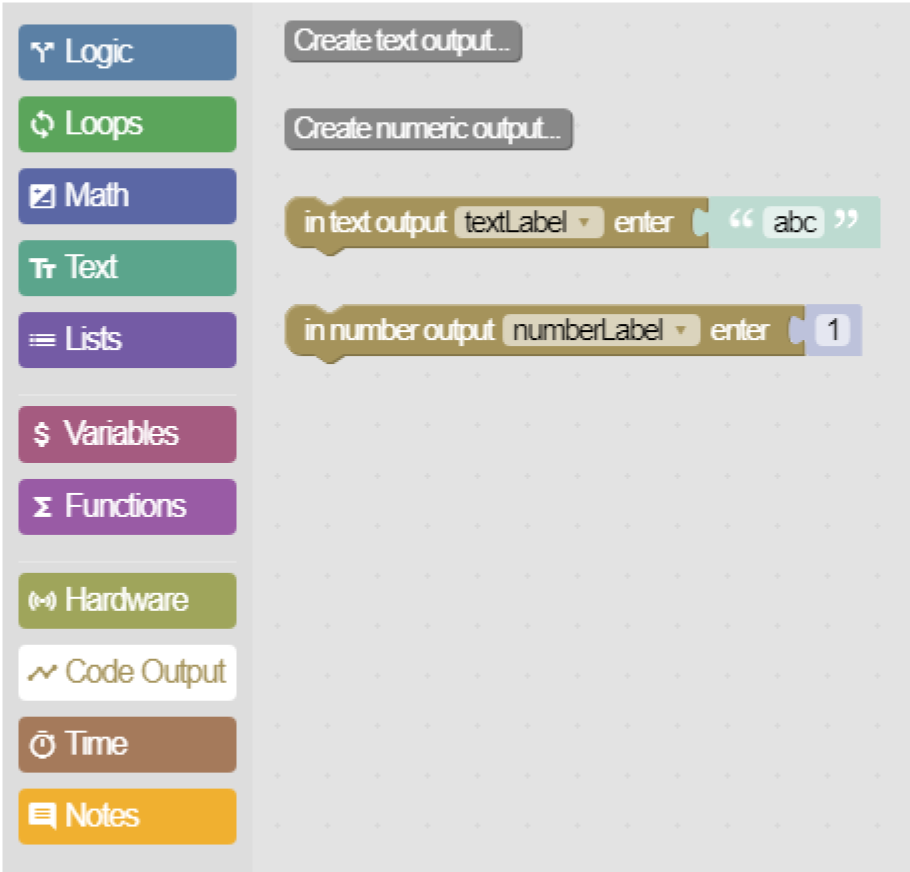
The color of the //code.Node RGB LED can be controlled by setting the brightness of the red (R), green (G), and blue (B) LEDs.

Smart Cart Ballistic Cart Accessory

The release of a projectile from the Smart Ballistic Cart Accessory can be controlled using the **release projectile** block.

Code Output

Use the Code Output blocks to provide text or numeric output in a Digits *(on page 47)* or Table *(on page 42)* display.

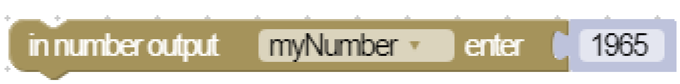


Text Output

This block outputs a string of text.

