

# Watching Ice Melt (Electronically!)

---

## *A real scientific experiment with Scratch and a Picoboard*

By Myra VanInwegen

Scratch, a programming language and environment from MIT's Lifelong Kindergarten, is a fun way to start to learn computer programming. A Picoboard (pictured right) allows you to get input from the real world into your Scratch programs via various sensors.

In this report, I describe how to use the Picoboard and a thermistor, along with some Scratch programs I have written, to investigate the phase change of ice into water. This project requires some familiarity with Scratch, although you do not have to be an expert. The target audience is a teacher in secondary school, although it would also work fine for a parent to do some science/computing investigations at home.

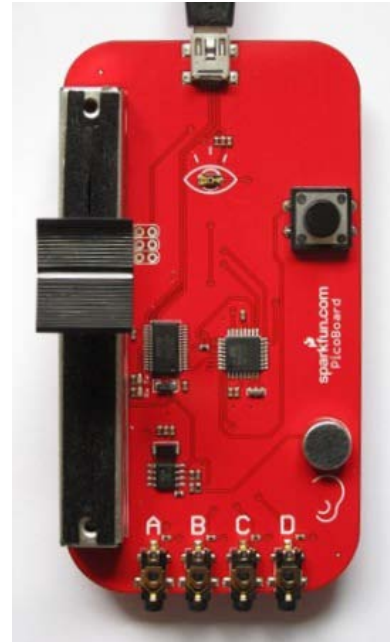
### **Project summary**

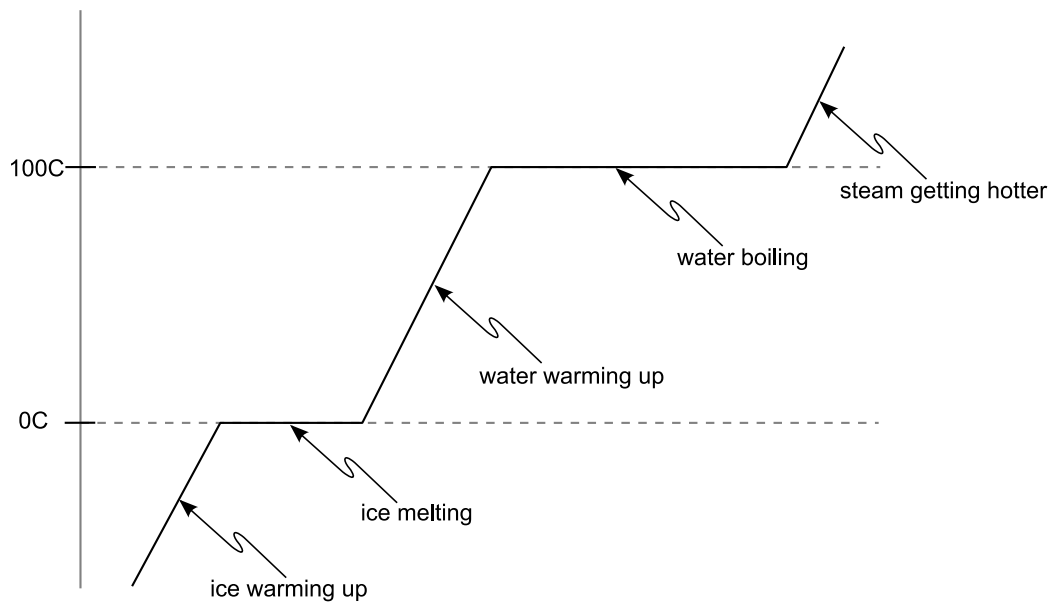
This project builds on my *Fun with Data* project. That project describes how to use the `data_collector.sb` and `data_plotter.sb` Scratch programs (available from the same place you got this document) to collect a set of data points from one of the Picoboard sensors and then to plot and do some basic investigations on the data. Here I describe how to attach a thermistor to an analogue port of the Picoboard to study the behaviour of ice as it melts. At the end, I suggest some other experiments you might do with thermistors.

