**Arduino:**

**Radioactivity Half-Life Simulator**

**Teachers Guide**

**Richard Born, Associate Professor Emeritus**

**Northern Illinois University**

**Operations Management and Information Systems**

***Introduction***

The study of radioactive decay and half-life is an essential part of any course involving atomic and nuclear physics. The most common way to study this in the physics laboratory is by the use of an isogenerator that produces a small sample of radioactive barium, whose half-life is about 2.6 minutes. The cost of a barium isogenerator is typically around $250.

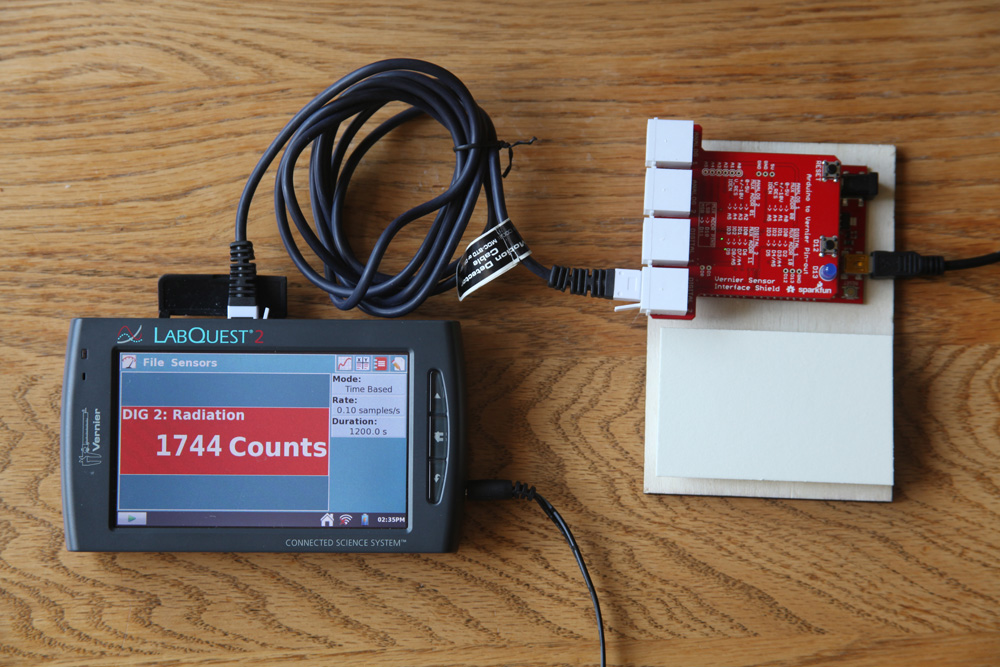
This simulated lab uses an Arduino™ radioactivity half-life simulator that can be used to study principles of radioactive decay at a significantly reduced cost. The Arduino program (they call them "sketches" when you are working with Arduino) that produces the simulation can be run on the SparkFun® RedBoard (ARD-RED) sold by Vernier, to which is attached a Vernier Arduino Interface Shield (BT-ARD). An MDC-BTD cable (with BTD connector on both ends) is connected to the Vernier Arduino Interface Shield and to a digital port on a data-collection device such as LabQuest 2. The Arduino requires electrical power. There are two ways to supply this power:

- Use a standard Arduino power supply, which is 9 to 12V DC, 250 mA or more, 2.1 mm plug, center pin positive. There is a round socket for the power supply on the Arduino.

- Connect a USB cable from the Arduino to a USB charger (of the type used for cell phone or a computer). That is the situation shown in Figure 1 below.

The LabQuest App is then used to select the radiation monitor as the sensor. The Arduino sketch pulls the Arduino digital line 6 high and then low on each simulated atomic decay, thus registering a count on LabQuest 2. In addition, the LED (D13) on the interface shield flashes on each of the simulated atomic decays. In a nutshell, this setup functions like a real radiation meter that is monitoring a radioactive material. *Note that the simulated sensor radiation source will not auto ID, so students need to set up the Radiation Monitor in Sensor Setup before starting the lab.*

See Figure 1 for a picture of the apparatus setup.