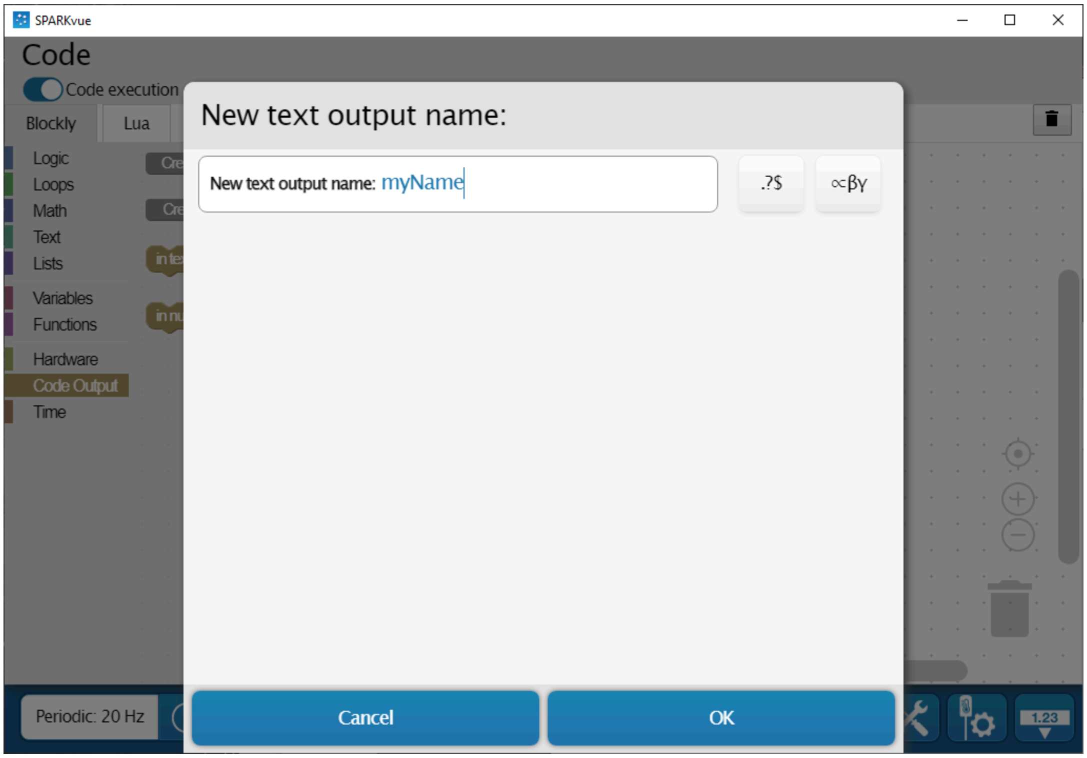
Numeric Output

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

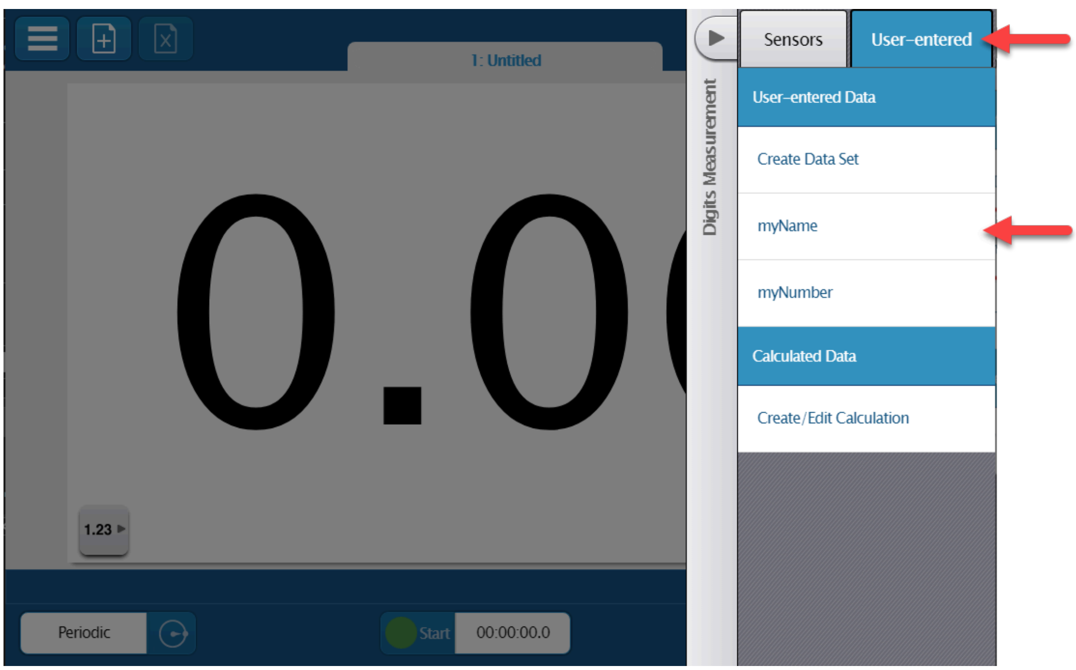


Create a Text Output

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

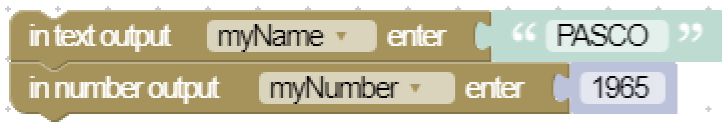


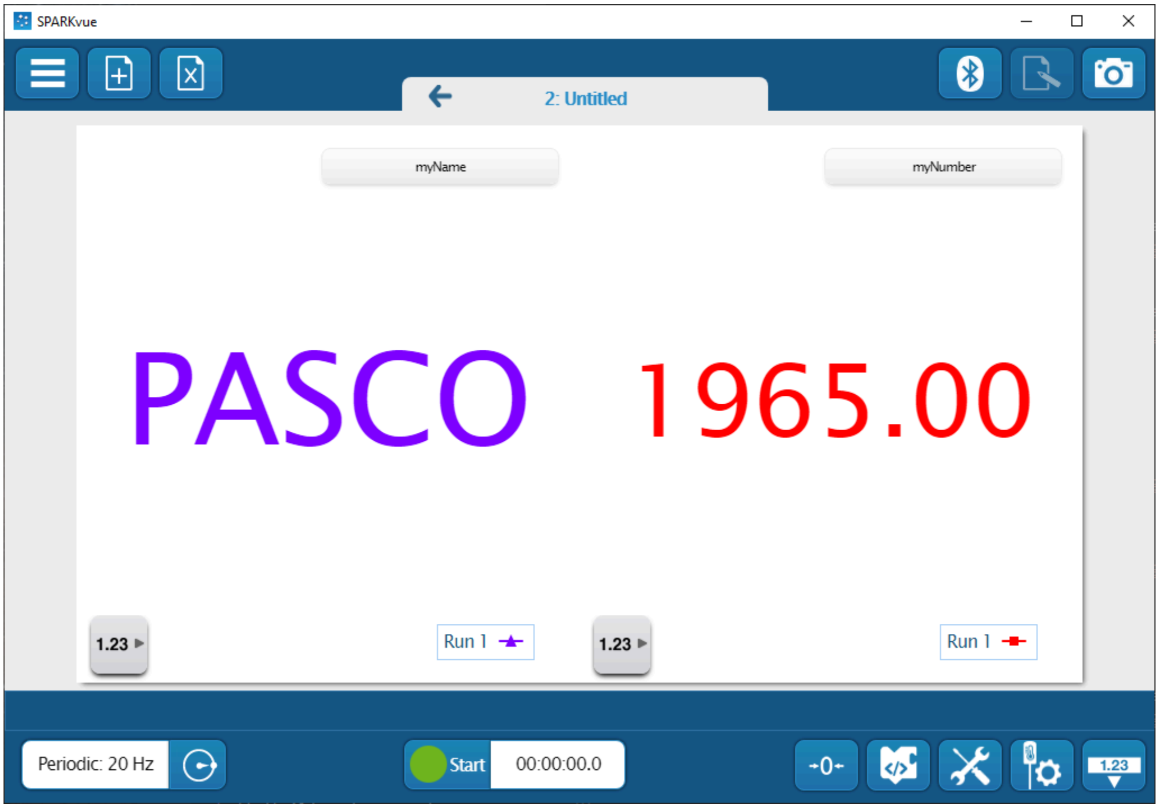
2. Close the Code tool.
