Determining the *Length* of a *Moving* Object with Ari's *ToF* Sensor

Ozobot Editor Experiment—Student Guide

by Richard Born, Associate Professor Emeritus, Northern Illinois University

Introduction

Measuring the length of an object is much easier when the object is at rest. You can simply place the object's length along a ruler with one end at the zero mark. The mark at the other end of the object will be its length. You can also accomplish this by using Ozobot Ari's time-of flight (ToF) sensor by lining up one end of the object with the sensor and letting the beam reflect from a matte surface held at the other end of the object.

In this lesson, however, the object will be moving at a constant velocity. While it is possible to use the techniques described above, doing so would be much more difficult. For example, if you used a ruler, you would have to move the ruler along with the object keeping one end of the object at the ruler's zero mark, while simultaneously reading the mark where the other end of the object is along the ruler.

Experiment Setup

Figure 1 shows the setup for performing this experiment. Ari_1, which is at rest, emits a ToF signal about five time per second toward the background board, with the ToF data captured in an Ozobot Editor list. Ari_2, with the object (a strip of cardstock paper) of unknown length *L* attached to its side moves to the right at a constant speed *v*. This speed is set by a move block in which the move distance is the length of the background board. Note that the background board is nearly twice that of the moving object. Also, note that Ari_2 could be replaced with an Ozobot Evo, as ToF sensors are not needed here. Now would be a good time to view the video, *Determining the Length of a Moving Object.m4v,* that accompanies this lesson.

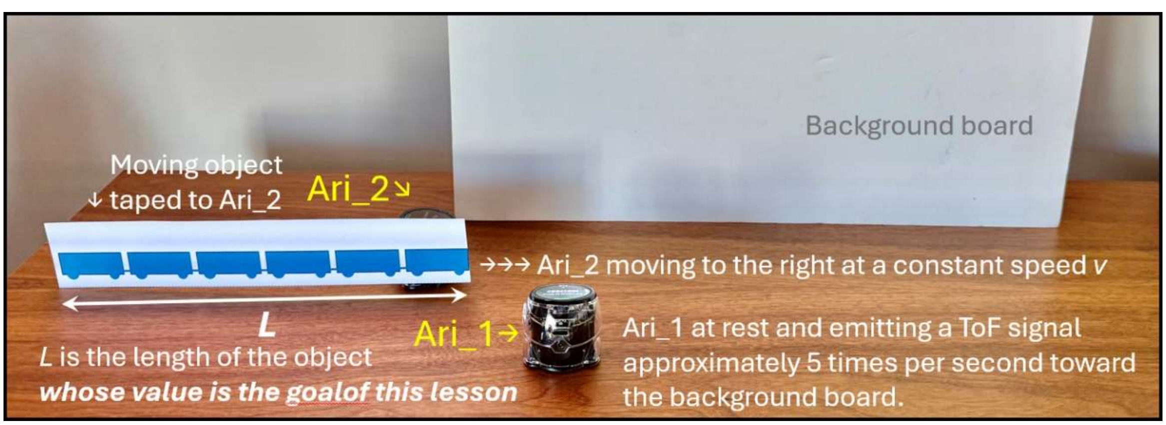

Figure 1

The Ozobot Editor Program

First, let's take a look at the four *set…to* blocks that initialize several variables used in our program. The total *runTime* for the program is set to 20 seconds, allowing enough time for the object of unknown length *L* to travel to the far-right end of the background board. If you need more that 20 seconds of run time, you may need to increase the value of the variable *runTime*. The variable *ToFcounts* keeps track of the number of ToF signals emitted by Ari_1. These ToF readings (in meters) are stored in a list called *ToFlist*. If necessary, you can increase the length of this list beyond 200 items. The Boolean variable *done,* initialized as *false*, will later be set to *true* when the total *runTime* is complete.