*Figure 1*

Even though the student handout (and these Teacher Notes) assume that the student data is taken with a LabQuest 2 directly (not connected to a computer), with just small variations to the lab, it can also be done using any of the following:

- Logger *Pro* and a computer connected to a LabPro, original LabQuest, or LabQuest 2

- an original LabQuest

- TI calculator and LabPro or CBL 2 (using the DataRad program)

- TI Nspire™ and lab cradle

***The Four Simulated Isotopes***

One advantage of this simulated version of the lab is that it can simulate different radioactive isotopes. The table of Figure 2 indicates typical values for the decay constant *λ* and half-life *t½* for each of the four isotopes simulated with the Arduino sketch. Note that *t½* = 0.693/*λ*. It is advisable that students collect data for a time period of at least two to three half-lives. Each run of the sketch randomly selects one of the four isotopes. The Arduino sketch also provides random initial decay rates of approximately 110, 230, and 320 decays per minute.

|  |  |  |
| --- | --- | --- |
| **Isotope ID Number** | **Decay Constant, *λ* (seconds-1)** | **Half-Life, *t½*, (seconds)** |
| 0 | 0.01250 | 55 |
| 1 | 0.00625 | 111 |
| 2 | 0.00357 | 194 |
| 3 | 0.00227 | 305 |

*Figure 2*

Figure 3 shows partial data collection for isotope 3 and an initial decay rate of approximately 230 decays per minute. Again, both the isotope and initial decay rate were randomly selected by the Arduino sketch.

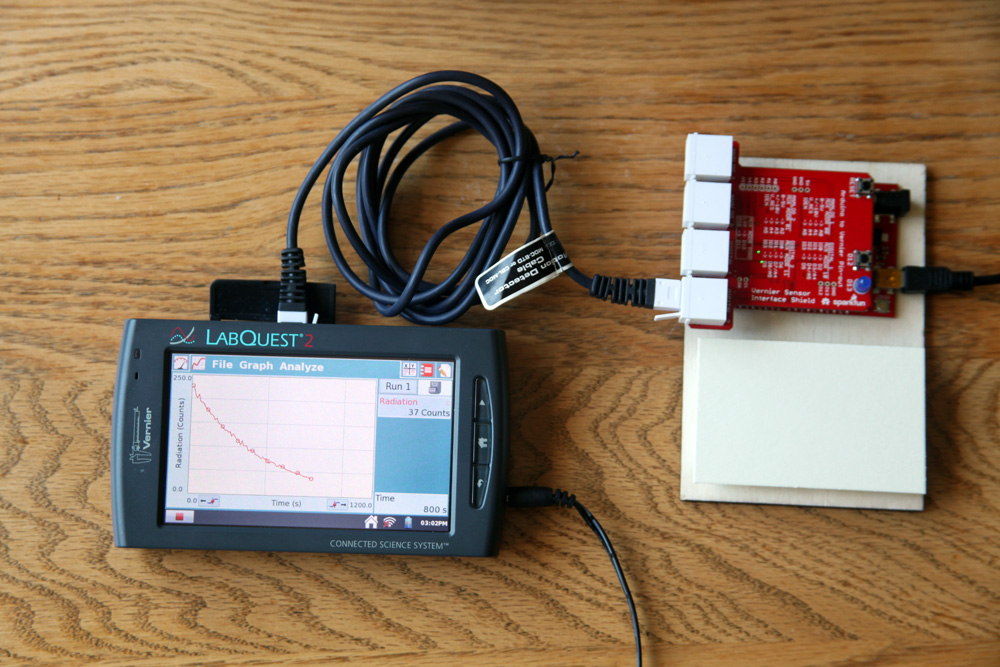


Figure 3

***Running the Radioactivity Half-Life Simulator***

The student version of this lab is just a very slight modification of Experiment 3 in the *Nuclear Radiation with Vernier* lab book. That lab assumed students would use a source called an isogenerator to produce a sample of radioactive barium. The isogenerator contains cesium-137, which decays to barium-137. The newly made barium nucleus is initially in a long-lived excited state, which eventually decays by emitting a gamma photon. The barium nucleus is then stable and does not emit further radiation. Using a chemical separation process, the isogenerator allows the students to remove a sample of barium from the cesium-barium mixture. Some of the barium is left in the excited state and will subsequently decay. This adds to the background radiation. Students are asked to study the decay of the excited barium and measure the the activity and lifetime of the excited barium.