3. Click , select a template, then select the Digits display **1.23**.
4. Click **Select Measurement**, select the **User-entered** tab, then select the name of the text output.



Example 1: Displaying Text Versus Numbers

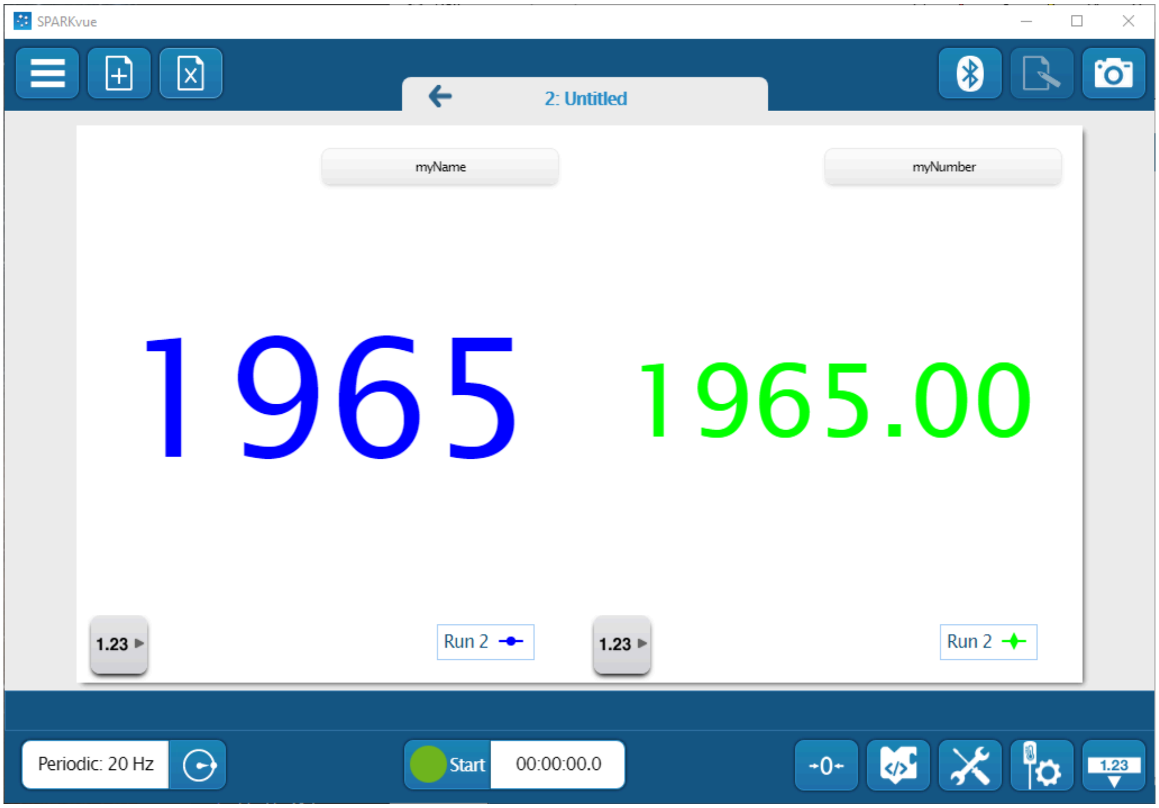
This example contains two Digits displays with one showing text output and the other shown numeric output.





