Getting Started with the Mantis Climate Sensor

Contents

Document Propose

This document will explain how to use the Mantis Climate Sensor with the CPALMS Code app to first time users

Prerequisites

Bluetooth Dongle

The Bluetooth dongle must be connected in order for Scratch projects to interact with the Mantis Sensors. If you are having issues connecting to the dongle see the document entitled “GettingStartedWithCPALMS-code” and go to sections “Bluetooth Dongle” and “Bluetooth Dongle Troubleshooting”.

Enable Flash

Flash must be enabled in the browser in order for the Scratch project to run. Instructions on how to enable Flash in your particular browser can be found on the CPALMS-Code App site at:

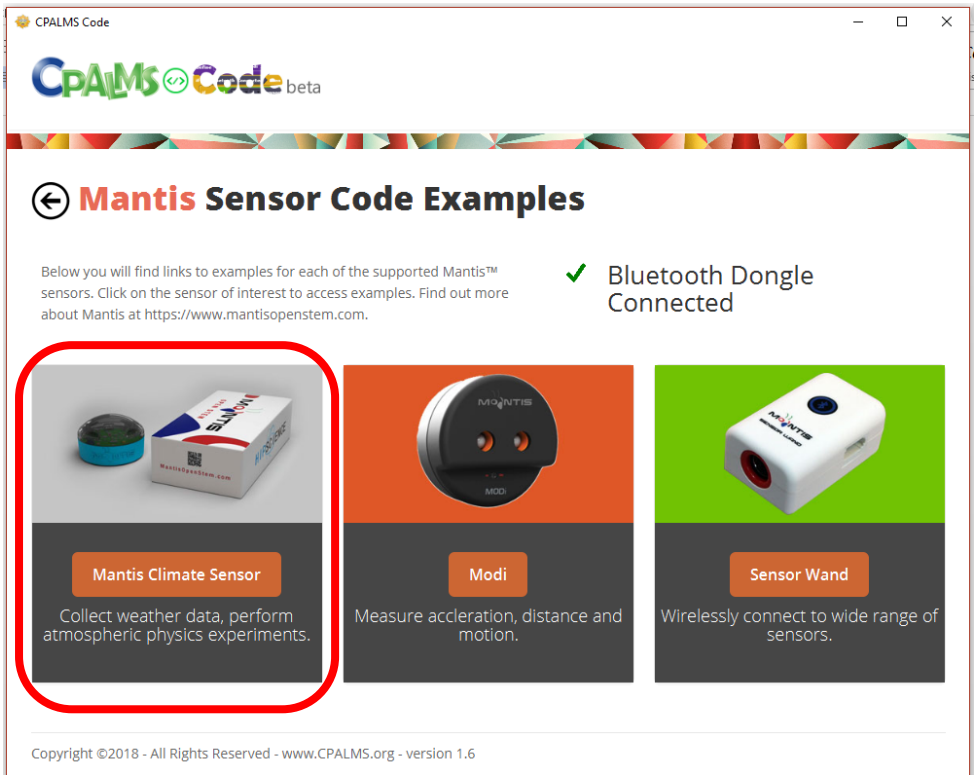
Windows: http://www.cpalms.org/page793.aspx

Mac: http://www.cpalms.org/page792.aspx

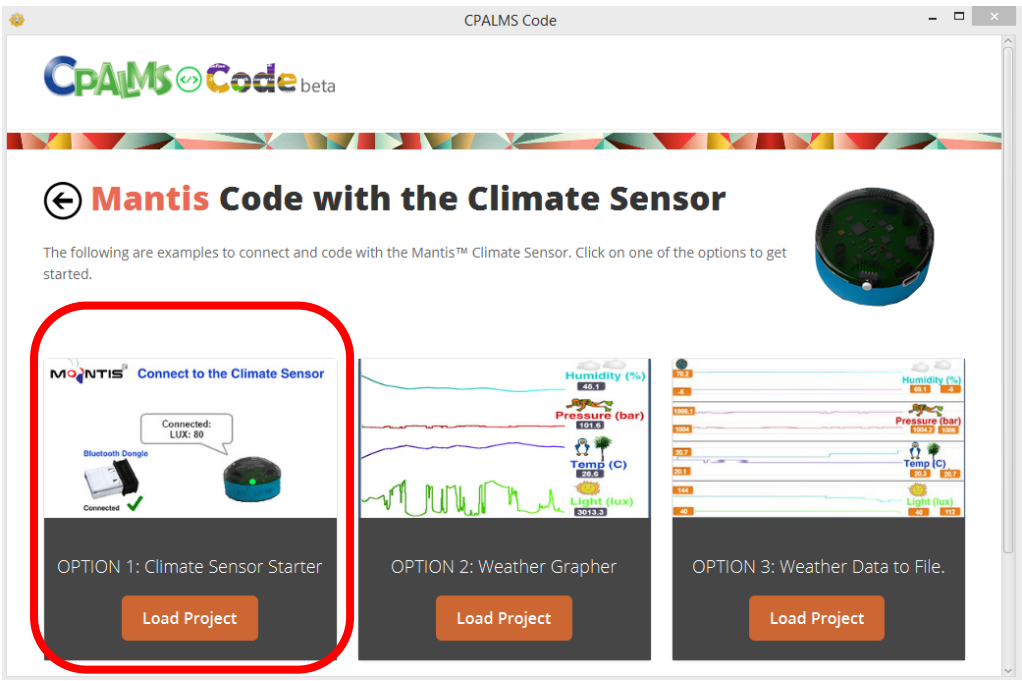
The Scratch project will load in the system default browser in a separate window or tab. Be sure your system default browser is set to Chrome, Firefox, Opera, or Safari (Mac). *The Edge browser will not work well.*

Running the Starter Project

From the CPALMS-Code App home page click on Mantis Sensors panel. Choose the Mantis Climate Sensor option from the Mantis Sensor Code Examples page:



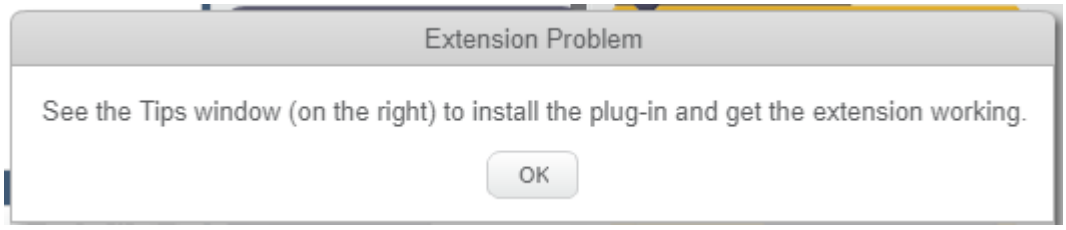
From within the Climate Sensor page choose Option 1: Climate Sensor Starter:



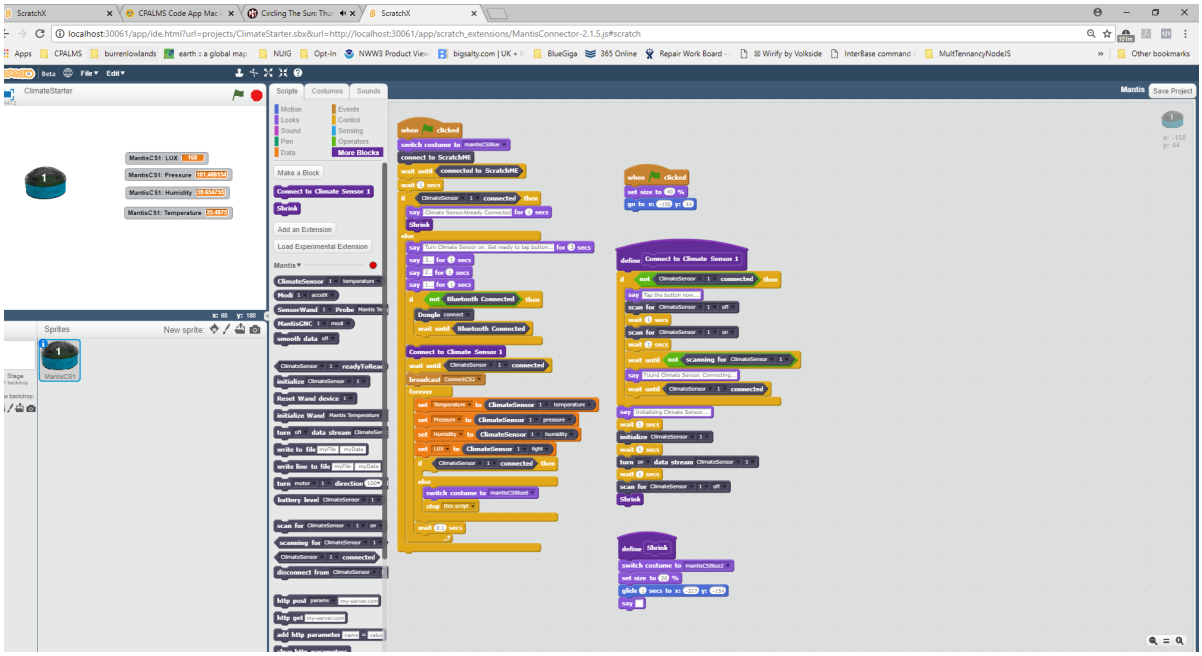
The project will open up in your system default browser in a separate window:

Click to continue:

When the project loads, click OK on the next error dialog. It is safe to ignore, there is no error:



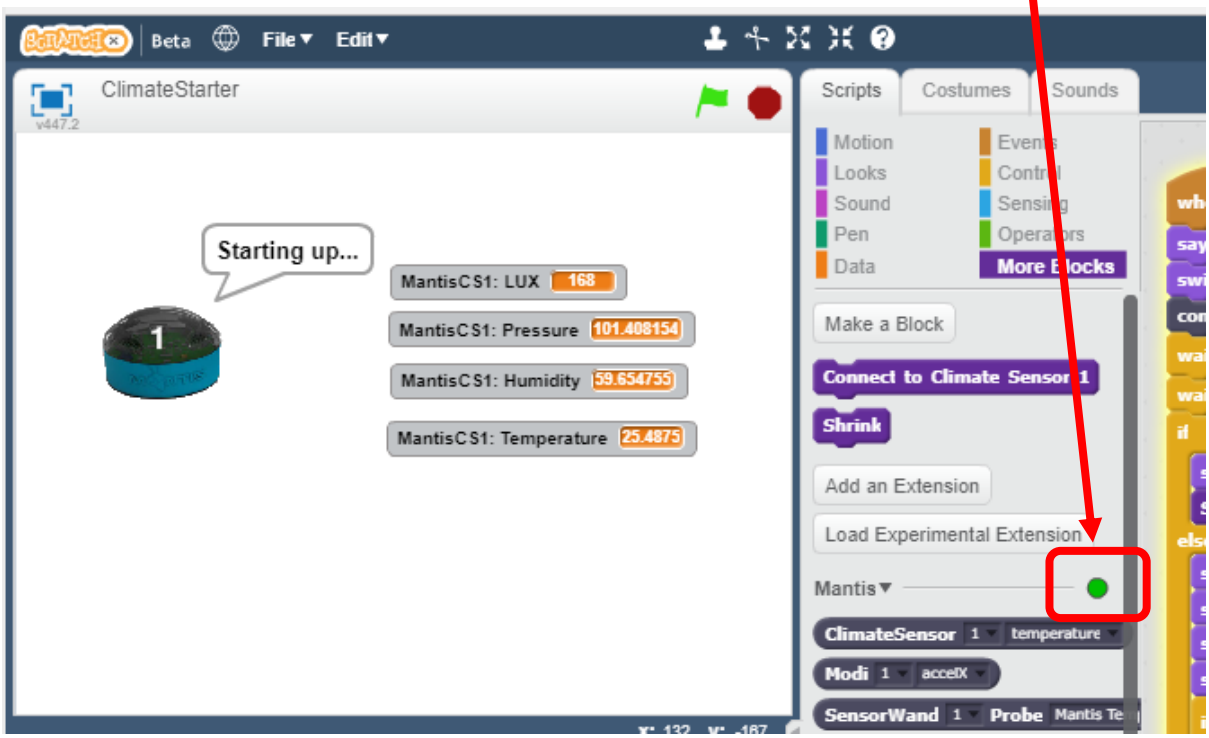
You will now see the Climate Sensor Starter project:



Connecting to the Climate Sensor

Starting the Climate Sensor Starter Project:

Click the green flag to start the project. The Mantis Driver status light should turn green to indicate successful driver status:



The Climate Sensor sprite will guide the user through the connection sequence. The states of the sequence are now given:

Turn On the Climate Sensor

The user is prompted to turn on the Climate Sensor. It is best to start the sequence with the Climate Sensor in the off state to ensure it is not already connected to another project. To turn on the Climate Sensor hold the white button down for approx. 3 seconds until the flashing green light appears.

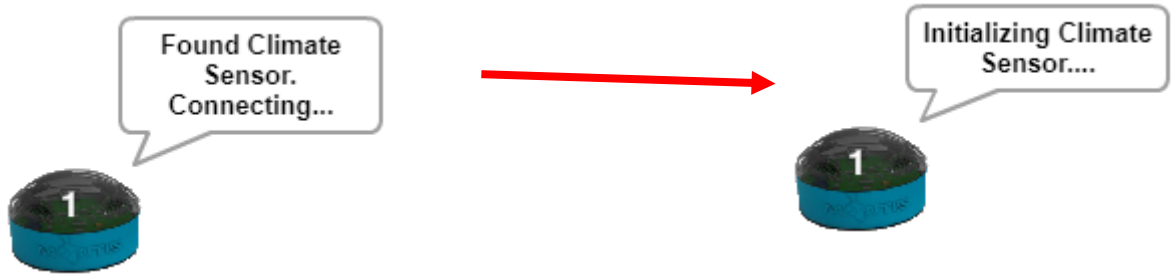
Tap the Button

Around 5 seconds later the user is prompted to tap the button. This puts the Climate Sensor in advertising mode which will enable the CPALMS-Code to pair with it over Bluetooth.

Important: Tap the button immediately! Do not wait **longer than 2 seconds after the message appears**. If this opportunity is missed, just click the green flag and start again.

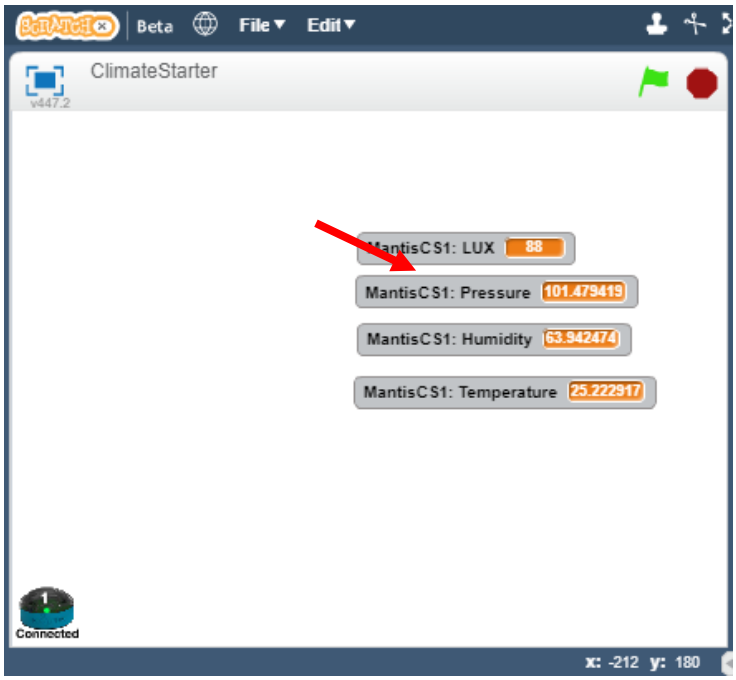
Pairing

If the button was tapped within the 2 second time window, the Scratch project will automatically find and connect to the Climate Sensor:



Connected

The Climate Sensor sprite will shrink, glide to the bottom left corner of the stage and indicate its connected state. Place your hand over the Climate Sensor to block the light and watch the LUX reading change:

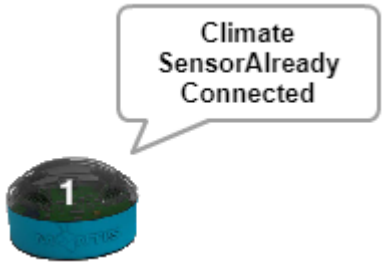


This connection sequence is the same for any Scratch project which imports the Climate Sensor Sprite because Scratch sprites can contain their own blocks of code. We will look into this in more detail later in this document when we build our own project from scratch (no pun intended).

Connecting to an Already Connected Climate Sensor

The code blocks in the Climate Sensor Sprite were built detect when the Climate Sensor is already connected. In this scenario the action is not to initiate a connection sequence but immediately start receiving data instead.

To see this in action click the stop sign on the project which was used to connect. Then click on the green flag:



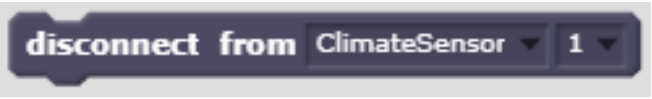
What's happening is even though the Scratch project has been stopped, the Climate Sensor is still connected to the CPALMS-Code App. When the project is started up again, the Climate Sensor sprite resumes receiving data from the Climate Sensor over the same Bluetooth connection.

How Do I Disconnect the Climate Sensor?

To disconnect the Climate Sensor, tap the white button once while it is connected. The Climate Sensor will indicate its disconnected state.

Be Careful!! It is easy to accidently tap the white button while handling the Climate Sensor. This will disconnect your Climate Sensor when you don't want to.

Also, it is possible to run the disconnect block. This is useful to disconnect when you've accidently connected to someone else's Climate Sensor and can't tap the button:



Connect Once and Run Multiple Projects

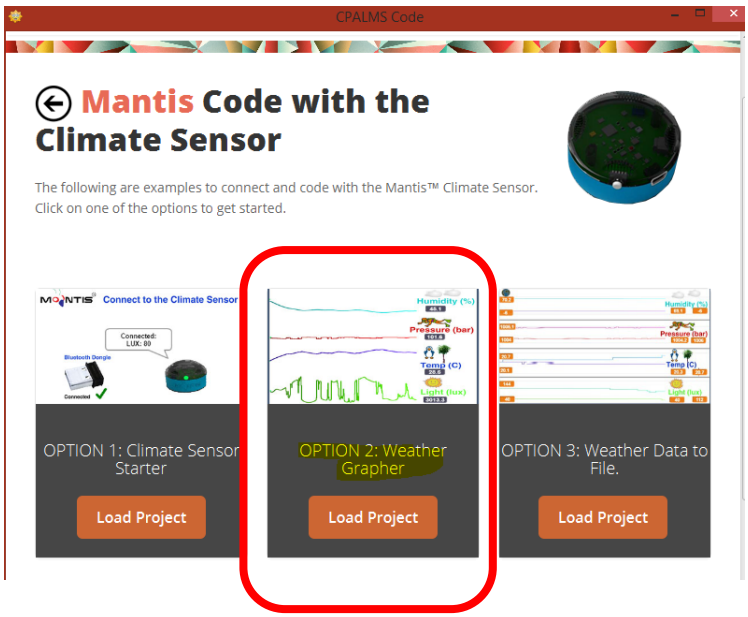
Once the Climate Sensor has been connected in one project, then all subsequent projects can automatically receive data from the same Climate Sensor. Let's do this with the Climate Sensor Starter Project and the Climate Sensor Grapher Project:

Step 1:

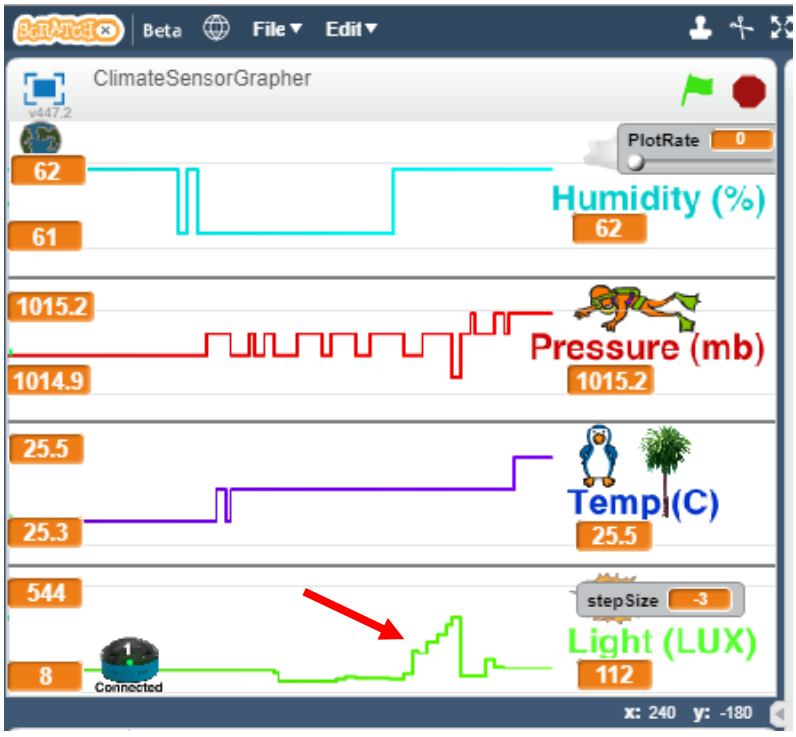
Connect to the Climate Sensor using the Climate Sensor Starter project as described above.

Step 2:

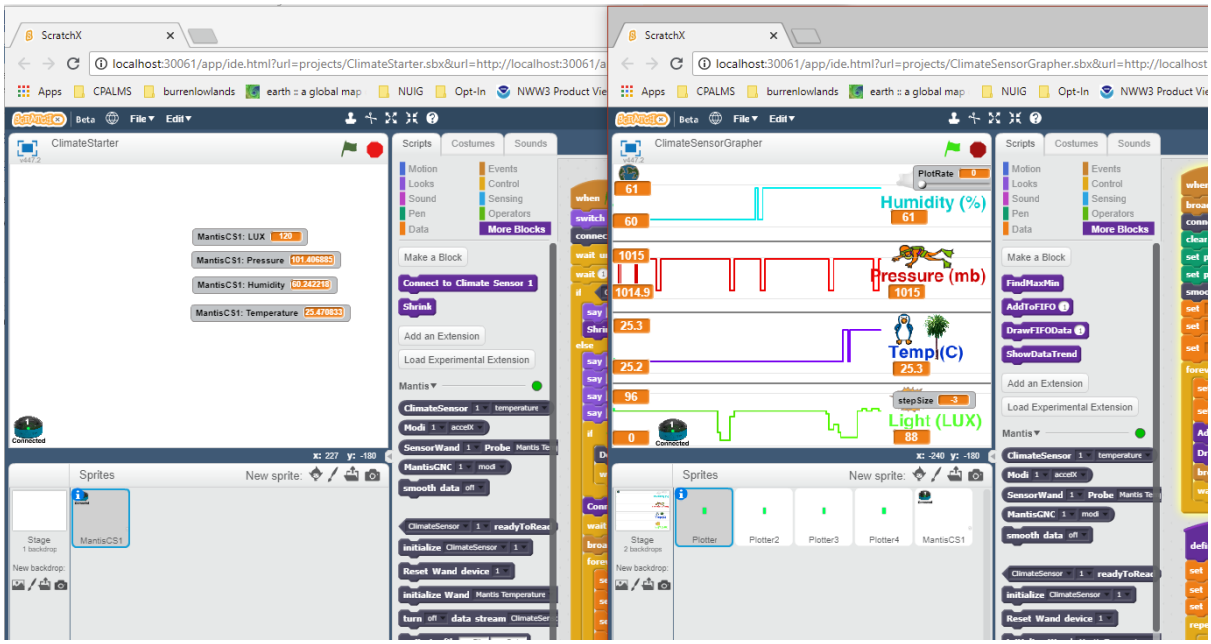
Start the Climate Sensor Grapher project:



Observe the Climate Sensor detect the connection and immediately start receiving and plotting data. Put your hand over the Climate Sensor and watch the light plot react to the change:

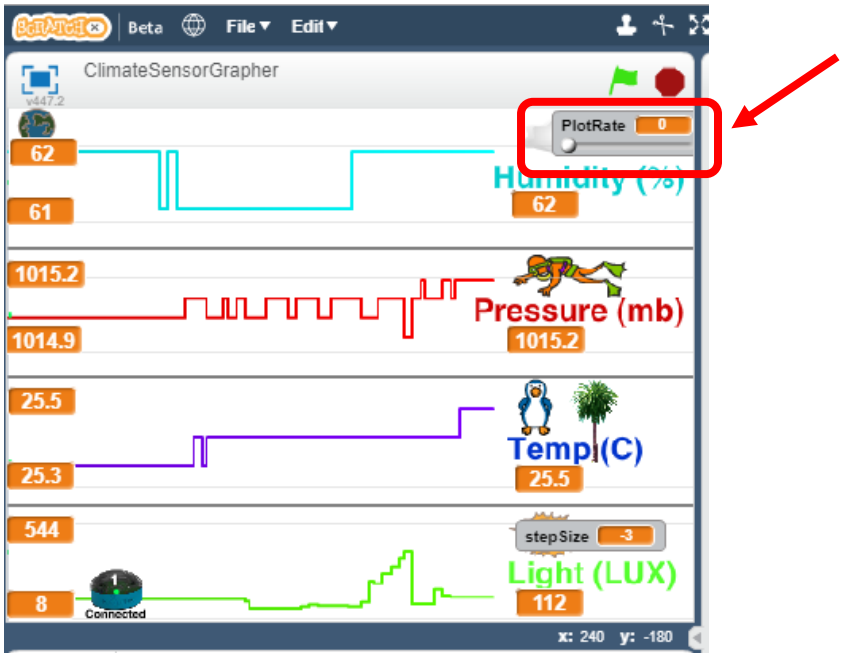


This picture shows the Climate Sensor Starter and Grapher Projects running simultaneously and sharing the same data connection:



Visualizing Data

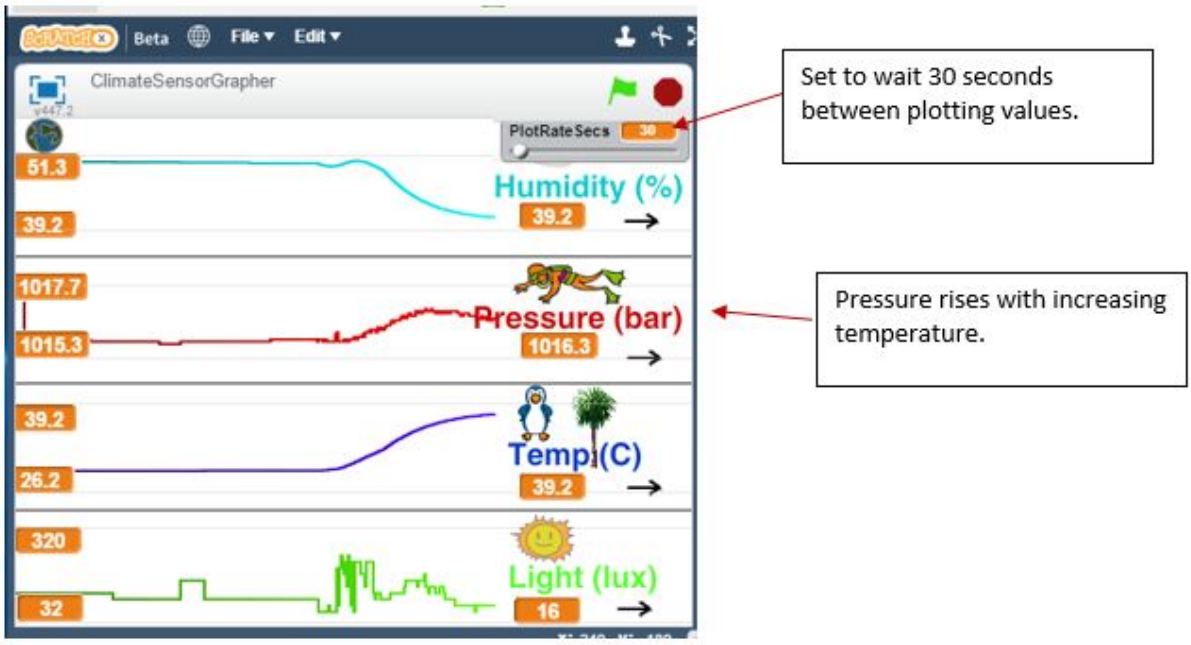
The Climate Sensor Grapher can be used to visualize data for short term laboratory experiments lasting for minutes and long term climate observations which can last for months. The data rate can be adjusted so that the entire observation period can be made to fit on the Scratch stage. The data rate is controlled by the variable “PlotRate” which can be adjusted by a slider:



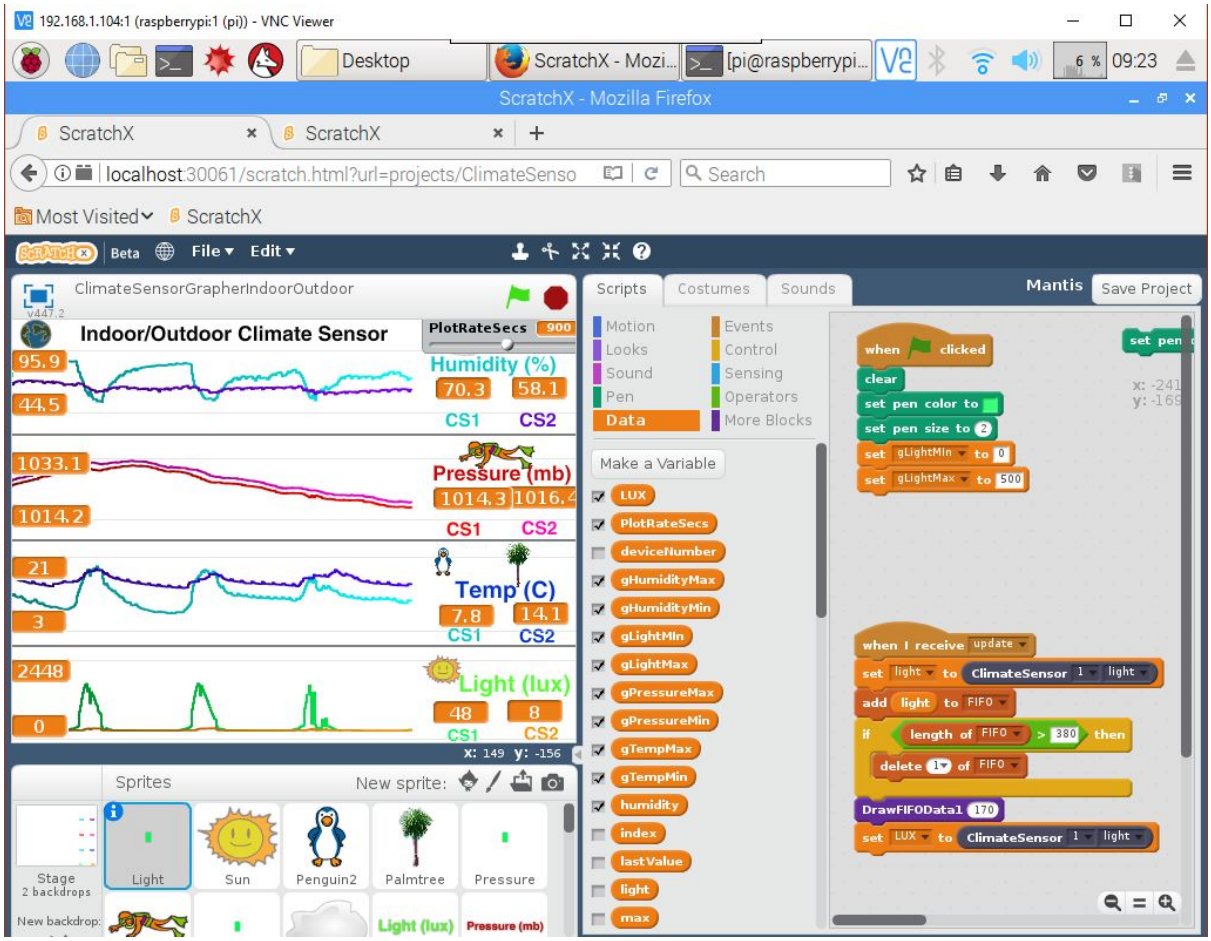
The PlotRate variable represents the number of seconds to wait between plotting a point on the graph. Therefore, when the plot rate value is a:

- **small number** – the graph fills the screen *quickly* as it is waiting *less* time in between plot points
- **large number** – the graph fills the screen *slowly* as it is waiting *more* time in between plot points

For laboratory experiments the plot rate can be set to a small number such as 5 -60 seconds to cover a 20 minute observation period:

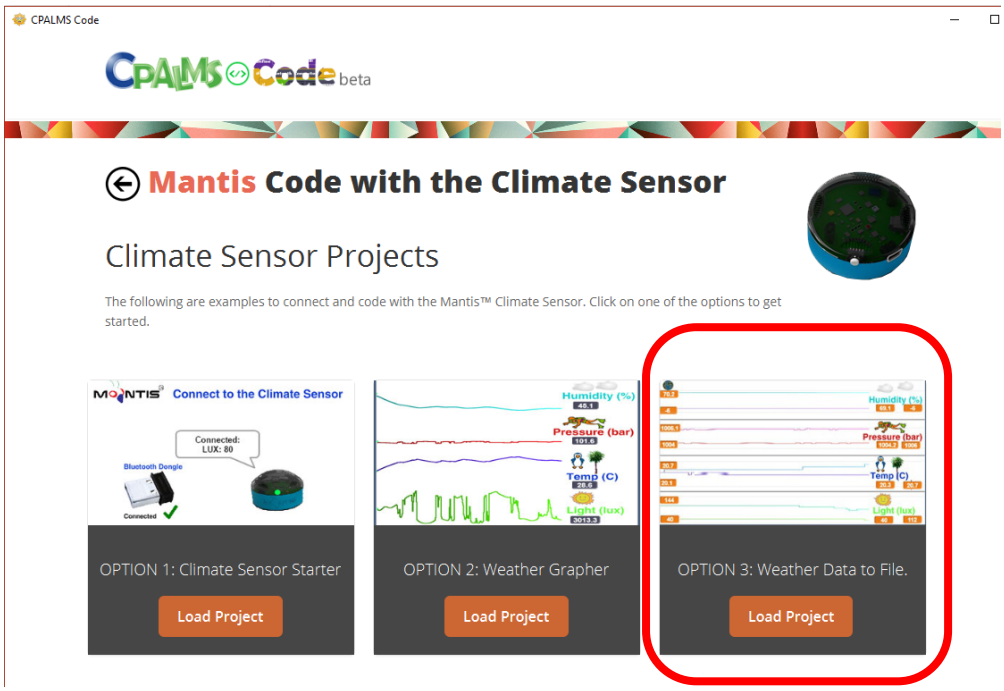


For climate observation the plot rate can be set to a large number such as 900 seconds (15 mins) to allow 4 days of weather data to be displayed on the Scratch stage:



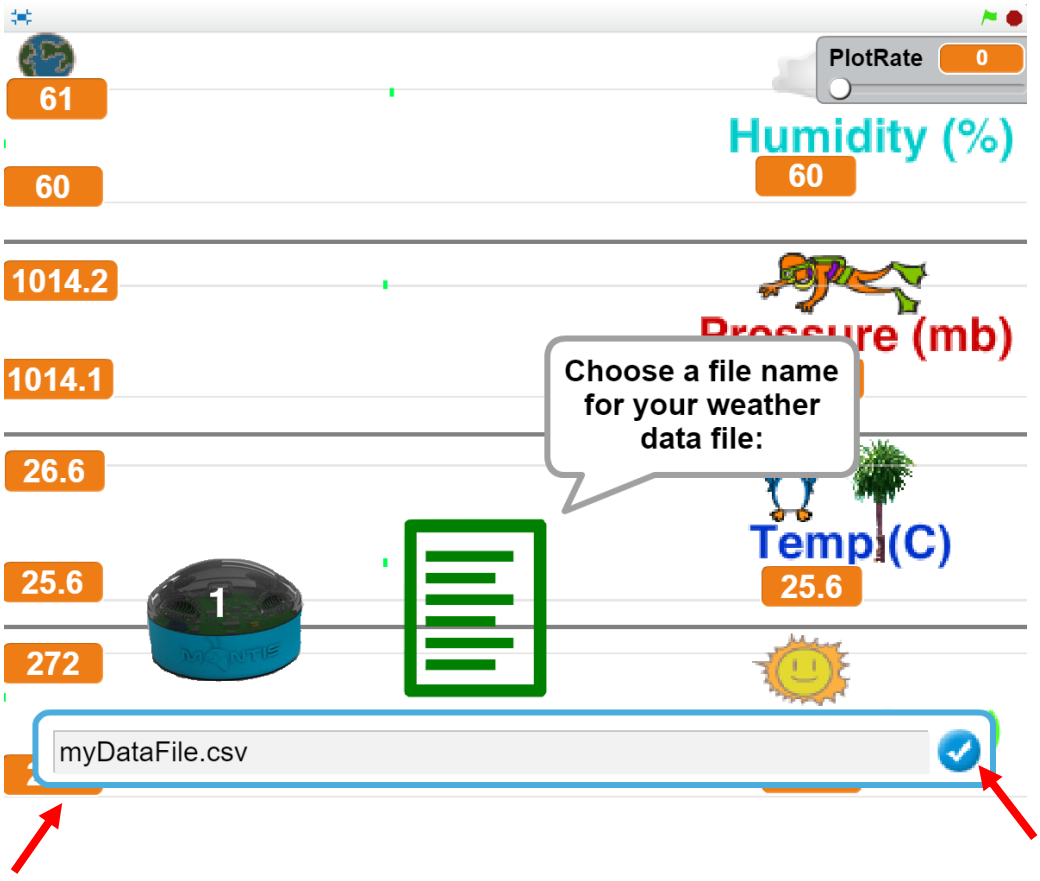
Saving Data

Data can be saved to a file by using the ClimateSensorGrapherDataLogger project which is started by choosing Option 3:



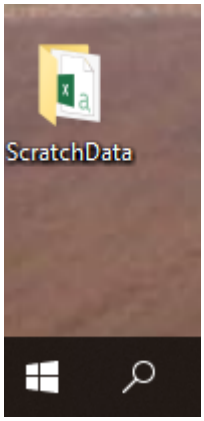
This project is exactly the same as the ClimateSensorGrapher project with the added feature of being able to create a file to which to write data.

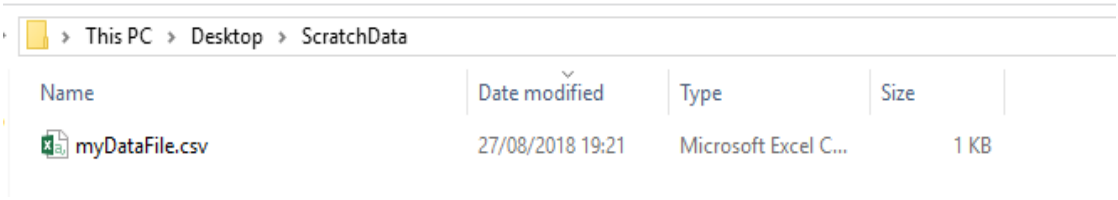
On clicking the green flag the user is asked for a file name:



It is recommended to end the file name with the ".csv" extension because the program creates the data in the (Comma Separated Values) CSV format. By naming with the .csv extension, the computer will know to open the file with a spreadsheet application such as Excel.