The hardware and software you need to simulate the radiation source and counter can be prepared as follows:

1. You need a computer with the Arduino Integrated Development Environment (IDE) installed on it. This is free and can be downloaded from www.arduino.org
2. A SparkFun RedBoard (Vernier order code ARD-RED) or any similar version of an Arduino must be connected to your computer with the provided USB cable.
3. Start the Arduino IDE software. Choose *Tools>Board>Arduino Uno*. Then choose *Tools>Serial Port* and select the port that the RedBoard will use for communication.
4. Upload the sketch file *BornVernierHalfLifeSimulator.ino*. It is part of the collection of Arduino sketches at https://github.com/VernierSoftwareTechnology/arduino
5. Connect a cable with BTD connector on both ends (Vernier order code MDC-BTD) to the Digital 2 connector on the Vernier Arduino Interface Shield. This cable will connect to LabQuest 2 or other lab interface. Note: This cable is the same as the cable used with any Vernier Motion Detector. If you have Vernier Motion Detectors, just borrow one of those cables.
6. When the students go to do the lab, the Arduino radiation simulator will not auto-ID, so they will need to tell the data-collection software that they have a Radiation Monitor connected to the lab interface. The figure below shows this being done on LabQuest 2.



Figure 4

1. As written, the student lab has data collected for 30 minutes. This should be long enough for any of the four simulated isotopes. Note that students can always terminate data collection sooner if an isotope with a shorter half-life is randomly selected by the sketch.
2. To restart the sketch, students press the Reset button on the Vernier Arduino Interface Shield. LED D13 will blink twice brightly, indicating the sketch has restarted. Then after a second or two, LED D13 will begin blinking randomly, with each blink representing a simulated radioactive atom decaying.

***Data Analysis***

The Arduino sketch that simulates radioactive decay assumes that background count is negligible, so there is no need to make adjustments for this.

1. From the LabQuest App, choose *Analyze/Curve Fit*, and check the checkbox for *Radiation.*
2. Since the graph shows a negative exponential shape, for the *Fit Equation* choose *Natural Exponent*. The LabQuest screen will then appear similar to that of Figure 5. This kind of an analysis is standard when considering radioactive decay that is characterized by the equation *R(t) = Roe-λt*, where *R(t)* is the decay rate at time *t*, *Ro* is the decay rate at time zero, and *λ* is the decay constant for the isotope.

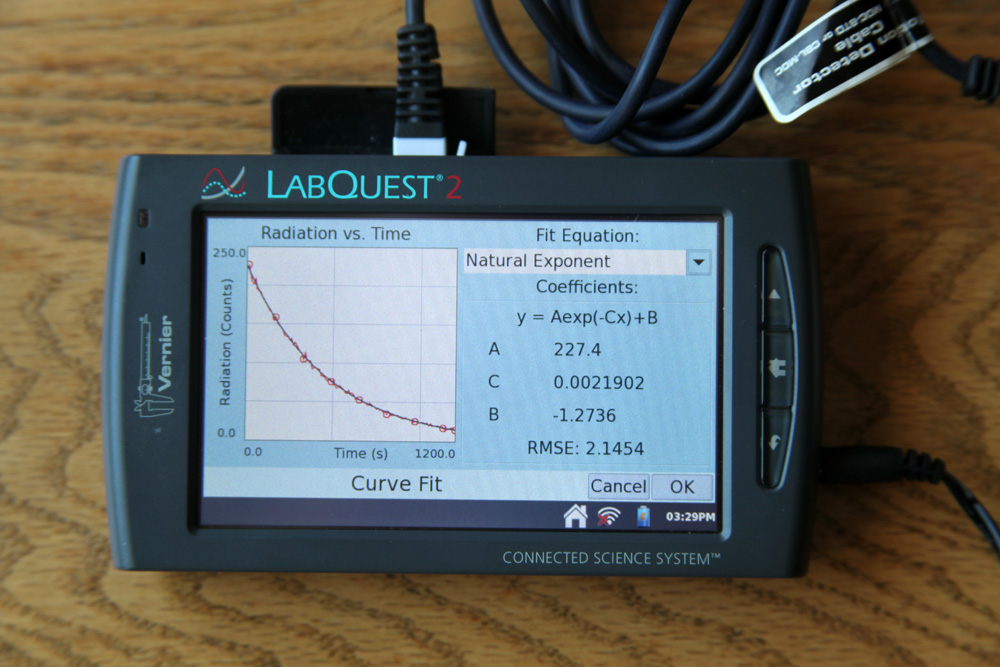


Figure 5

1. The most interesting constants from the above figure are *A* and *C*. *A* corresponds to the initial decay rate *Ro*, and *C* to the decay constant *λ*. The data shown in Figure 5 indicate that the initial decay rate is about 227 counts/10seconds. The decay constant is about 0.00219 s-1. Since *t½* = 0.693/*λ*, the half-life of our simulated isotope is about 316 seconds.
2. Students often confuse the decay constant parameter  with the half-life *t*1/2. The decay constant  is larger for more rapidly decaying elements and has dimensions of time­–1, while the half-life has dimensions of time, and is smaller for more rapidly decaying elements. The decay constant  is equal to the fit parameter C in the Natural Exponential fit of Logger *Pro* and LabQuest. The two parameters can be related in the following manner. After one half-life has elapsed, half of the radioactive nuclei have decayed, and so the activity is also cut in half. From the rate equation we can relate the decay constant to the half life.