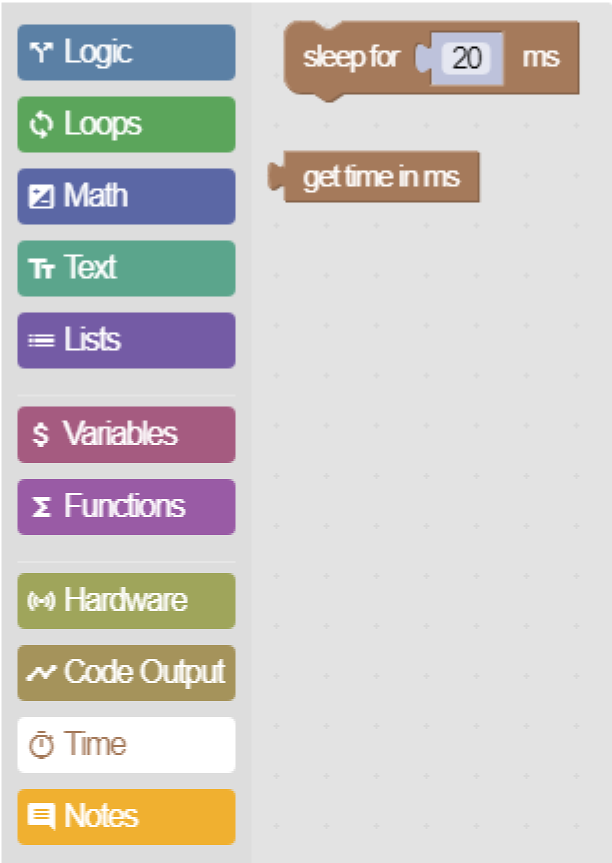
Example 2: Displaying Numbers in Text Output

It is possible to display numbers in text output. However, the numbers will be interpreted as a string of text instead of a quantity of measurement, which means that mathematical operations cannot be applied on the output. When showing a number, you can see the difference between a text and number display.

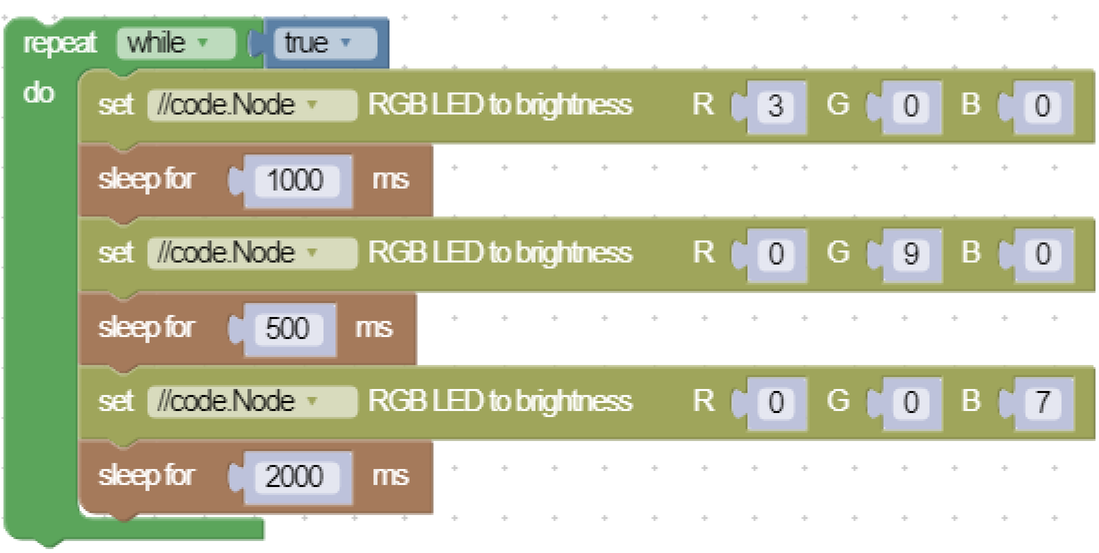

SPARKvue
2: Untitled
myName
myNumber
1965
1965.00
1.23
Run 2
1.23
Run 2
Periodic: 20 Hz
Start
00:00:00.0
1.23

