Newton's Law of Cooling: a piece of science behind C.S.I.

(exp. id 20210329-1-v1)

An experiment proposed by
Beatriz Padín – Santiago de Compostela (Spain)

Overview

"The time of death was between 11:00 pm and 2:00 am". We've heard this statement many times in films and TV series in Crime Scene Investigation (C.S.I.). What is the science behind it? One of the forensic ways to get a rough estimate of the time elapsed between the death of the victim and their remains being found.
In this activity you will study experimentally the cooling process of some objects (not bodies...).
There is some simple theory for how the cooling will happen, following a very simple idea: Newton's Law of Cooling. According to this law, the rate at which a hot body loses heat is directly proportional to the difference between its temperature and the temperature of its surroundings, as

$$\frac{dT}{dt} = -k\left(T(t) - T_{env}\right) \,, \tag{1}$$

where t is the time, $T(t)$ the temperature of the object at time t, T_{env}, the temperature of the surrounding environment, assumed to be constant, and k, is the "cooling constant", which depends on the surface area of the body, its heat capacity, the roughness and other characteristics of its surface, etc. This parameter also changes depending on whether the experiment is carried out under natural or forced convection.
This law can be solved analytically, leading to this equation:

$$T(t) = T_{env} + \left(T_0 - T_{env}\right)\exp\left(-kt\right). \tag{2}$$

Here, T_0 is the temperature of the object at time $t = 0$ s. This simple law describes how the rate of the cooling process decreases as the body cools down, and shows that the temperature of the body exponentially approaches the temperature of its surroundings.
The possible topics to study are:

- how does the rate of heat transfer change with time?
- What is the value of the cooling constant?
- What factors affect the cooling constant?

Materials

To analyse the cooling process, you must measure time and temperature. This can be accomplished using a thermometer and a stopwatch. In this activity, we use a digital temperature sensor to make the measurements.

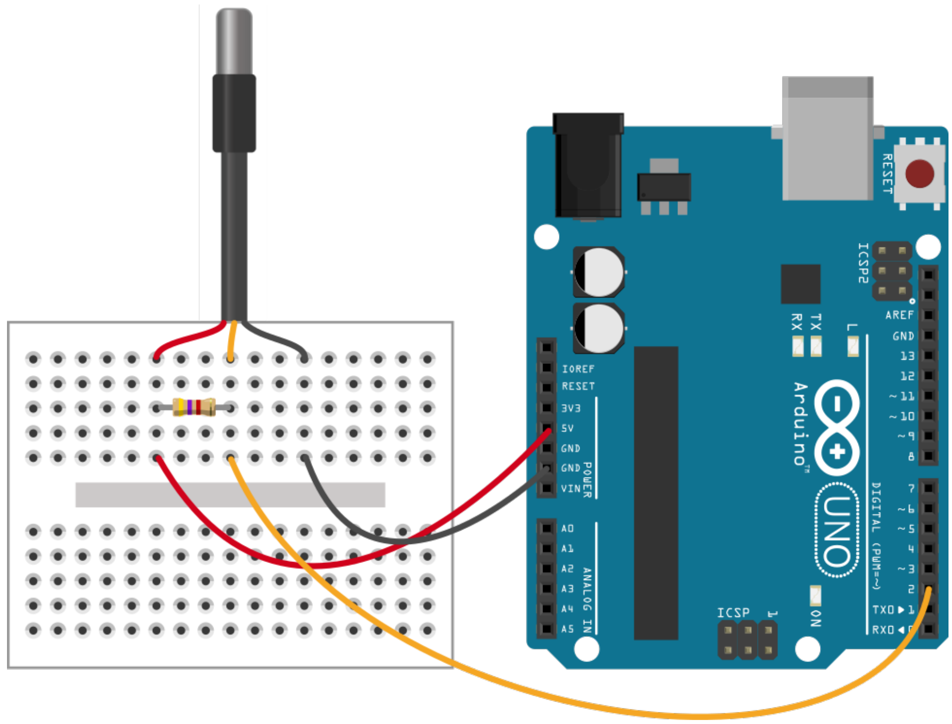

Figure 1: How to connect the sensor to Arduino.

1. Arduino UNO
2. A computer with the Arduino IDE installed
3. DS18B20 Digital Temperature Sensor (waterproof)
4. 4.7 kΩ resistor
5. Breadboard
6. Jumper cables

To read the values measured by the sensor we use an Arduino microcontroller and the Arduino programming language. To keep this document reasonably short, we assume that you are already familiar with Arduino [1].

So, from now on, we assume that you have installed the Arduino IDE in your computer, and that you already know how to write a sketch, how to add a library to the Arduino IDE, how to upload a sketch to the board, and how to open the serial monitor.

Connecting and programming the sensor

Connect the DS18B20 sensor to Arduino. According to its documentation, the sensor must be powered with a $5V$ DC source via the black and red cables. The third cable is used to communicate with Arduino and must be connected to a digital pin (e.g., pin 2). A 4.7 kΩ resistor must be put across the signal pin and the $5V$ pin (see Fig. 1) [2].