It may seem strange that Scratch, a programming language usually used in primary school and the first few years of secondary school, is used for this project, when phase change does not seem to be studied in any depth before GCSEs (when students are 14 or more years old). It happened this way because when I got the Picoboard the most interesting thing I thought of to do with it was to try to replicate the first part of the phase change graph temperature graph for ice. Hence this project!

### **Phase change**

Phase change, which is studied in chemistry and physics, happens when materials change state, where the basic states of matter are solid, liquid, and gas. So melting, boiling, freezing, condensing, and sublimation are all examples of a change in state. The famous graph that describes the phase changes in water is below. It shows the temperature that results from taking a block of very cold (well below freezing) ice and applying a constant amount of heat (energy) to it.

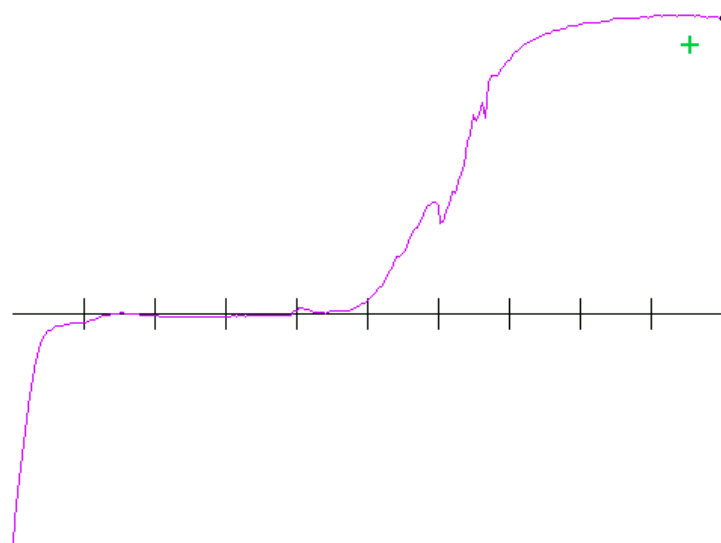




At first the ice warms, but when it hits the melting point (0C), the temperature remains constant. The heat that is being applied to the ice does not warm it up; instead the energy is used to melt the ice. Once the ice is all melted, then the temperature of the water begins to increase until it hits the boiling point. As it boils or evaporates, the temperature remains at 100C: the energy being applied is used to turn the water into water vapour/steam. After this, as heat is still fed into the system, the temperature rises again.

It occurred to me to try to replicate the first part of this graph, following the transition from ice to water, by using a temperature sensor attached to an analogue input of the Picoboard. Reading up on the Picoboard, I found that the analogue inputs are detecting resistance, so what I needed to attach was a thermistor, which is basically a resistor where the resistance varies with the temperature. My idea was to freeze the thermistor into the middle of a block of ice, then pull the ice block out of the freezer and monitor the readings from the thermistor using the Picoboard as the ice slowly melted. I used my `data_collector.sb` Scratch program to collect the data and `data_plotter.sb` to draw the graph.

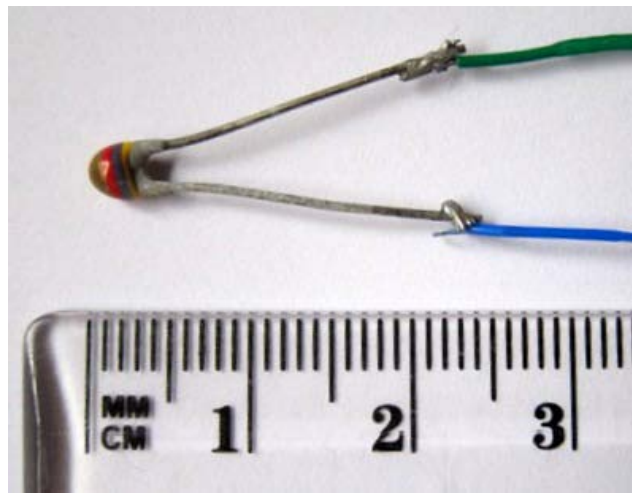
I had hoped that my experimental data would show the plateau in temperature as the ice melts. It does: one of the data sets that I recorded is plotted to the right. (This plot is a created by the `data_plotter.sb` Scratch program.) The first part of the graph is a steep rise as the ice starts to warm up. Then the temperature levels off while the ice melts. Once the ice is all melted, the temperature starts to rise again, but instead of continuing to increase steadily, as