Time

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

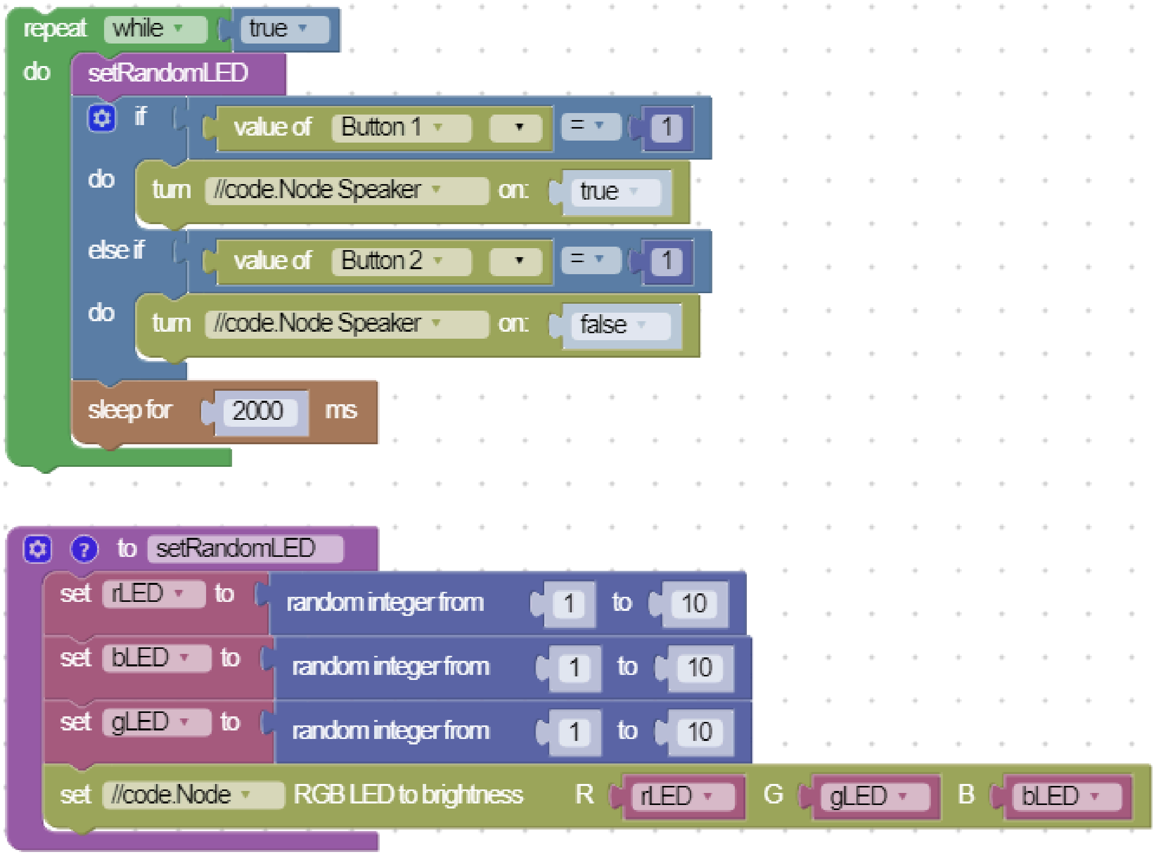


Example 1: Using Sleep Blocks



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

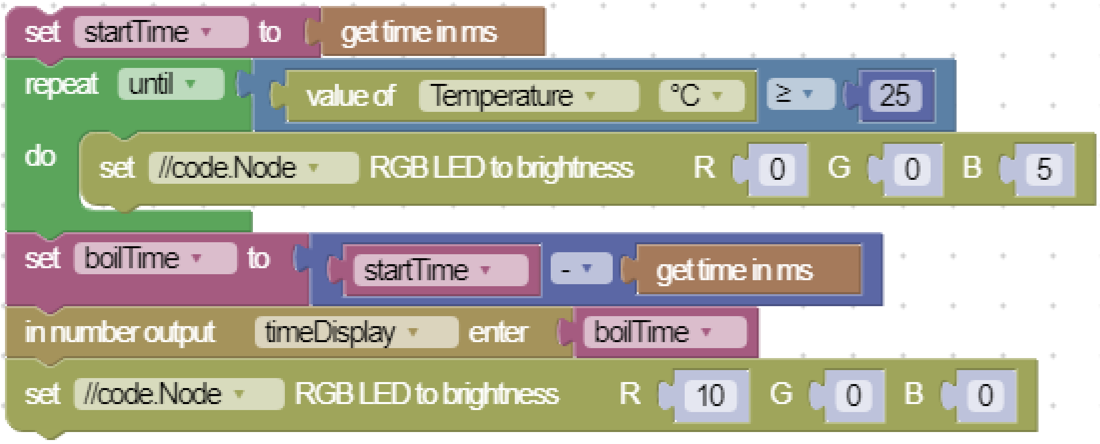
Example 2: Sleep Block Causing an Error



Say we 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 Button 1 is pressed, the speaker may or may not turn on. This is because the code only checks the button state *after* every two seconds (when the LED changes). If the button is not pressed down during that check then the code will not turn on the speaker. To get the speaker to turn on Button 1 must be pressed and *held* so that when the LED changes the state of Button 1 state will be `true`. To turn the speaker off, we must press and hold Button 2 so that it is held down the same time the LED changes.