There is sufficient information in the student guide to perform this conversion, although some students with weak algebra skills may have difficulty with it. You may choose to work through this step with your students.

1. In Step 4 of Analysis students perform a curve it on only the first 15 minutes of data. This is important, because the fit will sometimes be poorer if all 30 minutes of data are used. The counts during the first 15 minutes are largely due to the barium, while the counts in the last 15 minutes are mostly from non-barium sources. The many noisy points in the tail of the exponential may unduly influence a fit of the entire run.  
     
   You may want to have students investigate this effect, or to try various selections of data during the first 15 minutes (*e.g.*, 2–13 minutes, or 5–15). The resulting value for the lifetime will vary somewhat, giving an indication of the uncertainty of the measurement. Using our data we get variations about 0.05 minutes around the typical value shown here.
2. Note that the calculator, computer, and LabQuest versions of the activity use different notation for the fitted equation. Unlike Logger *Pro* and LabQuest, the calculator program DataRad., uses seconds as the x-axis time unit, so that the exponential fit parameter must be converted from s-1 to min-1 (s-1= 60 min-1) to obtain a lifetime in min-1.
3. This Arduino Radioactivity Half-Life Simulator gives the teacher the opportunity to have students investigate radioactive decay and half-life without using actual radioactive materials and to do so at less dollar cost. In addition, since one of four simulated isotopes and one of three initial decay rates are randomly selected for each run of the sketch, different lab groups are likely to obtain different experimental results.

sample results

As mentioned above, this simulation randomly selects one of four "isotopes". Here are statistics on actual runs with the simulation for each of the isotopes. These data are from over 40 runs with each. This should give instructors an idea of typical results.

Isotope Half Life

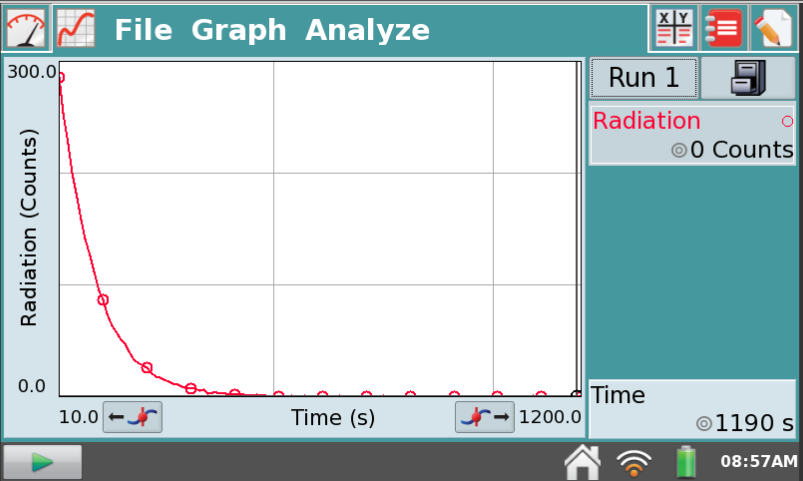
(s)