in the diagram above, my graph indicates that the temperature rise slows down, until the temperature levels off at room temperature. The reason for this difference is because in my experiment I am not putting heat into my ice/water at a constant rate: it is just sitting in a normal-temperature room. The amount of heat (energy) flowing in to the ice/water is proportional to the difference in temperature between the water and the room. When the ice has just melted and the water is at 0C, the difference in temperature is over 20C, so heat flows into the water at a high rate, warming the water fairly quickly. As the water warms up, the difference in temperature between it and the room is less, so less heat flows into it, so its temperature rises more slowly. Beautiful, isn't it?

## Do it yourself

Here is how you can replicate my melting ice experiment. First buy a thermistor (and a Picoboard if you don't already have one). Thermistors are very cheap. I got one from Maplin for about £1. Most thermistors available nowadays are very small, with short 25mm leads. This isn't so convenient for freezing into a block of ice, because the thermistor would have to be quite near the surface of the ice block in order to have the ends of the leads poking out of the ice. I solved this problem by soldering some wires onto my thermistor, but an easier solution is to buy a thermistor with long leads. (At time of writing I found thermistors with long leads on sale from an ebay.co.uk seller southern\_temperature\_sensors\_ltd, and from <http://uk.mouser.com>.)



Thermistors are rated by their resistance at room temperature. The one I bought from Maplin was rated at 4.7K $\Omega$ , and that gave me a good range of values when I did my experiment. I think that any value around 5K $\Omega$  to 10K $\Omega$  would work well. (Note that  $\Omega$ , the Greek letter omega, is used for Ohms, a measure of resistance. K $\Omega$  means kilo-Ohms. A lower number means that it offers less resistance to electricity.) Thermistors with room temperature values even well out of this range might work well enough, since the data returned by the analogue inputs on the Picoboard are floating point numbers (which means that they have a decimal point in them and can represent fractional values; they are not just whole numbers). Thus even if the difference between the high and low values is not great, the floating point results mean that you should get enough detail in your measurements.

Put your thermistor into the middle of a pot of water and put the pot in a good freezer. You want to get the ice block well below 0C so you can record the ice heating up to 0C before the temperature levels off as the ice melts. A good way to get the thermistor to stay in place as the water freezes is to put a stick across the top of the pot of water and twist the leads around the stick. By the way, you don't need to worry about the low temperature damaging your thermistor: all the thermistors I looked at have an operating temperature range of -40C to +125C.

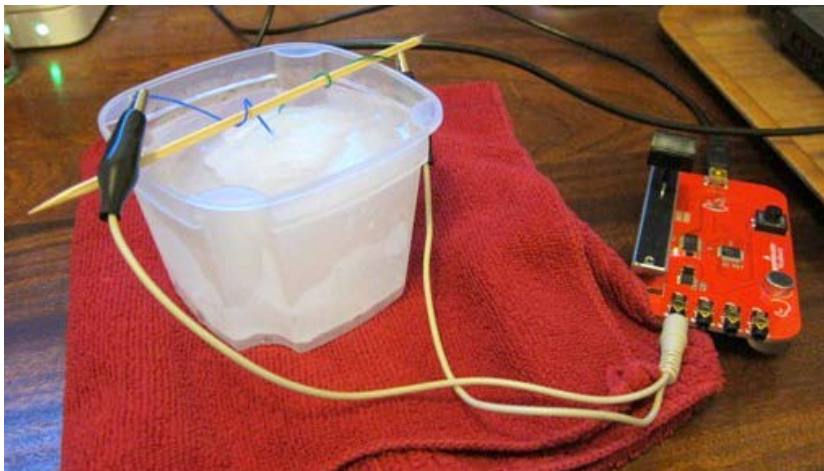
Once the block is frozen solid, you can start your experiment. Before taking the ice out of the freezer, start up Scratch and then load up the data\_collector.sb program. Make sure the Picoboard is

working (see the tips in my *Fun with Data* document). Before collecting data you must change the program to choose the sensor to read from.