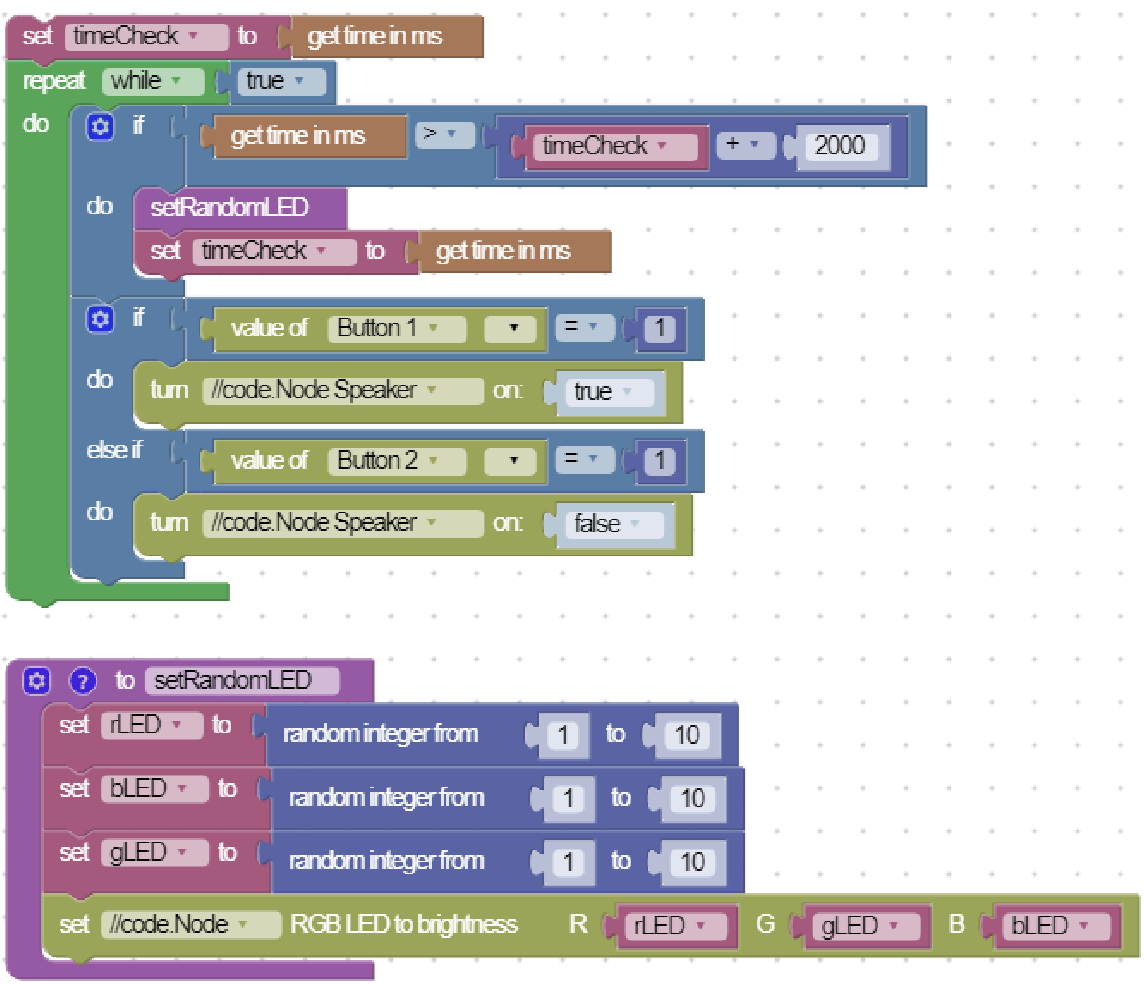
Tip:
See Example 4 *(on page 123)* for a method to fix this problem.

Example 3: Getting the Instantaneous Time



We can use the **get time in ms** block to get a timestamp of when an event happens. In our example we can use the block to determine how long it takes for the temperature to reach 25 °C. The time can be displayed using the **number output** block from the Code Output *(on page 117)* group.