1 56 to 63

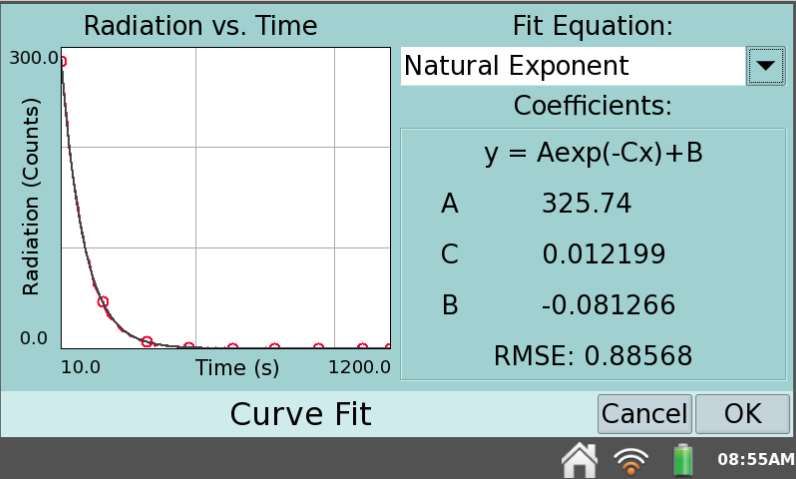
2 110 to 119

3 195 to204

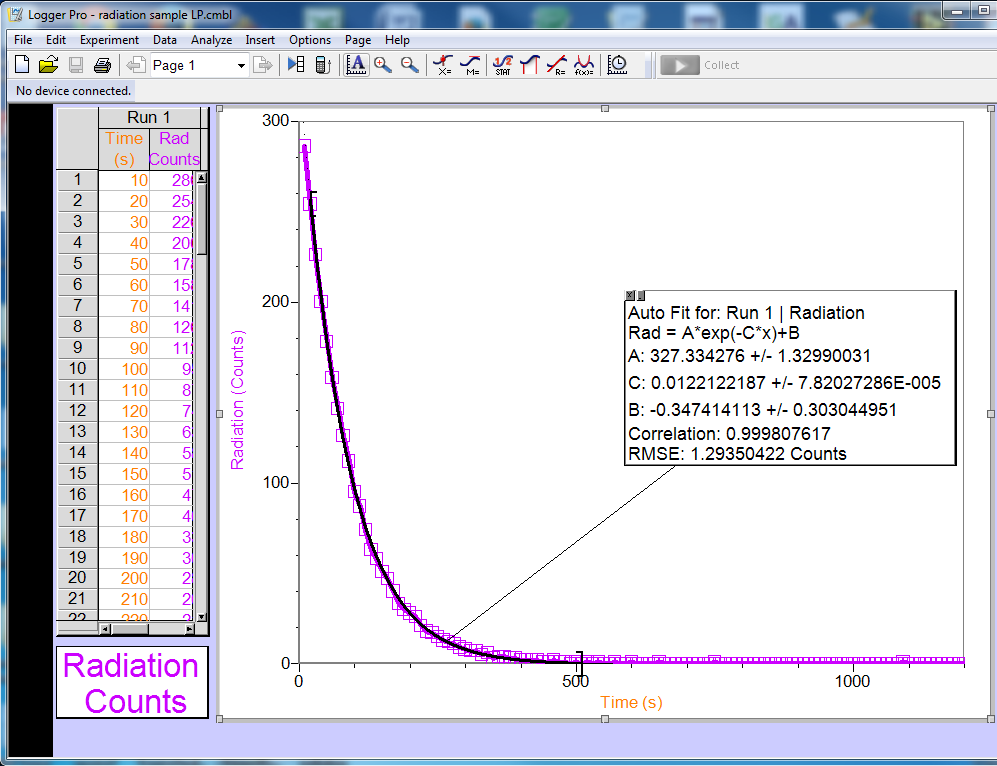
4   316 to 321



*Graph results displayed on LabQuest 2*



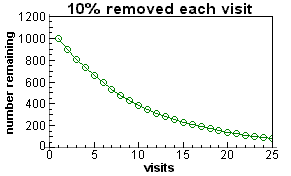
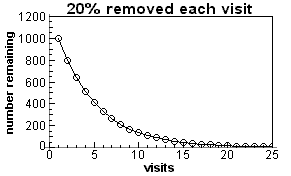
*LabQuest 2 screen after curve fit*



*A different run graphed and analyzed in Logger* Pro

answers to preliminary questions

1. Graph is a decaying exponential. The first few values are 1000, 900, 810… (with integer part of 10% taken each time).
2. Second graph decays more quickly: 1000, 800, 640…

Answers to analysis questions

1. The count rate decreases in time, falling to less than 10% of the initial value. This is consistent with activity being proportional to the amount of remaining radioactive material, since as material decays, less remains, so the activity must decrease.
2. The three graphs have a similar decreasing shape, although the time-axis scale of the barium data is different from that of the candy graphs. The vertical axes have different units (candy remaining and counts/interval). They are similar because in each case the decay process proceeds at a rate proportional to the remaining candies or radioactive nuclei.

6. We start with the rate equation, and then use the definition of the half-life as the time it takes for the activity to drop to one-half the original value:  
 **

8. Students can determine the number of half lives of their sample 25 minutes would be and use the equation

Fraction Remaining = (1/2)^ (number of half lives)

To calculate the fractional amount left after 25 minutes.

Answers to Extensions

1. A graph of ln(counts/interval) *vs.* time should be a straight line of negative slope. The slope is –, or the negative of the decay constant. With no background radiation, as is the case with this simulated data, the graph should be nearly linear.