Near the bottom of the cat's script is a block saying "add <light> sensor value to <data>".

(Here I am using <> to indicate a part of a block that can be changed, with a sample value

between the <>.) We're not measuring light in this experiment; instead we are measuring the resistance of the thermistor using one of the analogue inputs. Say you're going to attach the thermistor to analogue input A. Use the menu in the block to select the resistance-A input, so it reads: "add <resistance-A> sensor value to <data>".



Now you are ready to go. Take the ice block out of the freezer. Grab one of the Picoboard crocodile clip attachments, put one clip on each lead of the thermistor in your block of ice, and plug the other end into the Picoboard. Then click the green flag in the Scratch window. The cat will ask you how long you want the test

to last (in hours and minutes) and also what time it is. The time information will allow you to determine at what time of day some interesting event in the data occurred. For this experiment the time of day doesn't really matter, so you can just enter 0. When I did my experiment I had about 350ml of water in my pot, and it took about 9 hours for all the ice to melt and the resulting water to come up to room temperature. Thus 9 or 10 hours would be a good amount of time for your test if you have a similar amount of water.

The program will then take readings at equal intervals for the duration of the test. After all the data is collected, the cat will invite you to save your data to a file by right-clicking on the "data" list box in the display area.

## Display the data

Use the program data\_plotter.sb to draw a graph of the data on in the Scratch display area. To load the data you collected and saved earlier, right-click on the data box in the display area.

There are some ways to configure the plotting; see the code (or the *Fun with Data* project) for more information. The one that is most relevant for this project is the "flip\_y" variable. This variable is set in the first block in the cat's main script. If it is set to true, the graph will be drawn upside down, so a larger data value will be shown lower on the screen. I added this option because all the thermistors I have come across have a negative temperature coefficient, which means that when the temperature increases, the resistance goes down. If you simply plot the value read from the thermistor, you will have your graph going down as the temperature increases, which is very



counter-intuitive. If you have a thermistor with a negative temperature coefficient and “flip\_y” is set to true, a lower value read from the thermistor (which corresponds to a higher temperature) is shown higher on the graph, so it corresponds to what you expect.

To plot your data, click the green flag.

### More fun with thermistors

Thermistors are fun and cheap, so why buy just one? Buy several and hook them up to your Picoboard at the same time. The `data_collector.sb` program at the moment only records one lot of data at a time, but as it's a Scratch program, this is easy to change. Make some more list variables, and find where the data value from the sensor is added to the list. Alter it so that several code blocks are used to add the data from the other sensors you want to record to the other lists. After the data is collected, right click on each of them to save them out.

Then you can plot them all together using the `data_plotter.sb` program. To make sure that the graphs are scaled the same along the y axis, you'll want to manually set the “max” and “min” variables to the maximum and minimum values of all the data sets, as described in *Fun with Data*.

Here are a few ideas of things you can do with two or more thermistors.

1. Experiment with the transfer of heat across a surface by partially filling a large non-metallic bowl (like a Pyrex mixing bowl) with hot water and carefully lowering into this a can filled with cold water. Dangle a thermistor into each body of water and record the data as the temperature of the water in the can rises and the temperature of the water in the bowl falls.
2. See how the colour of an item affects how it absorbs the sun's energy by taking two thermistors and sandwiching each between bits of cardboard. Paint the top of one sandwich white and the other black, and monitor the temperature of the thermistors. Does the black one get hotter than the white one?
3. See how a greenhouse works by attaching two thermistors to your Picoboard, putting one in a glass baking dish (with a lid) and leaving one in the open air. Put them both in the sunshine and monitor the temperature for awhile. Does the one in the glass dish get hotter than the one in the open air?

These are just some ideas. I'm sure you will come up with some of your own.