Example 4: Perform Synchronous Functions

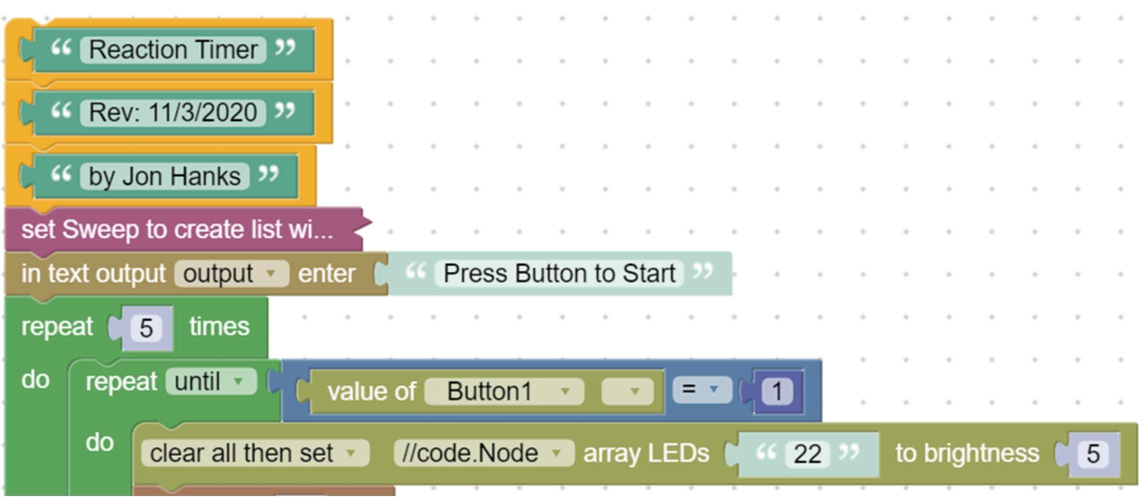


We can use the **get time in ms** block if we want tasks to happen every *x* amount of seconds . In this example, we set a variable `timeCheck` to the time stamp at the start of the program. We then use an if statement that has a condition where `timeCheck` will be set to the current time stamp every two seconds and also run the `setRandomLED` function *(on page 110)*. At the same time, a separate conditional statement is set so that the //code.Node speaker will turn on by pressing Button 1 or turn it off by pressing Button 2. The loop does not have to stop before turning the speaker on or off. This does not work with the sleep blocks as see in Example 2 *(on page 122)*.

Notes

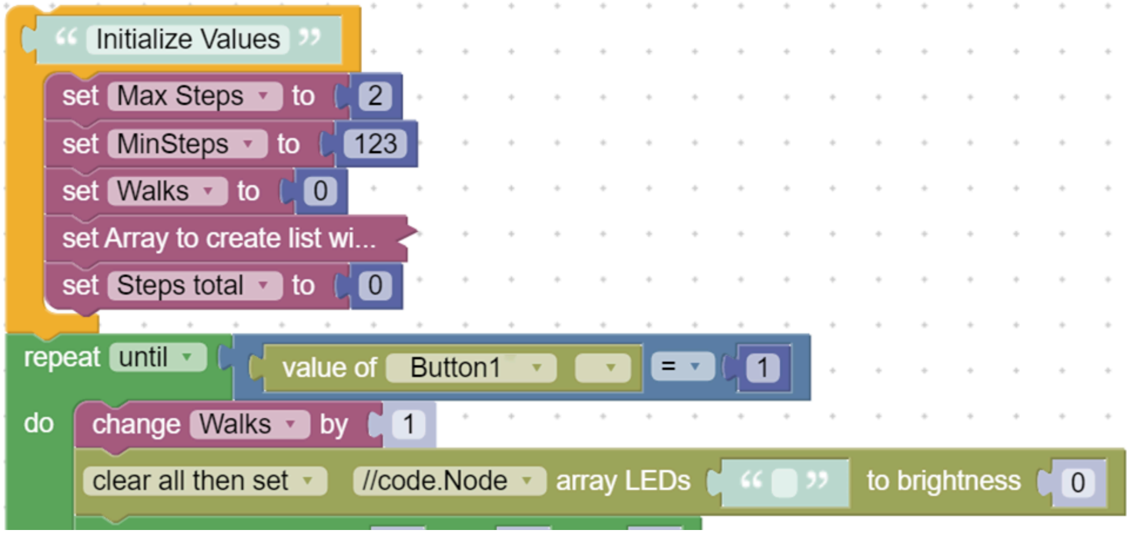
Use the Notes blocks to add comments to your code that are ignored by your program. Insert a single note between blocks or associate a note with a group of blocks.

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 above or below a block.

Block Group Notes

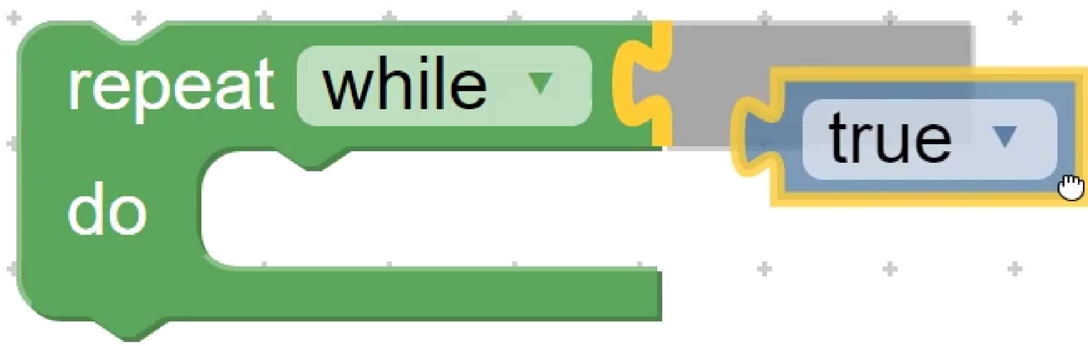


Block Group notes work like single line notes, except that you can associate multiple blocks with a note. Another useful feature of the block group note is being able to move a group of blocks by clicking and dragging the note block.

Build Code Using Blockly

Create programs by dragging and dropping blocks onto the Code tool workspace.