You only have to type in the filename. The file will be placed in the ScratchData folder on the desktop. The result from the example above:

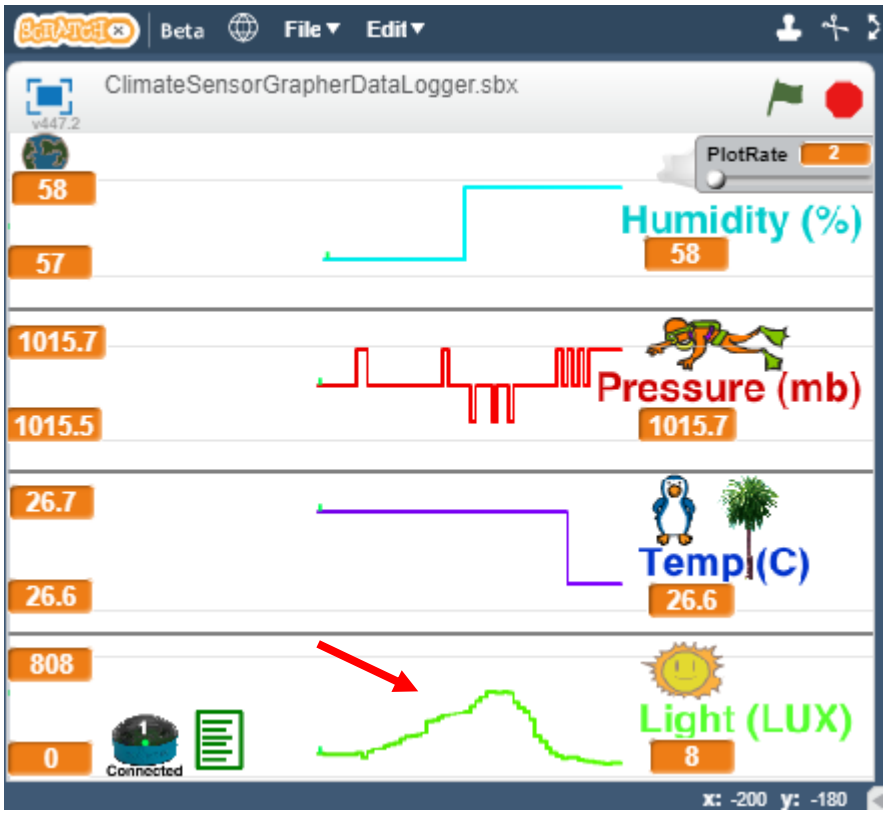




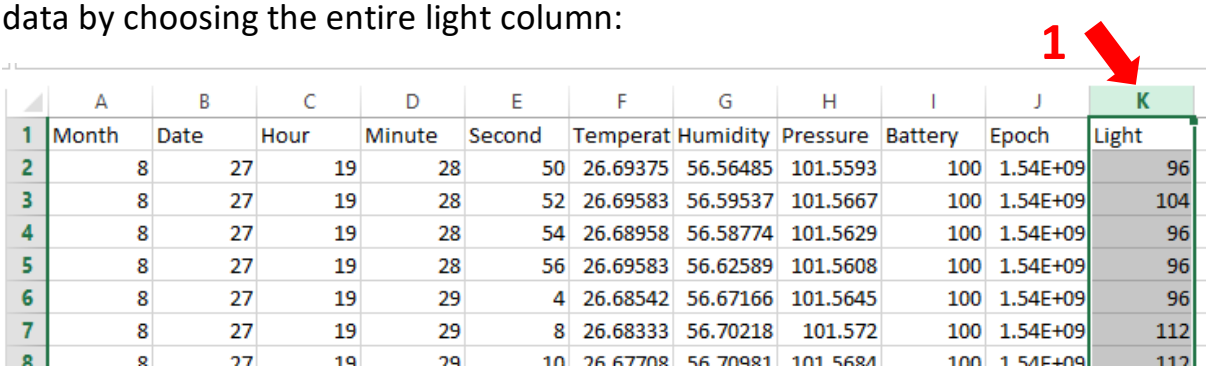
Plotting Saved Data

Let's plot the data which we saved to the file "myDataFile.csv".

We set the plot rate to 2 seconds and moved the Climate Sensor slowly towards more light and then moved it to a dark place more quickly. The light data plot looks like this:



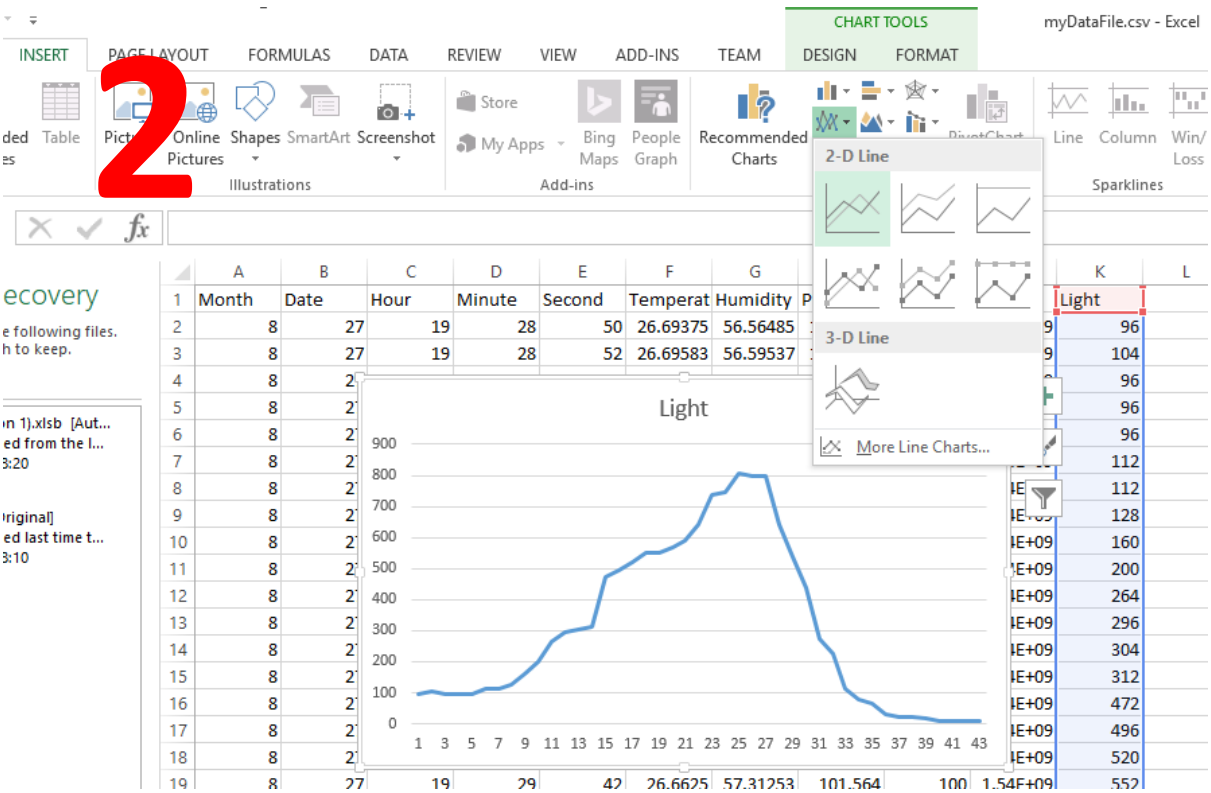
Let's see what the file in "myDataFile.csv" looks like. Open the file and plot the data by choosing the entire light column:



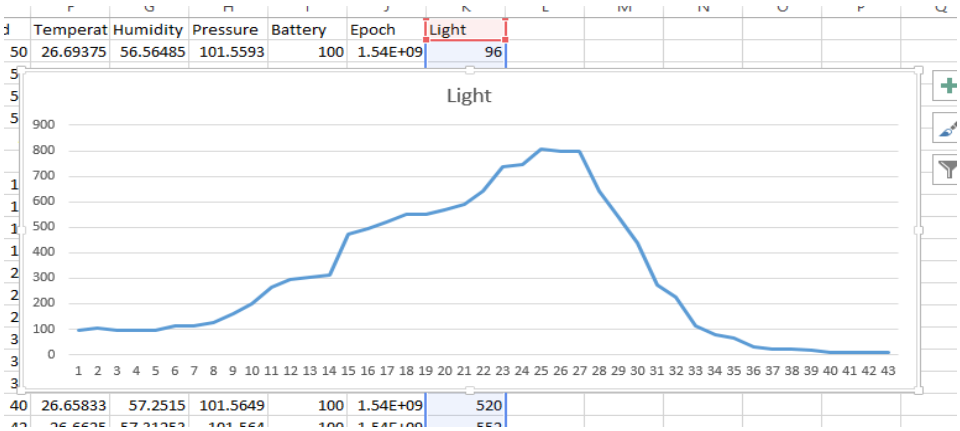
	A	B	C	D	E	F	G	H	I	J	K
1	Month	Date	Hour	Minute	Second	Temperat	Humidity	Pressure	Battery	Epoch	Light
2	8	27	19	28	50	26.69375	56.56485	101.5593	100	1.54E+09	96
3	8	27	19	28	52	26.69583	56.59537	101.5667	100	1.54E+09	104
4	8	27	19	28	54	26.68958	56.58774	101.5629	100	1.54E+09	96
5	8	27	19	28	56	26.69583	56.62589	101.5608	100	1.54E+09	96
6	8	27	19	29	4	26.68542	56.67166	101.5645	100	1.54E+09	96
7	8	27	19	29	8	26.68333	56.70218	101.572	100	1.54E+09	112
8	8	27	19	29	10	26.67708	56.70981	101.5684	100	1.54E+09	112