The following sketch reads the temperature every 30 seconds and prints the values to the serial monitor of Arduino. It requires the inclusion of the `DallaTemperature` library.

```
#include <OneWire.h>
#include <DallasTemperature.h>

// The sensor is connected to Arduino's digital pin 2
```

```
const int ONE_WIRE_BUS = 2;

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensor(&oneWire);

// The measurements are taken every 30 s
long previousTime = 0;
long interval = 30000;

void setup() {
  Serial.begin(9600);
  Serial.println("t [s],T [degrees]");

  sensor.begin();

  // Initial measurement (t=0 s)
  float temperature = sensor.getTempCByIndex(0);
  Serial.print(millis()/1000);
  Serial.print(", ");
  Serial.println(temperature);
}

void loop() {
  unsigned long currentTime = millis();

  if (currentTime - previousTime > interval) {
    previousTime = currentTime;

    sensor.requestTemperatures();
    float temperature = sensor.getTempCByIndex(0);

    Serial.print(currentTime/1000);
    Serial.print(", ");
    Serial.println(temperature);
  }
}
```

Experimental procedure

The first step is to record the ambient temperature, as it will be used in the data analysis. Although it may change slightly, we assume that it is constant throughout the experiment.

To make the measurements, put some hot water inside a beaker and introduce the waterproof temperature sensor in the water. Run the Arduino program to measure the temperature of the water every thirty seconds. When you have finished taking the measurements, save the collected data into a CSV file or into a spreadsheet (you can, e.g., simply copy and paste them from the serial monitor), according to your preferred analysis method.

To study the factors affecting the value of the cooling constant k, the experiment can be repeated under different conditions:

- changing the surface area of the water being cooled;
- trying with containers of different materials or painted in different colours;
- having cooling take place under forced convection.

Data Analysis

Plot the temperature–time graph for the cooling water, and, from this graph, try to extract several conclusions, at least qualitatively.

- Look at the slope of the temperature–time graph and check for the validity of the cooling law: initially heat is transferred at a high rate – the graph has a steeper slope –, while, as thermal equilibrium is approached, the heat transfer rate decreases and the slope becomes more horizontal.
- What happens to k when the surface area S is increased? How does k depend on S?
- As you know from everyday experience, cooling takes place faster under forced convection. How does this reflect on the value of k?

To make a quantitative measurement of the cooling constant k, first linearise the function, writing its logarithm:

$$\ln(T - T_{env}) = \ln(T_0 - T_{env}) - kt\,. \qquad (3)$$

The graph of $\ln(T - T_{env})$ vs. time looks as a straight line, whose slope is $-k$:

To determine the line of best fit (the straight line that best approximates the given data), use dedicated spreadsheet functions or some software for data analysis, such as R or PYTHON.

Extension

In general, the cooling process involves various heat transfer phenomena: conduction, convection and radiation. The relationship stated above between temperature and time in a cooling process was first established by Newton in 1701. Very soon after Newton, other scientists found discrepancies from his cooling law, but the deviations from the exponential law were attributed to experimental errors. According to U. Besson [3], “The discreet charm of the proportionality and the confidence in the simplicity of natural laws seem to have strongly influenced Newton and will influence many later scientists”. Taking this approach into account, reflect about the following aspects:

- The relationship between experimental data and mathematical models, and the analysis of the possible discrepancies between them.
- The range of experimental values where an empirical law is valid, and the need to change the model when the range of values is enlarged.
- A critical reflection on the history of science through historical case studies.

References

[1] If this is your first time with this microcontroller, you will find all the necessary information on how to start using it in the Instructable “Beginner’s Guide to Arduino” (`https://www.instructables.com/A-Beginners-Guide-to-Arduino`).

[2] For the instructions on how to connect and program the DS18B20 sensor, you can refer to “Low-cost Sensors in the Physics Classroom (by bpadin). Step 4: DS18B20 Digital Temperature Sensor” (`https://www.instructables.com/Low-cost-Sensors-in-the-Physics-Classroom/#step4`).

[3] U. Besson, The History of the Cooling Law: When the Search for Simplicity can be an Obstacle: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.468.4908&rep=rep1&type=pdf

For the instructor

(exp. id 20210329-1-v1)

1. Newton's cooling law is an approximate law. For the data to fit the exponential law, the starting temperature of water should be about $30°$C above the room temperature, and the beaker with hot water should be placed on an insulating material to maximise cooling by convection with respect to other mechanisms.
2. Although the cooling law can be easily studied with a thermometer and a stopwatch in a "technology-free environment", the use of Arduino has great benefits. Students practice additional skills that are essential in scientific work, like connecting and programming sensors. Moreover, it makes Physics classes more engaging and motivating, and it encourages students to be creators, and not just consumers of technology.
3. If students are already familiar with Arduino, you can encourage them to make more complex programs or to add additional components like an LCD screen to read the measured values and an external power bank to make it portable. Or they can make a data logger, saving the measured values in an SD card.

The DS18B20 digital temperature sensor has been successfully used in different experiments with 13-15 years-old students without any previous experience in using Arduino.

Objectives, level, and duration

1. Primary objective: engaging students in inquiry based science learning.
2. Primary objective: learn about heat transfer.
3. Secondary objective: learn the basis of digital sensors with Arduino.
4. Suggested for: high schools and university.
5. Duration: three hours.

Further Info Online

Please leave feedback, suggestions, comments, and report on your use of this resource, on the channel that corresponds to this experiment on the Slack workspace "smartphysicslab.slack.com". Instructors should register on the platform using the form on `smartphysicslab.org` to obtain login invitation to the Slack workspace, and/or to request being added to the mailing list of smartphysicslab.