Build Code



The blocks are shaped like puzzle pieces to provide hints on how the blocks fit together. However, the blocks will only join together if the code is valid. To build a program:

1. Select a Block category from the menu.
2. Click and drag a block onto the workspace.
3. Click and drag a second block near an existing block until you silhouette of the block connecting to the existing block, then release.
4. Repeat this process for multiple blocks to build your program.
5. Click Done when finished.

Modify Blocks

Some blocks contain features which allow you to modify the parameters of the block.

Properties 

Allows you to modify the properties of a block. Click the icon to see the options.

Comments 

Allows you to add a comment or note to a block. In order to get this to appear, right-click (or press and hold) on the block and select Add Comment.

Drop-down Menu 

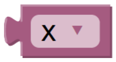
Some blocks have a drop down menu to change the behavior of the block. Click the down arrow to see the list of selections.

Enter a Numeric Value 

Blocks that look like this allow you to enter numeric values. Click on the block and enter the desired value.

Enter a Text String 

These blocks are used for text strings. Click the blank space and enter in any alphanumeric character.

Variables 

A magenta block with a drop down menu represents variables. Click the menu to select a different variable, rename the variable, or define a new variable.

Delete Blocks

Delete a single block or a group of blocks by dragging and dropping the blocks to the workspace trash . Click the trash to retrieve previously deleted blocks.

Delete all the blocks on the screen by clicking the trash button .

Disable Blocks

Disabling blocks can be useful when you want to test your code by removing blocks from the code without permanently deleting them. To disable blocks, right-click on the block then select Disable Block. On phones and tablets, press and hold on the block until the menu appears then select Disable Block.

Run a Program

Execute a Blockly program created in the Code tool.

To run a program, go the Experiment Screen and click .

When sensors are connected to SPARKvue, code is executed while recording data. Data recording will stop once SPARKvue reaches the end of the program code. You can also stop the code by clicking .

Note:

If sensors are not connected, the Start button changes to Execute . This provides the ability to run code without having sensors connected.

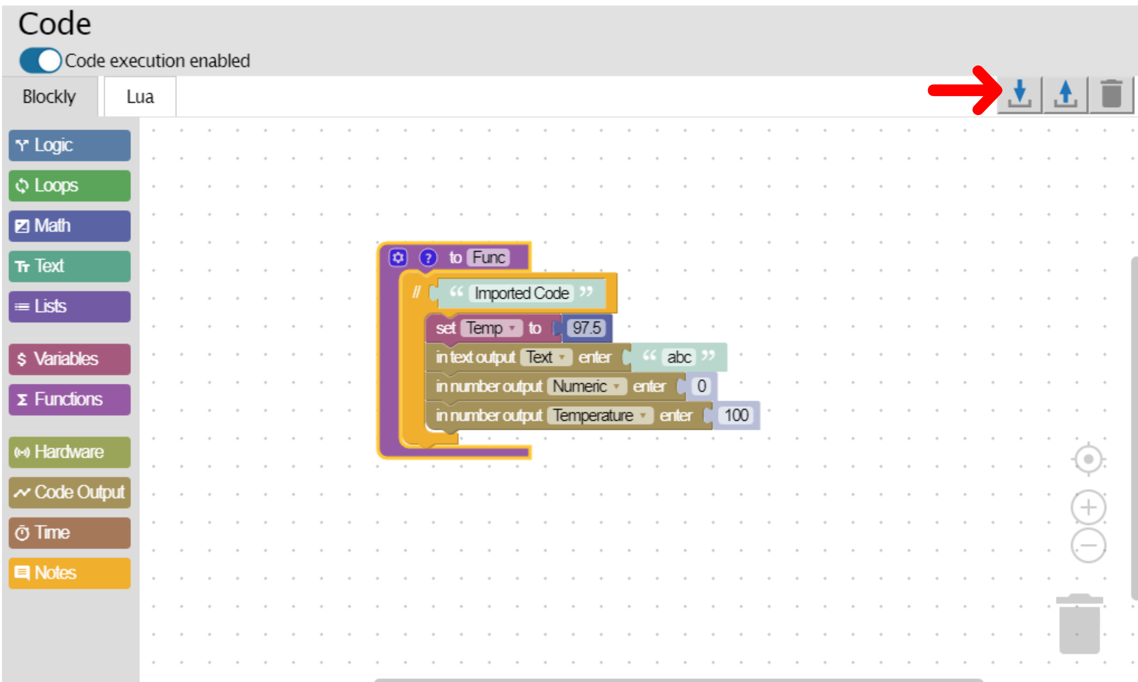
Tip:
If you do not want code to be executed during data collection, disable code execution by toggling the switch at the top of the Code tool.

Import and Export Code

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

A program created in the Code tool *(on page 86)* can be copied into another file by using the import and export tools at the top of the workspace. The Blockly code is exported as a .pcbx file which can be imported into a SPARKvue or Capstone file.

> **Note:**
> This feature is not 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 is 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 will be 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 will be renamed.