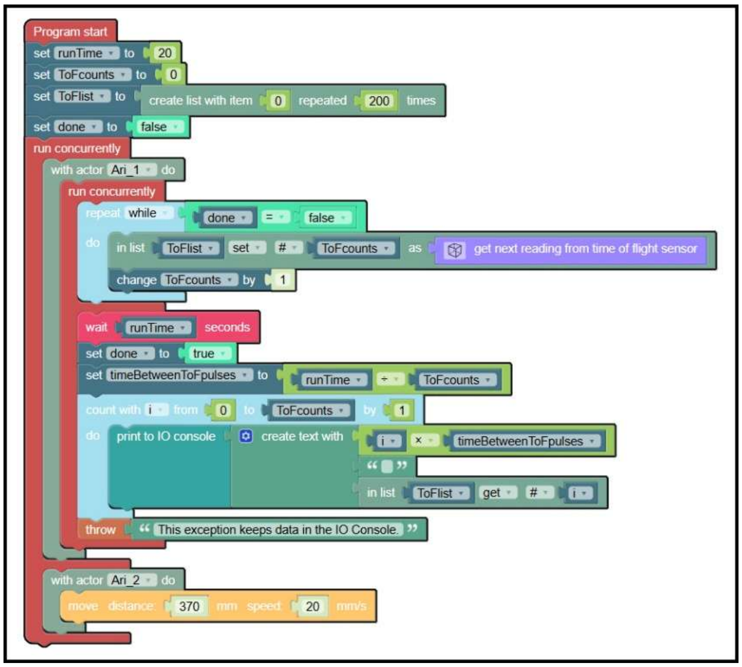

Figure 2

Now let's dive into the core of this program. It consists of three separate groups of blocks all running ***concurrently.*** The first and second group of blocks of code is assigned to actor Ari_1.

The first group of blocks gets one reading after another from Ari_1's time-of-flight sensor, storing the readings in consecutive elements of the array *ToFlist*. The second group of blocks begins by waiting for the number of seconds set by the *runTime* variable. The variable *done* is then set to *true*, stopping Ari_1 from obtaining any more ToF readings. The time between ToF pulses is then calculated by dividing the *runTime* by the total number of ToF readings stored in the variable *ToFcounts*. Finally, a loop prints the actual clock time to the IO console for each of the ToF pulses and corresponding ToF readings in meters separated by a blank space.

The third group of blocks simply starts actor Ari_2 moving at a speed of 20 mm/s (0.020 m/s) for a distance of 370 mm, the approximate length of the background board.

Typical Contents of the IO Console

You will be transferring the data in the IO console to a provided Excel spreadsheet for analysis. Figure 3 shows a portion of the typical contents of the IO console produced when running the program. To transfer the data, left-click and highlight all of the data in the IO console so that it has a gray background. Then right click anywhere in the gray region and select *Copy* from the pop-up window. This will copy the data to the Windows clipboard, so that it can then be pasted into the Excel spreadsheet. *Note that Ctrl-C will **not** copy the IO Console data to the clipboard.*

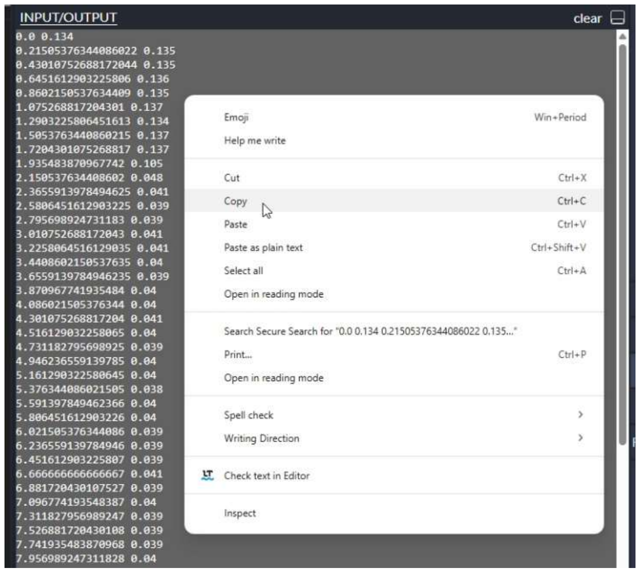

Figure 3

The Excel Spreadsheet

Figure 4 shows the first several rows and columns of the spreadsheet whose title is *Determining the Length of a Moving Object Spreadsheet.xlxs*. This spreadsheet can be downloaded by going to https://drive.google.com/uc?export=download&id=1PCa19EEwhwmMF93JzKW3g1RzIUSSR3uG. We will now discuss each of the columns A through F.

	A	B	C	D	E	F
1	Paste the data from the Ozobot Editor IO Console below:	**Character string** before the blank space:	Number of characters after the blank space:	**Character string** after the blank space:	Time (s):	ToF Reading (m):
2	0.0 0.134	=LEFT(A2,FIND(" ",A2)-1)	=LEN(A2)-FIND(" ",A2)	=RIGHT(A2,C2)	=VALUE(B2)	=VALUE(D2)
3	0.21505376344086022 0.135	=LEFT(A3,FIND(" ",A3)-1)	=LEN(A3)-FIND(" ",A3)	=RIGHT(A3,C3)	=VALUE(B3)	=VALUE(D3)
4	0.43010752688172044 0.135	=LEFT(A4,FIND(" ",A4)-1)	=LEN(A4)-FIND(" ",A4)	=RIGHT(A4,C4)	=VALUE(B4)	=VALUE(D4)
5	0.6451612903225806 0.136	=LEFT(A5,FIND(" ",A5)-1)	=LEN(A5)-FIND(" ",A5)	=RIGHT(A5,C5)	=VALUE(B5)	=VALUE(D5)
6	0.8602150537634409 0.135	=LEFT(A6,FIND(" ",A6)-1)	=LEN(A6)-FIND(" ",A6)	=RIGHT(A6,C6)	=VALUE(B6)	=VALUE(D6)

Figure 4

Column A *(Green)*
The data from the Ozobot Editor IO Console is pasted here. Each of the cells from A2 through cell A94 will contain a ***string*** similar to those shown in Figure 3.