To make a plot:

1. Choose the entire column of Light data
2. Choose the Insert tab
3. Choose Recommended Charts -> 2-D line chart



The graphs are similar, the horizontal axis is the graph can be stretched to be closer in scale to the graph in the Scratch project:



Create Your Own Project

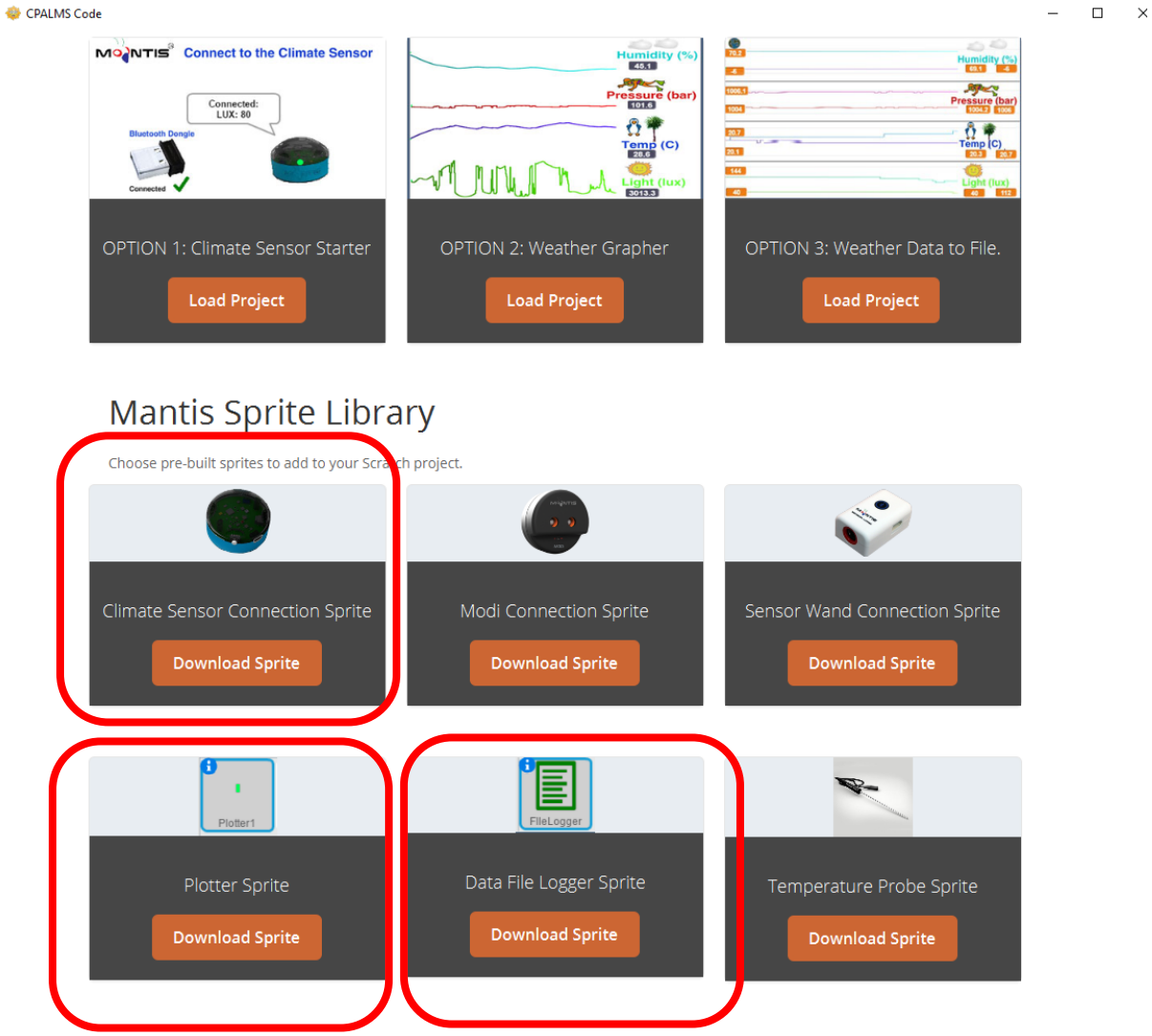
We will build a Climate Sensor project that will:

- Connect to the Climate Sensor
- Plot weather data
- Log the data to a file in .CSV Excel spreadsheet format.

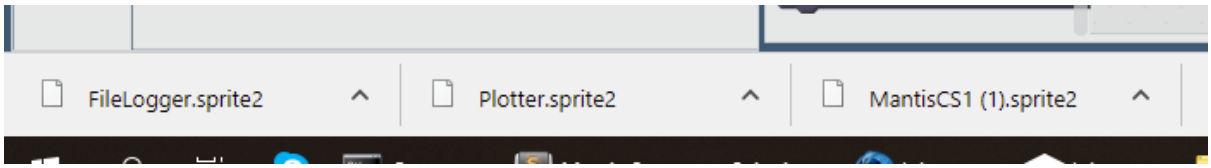
This example will show how to build a new project by importing sprites, thus saving time because each sprite comes with its own code blocks already made.

Scroll down the page so the Mantis Sprite Library panel comes into view.

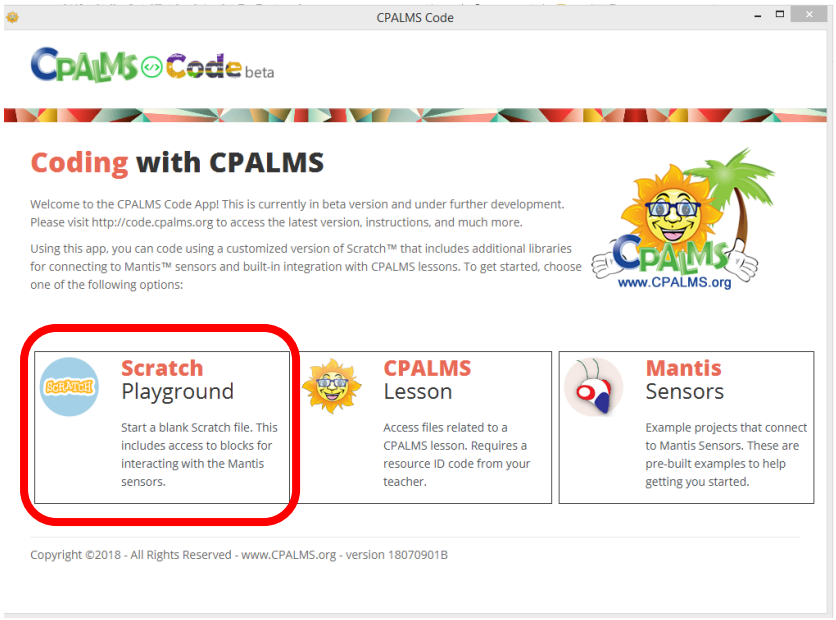
Download sprites for the Climate Sensor Connection Sprite, the Sensor Plotter Sprite and the Sensor File Logger sprite.



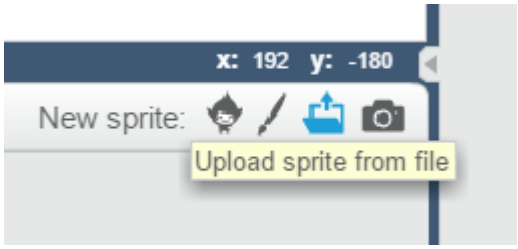
They will end up in your downloads folder:



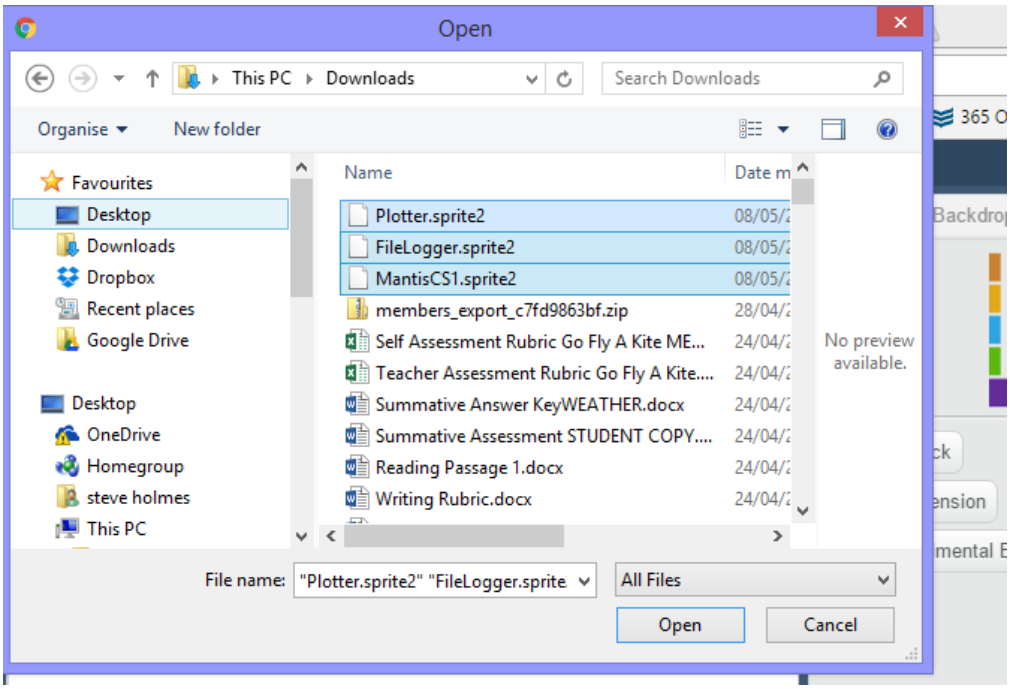
From the CPAMS Code home screen open a new project by clicking on the “Scratch Playground” option:



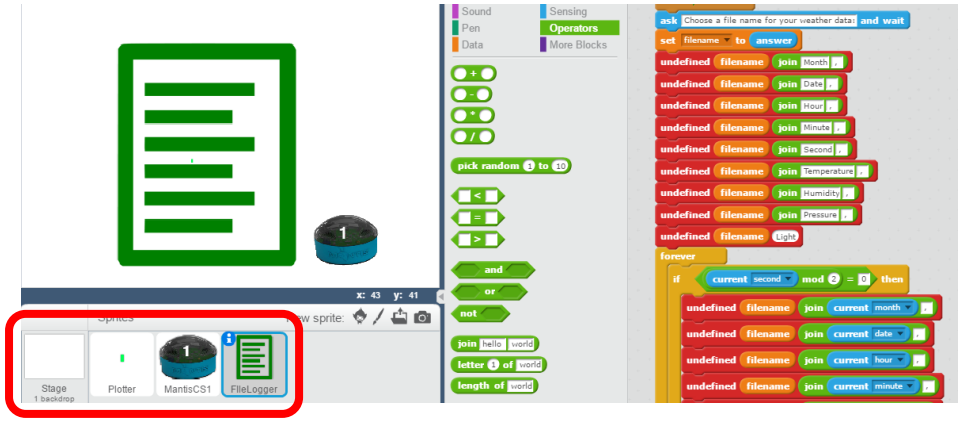
From the new project import these sprites into your new project using the “Upload Sprite from File” button:



Select the sprite files from where you downloaded them:



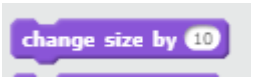
Congratulations! You saved yourself a lot of coding:



You can hide or shrink the sprites on the stage as you see fit.

Shrinking a sprite:

Go to the looks palette and select the size you would like the sprite to be:



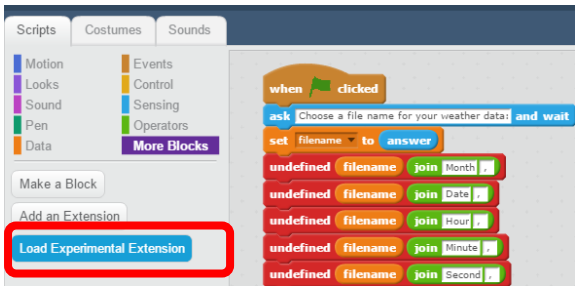
Hiding a sprite:

Click on the blue information icon in the top left corner of each sprite and select hide or use a Scratch block: 

But wait a minute! What are those red blocks? They look like errors. We'll deal with them in the next section.

Loading the Extension

Now we have to load the extension so that the red “undefined” blocks become recognized. Click the “Load Experimental Extensions” button and load the pre-configured extension:



All should be good now. The red blocks are now replaced with the blocks from the Scratch extension. Ignore the “Extension Problem” message when loading the extension. There is no problem.

Customizing the Pre-Built Sprites

Now that we have all of our sprites in the Scratch project, we need to make minor modification so they all work together properly.

Note: All pre-built sprites contain block code which works correctly. Only minor modifications which are described here are necessary to make project specific configurations.

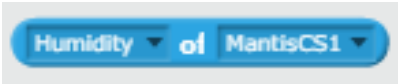
If you accidentally make a mistake in modifying the code which causes the sprite to not work properly you can simply upload the sprite again and start over.

Climate Sensor Variables:

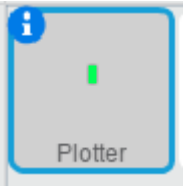
Note the Climate Sensor Sprite (and the other sprites) automatically creates its own variables within the project when it is uploaded:

Other sprites can access the Climate Sensor 1 variables like so:

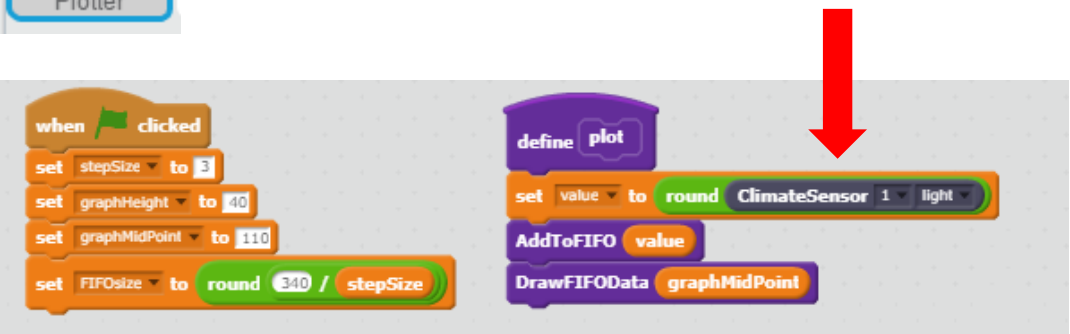
or



Using the Plotter Sprite



Put in the sensor block you wish to plot. Climate Sensor 1 light is a good block to plot in real time because you can wave your hand over the light sensor and see the changes immediately:



You only have to specify the Climate Sensor block in one place in the Plotter sprite – yay!

Plotter Variables

The position of the plot on the stage, height, width, and update frequency can be determined by setting the following variables:

The variable 

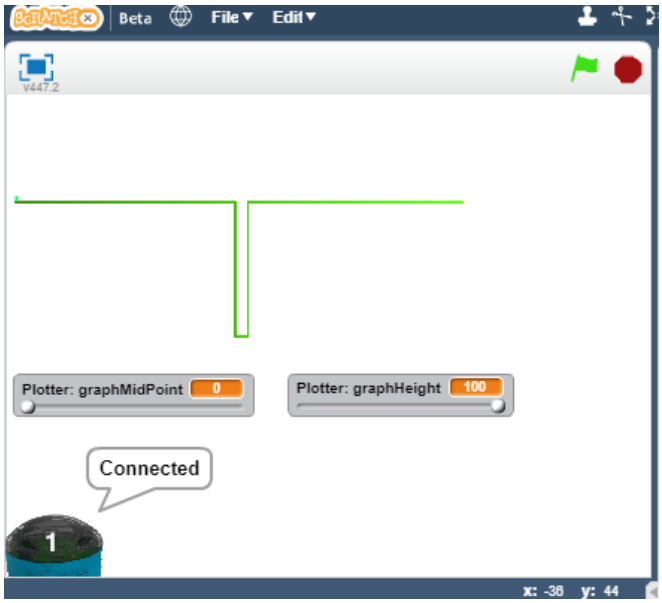
determines the y position of the graph i.e whether it sits high or low in the stage. A larger number will cause the graph to go closer to the top of the stage.

The variable 

determines the range height of the graph. When set to 100, the full range of the graph occupies 100 pixels. A smaller value will render the range of the graph in a narrower band.

and 

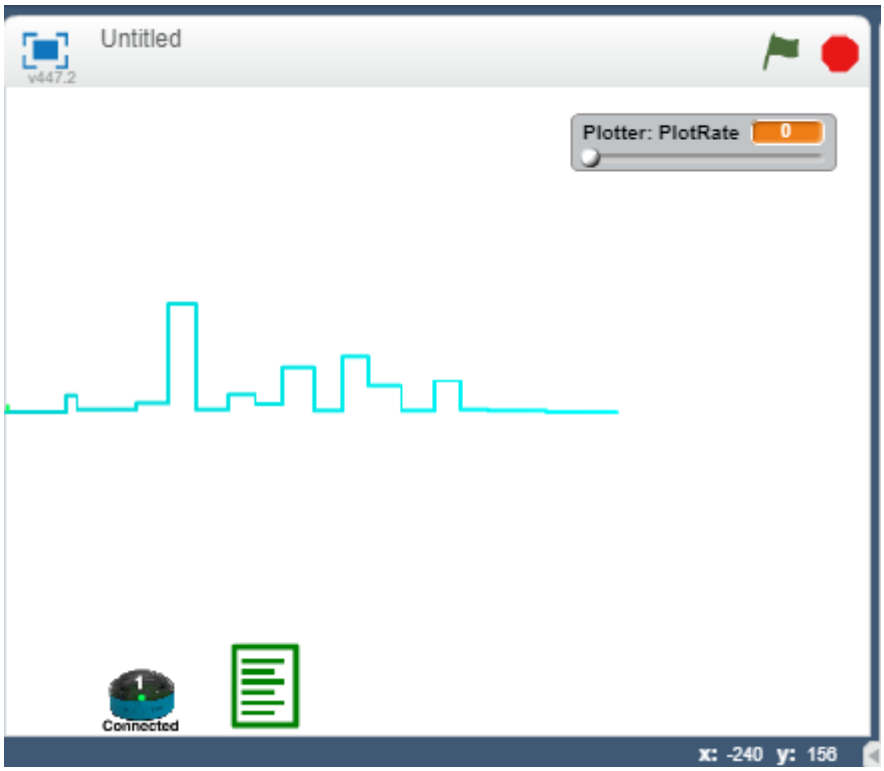
default values are set to 0 and 100 respectively to get the following plot position. Note the use of sliders for the variables which allows the exact positioning to be seen instantly for any given value.



The variable FIFOsize determines the width of the graph. The value determines the number of stored values which will be plotted. The default value is 340 / stepsize. This will cover the entire stage and leave a margin on the right.