Columns B *(Light Purple)*
The Excel LEFT function grabs the portion of the string up to but not including (explains the -1) the blank space that separates the elapsed time from the corresponding ToF reading.

Column C *(Dark Purple)*
We need to know the number of characters after the blank space. This is found by taking the length of the string and subtracting the index of the blank space.

Column D *(Light Purple)*
The Excel RIGHT function grabs the string at the far right of the string from the IO Console whose length is obtained from the formula in column C.

Column E *(Orange)*
The Excel VALUE function convert a string to a number so that calculations can be made when needed. Here, the elapsed time is converted to a number.

Column F *(Orange)*
Here, the VALUE function converts the ToF reading from a string to a number.

The Graph of ToF Reading vs. Time

This graph is shown in Figure 5. The graph was created by selecting columns E and F and then selecting *Insert/Chart/Scatter* from the Excel ribbon menu. All spreadsheet calculations and the graph appear automatically immediately after the Ozobot Editor IO Console data has been pasted into column A of the spreadsheet. You can also paste a different set of IO Console data into column A and the result and graph will be recalculated. If the number of lines in your pasted data from the IO console goes beyond row 94 in the spreadsheet, you will need to copy/paste the formulas in cells B94, C94, D94, E94, and F94 to the cells below these five cells.

The text in yellow was inserted to show you how to compute the experimental length of the moving object from the time data in the graph and Ari's speed that was set to 20 mm/s (0.020 m/s) in the move block. The inserted yellow text also shows that the experimental value of the length 0.258 m agrees to within 1.57% of the length 0.254 m as measured by an ordinary ruler.

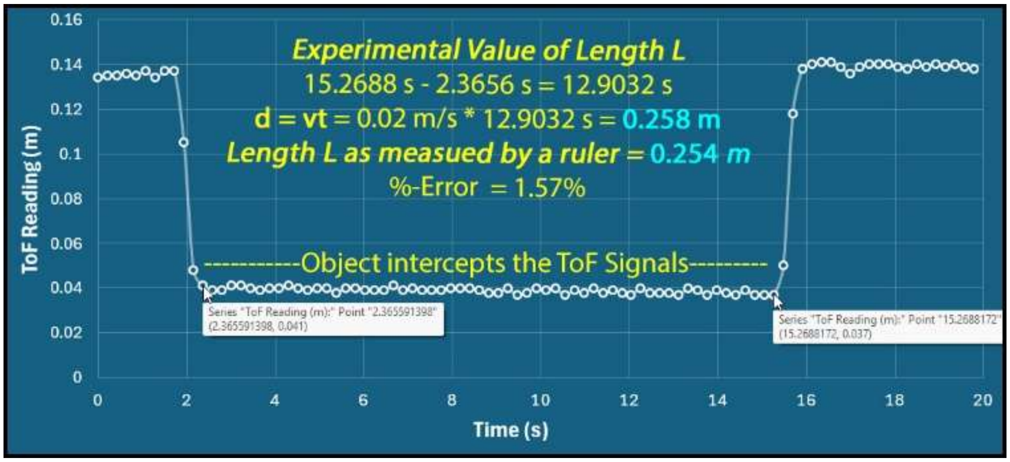

Figure 5

Running the Ozobot Editor Program

While you certainly can run the program in the Blockly Editor, it will run *much faster* if you run the program in the Python Editor. This is because time is not being used up highlighting blocks when using the Python Editor. Once the program has been loaded and connected to Ari (Share code ***fgd4hw3***), simply click on the *Code* button, and then select "Open in Python Editor". See Figure 6. You then click on the *Run* button whenever you are ready. You need not deal with the Python code or know anything about Python. ***To obtain the highest frequency of ToF readings and the most accurate length possible of the unknown object, you need to run the program in the Python Editor.***

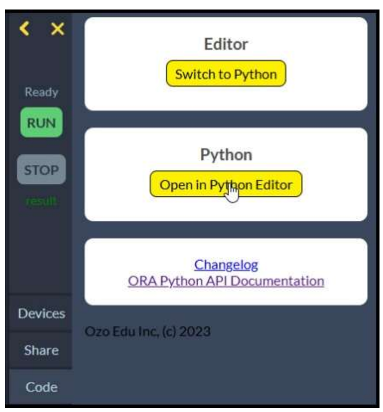

Figure 6

An Alternative Experiment

Instead of determining the length of a moving object when its speed is known (d=vt), ***you can also determine the speed of a moving object whose length is known (v=d/t)!***