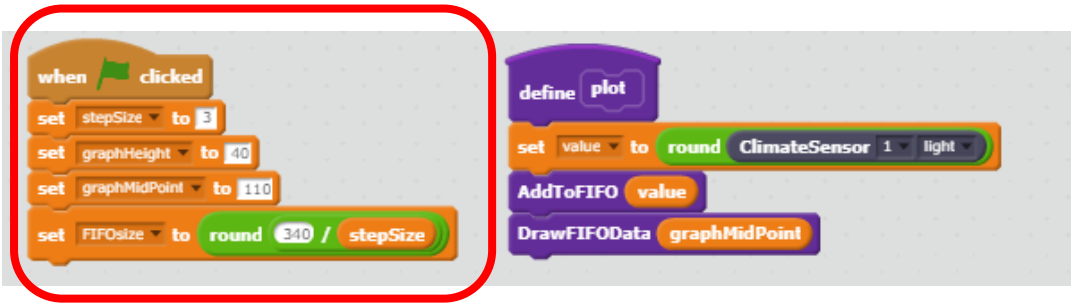
A smaller value holds less values, therefore, is narrower. Here we see the value of 100 creating a narrow plot:

The variable PlotRate determines the update frequency of the graph. Its value determines the amount of time in seconds which to wait between plotting each point. A value of 0 means the plotter does not wait and continuously streams a plot of current data. PlotRate can be used as a slider so the data rate can be adjusted easily:



Setting the Plotter Variables:

The variables described above can be set in the Plotter Sprite Script in the highlighted blocks:



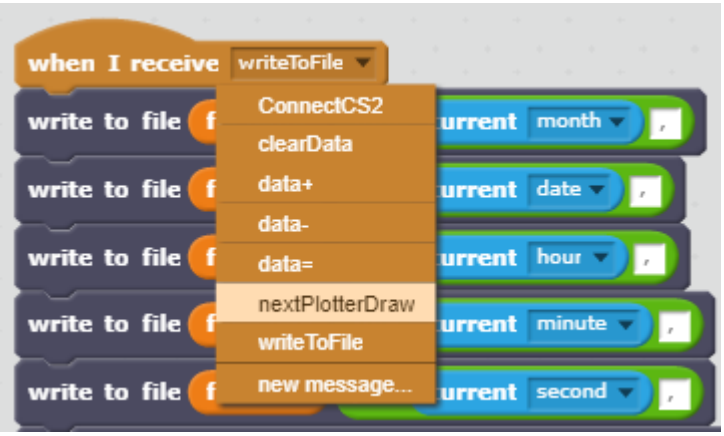
Using the File Logger Sprite

The File Logger needs to receive a broadcast message in order to write to the file. Its default state is to write to a file when the “writeToFile” broadcast message is received.

Synchronizing the Plotter and File Logger

However, there is no “writeToFile” broadcast message in this project. But the plotter sprite regularly broadcasts the “nextPlotterDraw” message each time it plots point.

If the “writeToFile” message is changed to receive the “nextPlotterDraw” message, then the File Logger will write data to the file every time the plotter draws a point on the Scratch stage:



Saving Data Autonomously

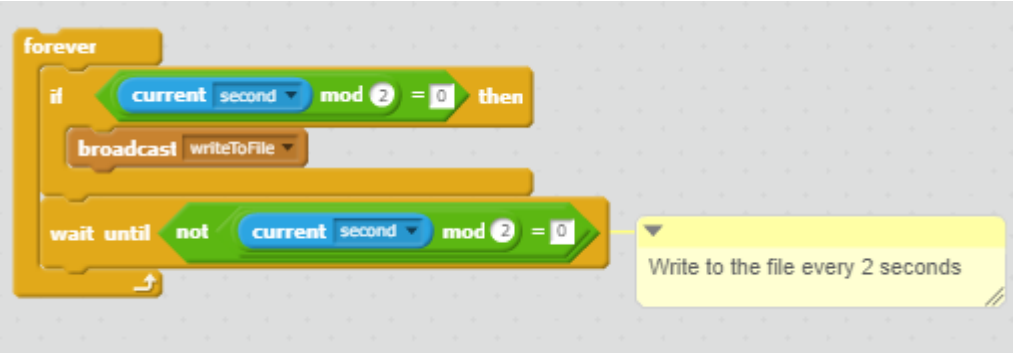
If the project did not contain a Plotter sprite, then a simple solution is to create a forever loop which broadcasts the “writeToFile” message at the desired regular interval:

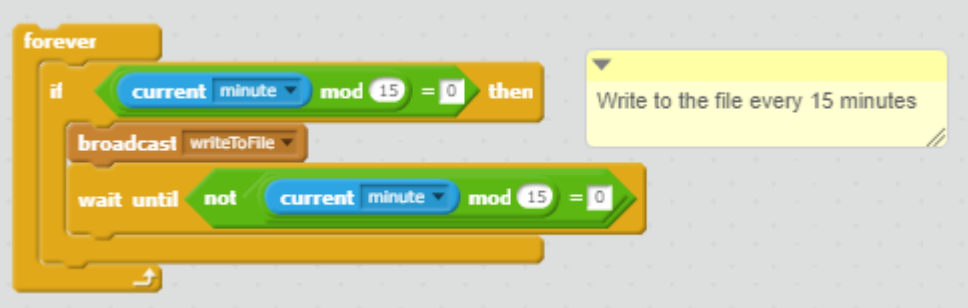
Difference between “write to file” and “write line to file”

The “write to file” block writes to the file *without* adding a new line. The “write line to file” block writes to the file then it adds a new line so that the next write to the file happens on the new line.

Using the Computer as a Timer

It is possible to use the clock on your computer as a timer. Here are some possibilities:

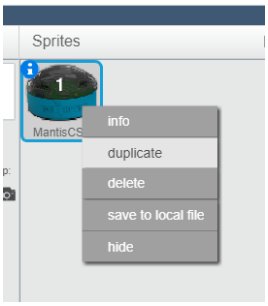




Using More than One Climate Sensor

The Climate Sensor Sprite is set to connect to Climate Sensor 1 by default. If you are only using one Climate Sensor then you don't have to do anything.

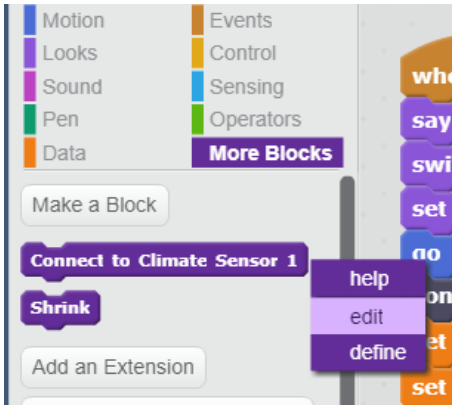
If you want to use a second Climate Sensor in your project the easiest thing to do is put Climate Sensor 2 into its own independent sprite by duplicating the Climate Sensor 1 sprite, then follow the steps below to modify the code:

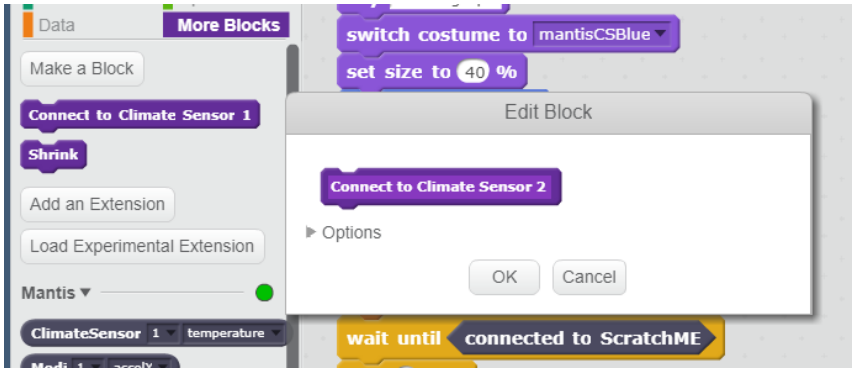


In the duplicated sprite go into the costumes tab and change the Climate Sensor number from 1 to 2.

Double click on the number and you'll be able to type the new number into it:

Go the blocks palette and change the name of the “Connect to Climate Sensor 1” block to “Connect to Climate Sensor 2”:

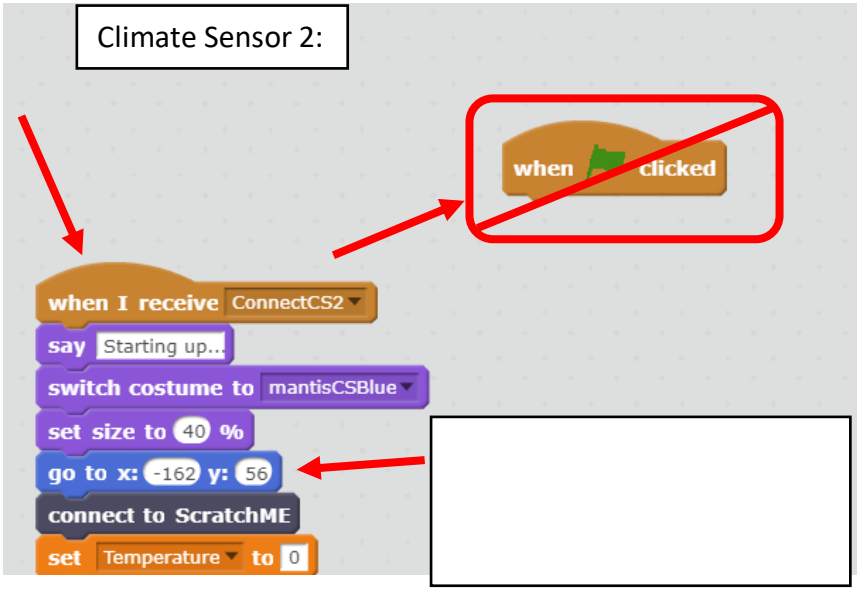




Notice the name of all occurrences of this block will change as a result of this action.

Important: The Climate Sensor 2 connection sequence must happen *after* Climate Sensor 1 has finished connecting.

Therefore, replace the green flag with the broadcast message “ConnectCS2“:



Notice Climate Sensor 1 broadcasts “ConnectCS2” after it has connected successfully. Since we duplicated Climate Sensor 1, we must remove the “ConnectCS2” broadcast in our new Climate Sensor 2 Sprite:

Note: If we had a third Climate Sensor, we could create a new broadcast message called “ConnectCS3” and duplicate the Climate Sensor 2 sprite to make it the Climate Sensor 3 sprite in the same manner.

In the duplicated Climate Sensor 2 sprite modify all occurrences of the Climate Sensor 1 block in the sprite's scripts to Climate Sensor 2 by changing the number in the drop down menu: