

## PocketLab Voyager Study of LIDAR Basics

Richard G. Born  
Associate Professor Emeritus  
Northern Illinois University

### *Introduction*

**LIDAR**—an acronym for *Light Detection and Ranging*—is a method for remote sensing to measure distances. While LIDAR commonly uses reflected laser light to accomplish this, students can investigate LIDAR principles by using Voyager’s Gyroscope and IR Range Finder in conjunction with the PocketLab-Scratch integration. PocketLab support has described a project in which Voyager was mounted to an RC BB-8 Star Wars toy to map a two dimensional image of a “room”. In this lesson, the author has swapped the Star Wars toy with a slowly rotating turntable and has made significant changes/additions to the *sbx* ScratchX file provided by the PocketLab team. Figure 1 shows a snapshot of the project setup used by the author. The complete action video is included with this lesson.



Figure 1

The main addition to the ScratchX program was an empty sprite with code to calibrate Voyager’s data with the ScratchX plotting area. The program is started with PocketLab rotating with the turntable. The code (for Sprite1) is shown in Figure 2. The code provides for a number of things:

1. A loop is repeated 30 times with Gyro and Range Finder values captured once every second. The loop was executed 30 times to ensure that the turntable makes at least one complete rotation during calibration. The 30 Gyro values are summed in *GyroSum*, and the largest of the 30 Range Finder readings, *maxDistance*, is determined.
2. After the looping is completed, the angular velocity is determined by finding the average of the 30 Gyro readings.
3. A message is then broadcasted to the other two sprites (Arrow1 and Button1), letting them know that calibration is complete, so that they can begin the job of mapping the “room”.

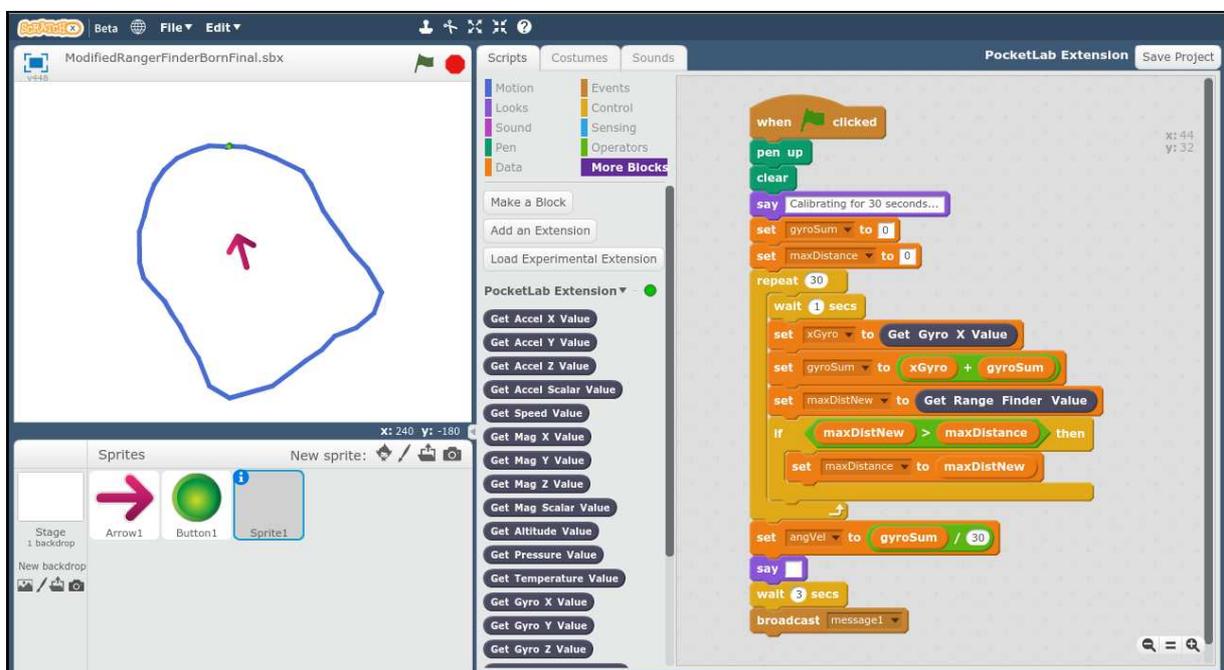


Figure 2

Figure 3 shows the code for the Button1 sprite. This code is started upon receiving the broadcasted message1 from the calibration sprite. The *maxDistance* which could be at most 2.2 meters for the Range Finder is increased by 0.05 m to play it safe. Since the y range of the ScratchX plotting area goes from -180 to 180, 180 is divided by the adjusted *maxDistance* to obtain the scale factor. This scale factor is used to compute *R* values that will fit in the ScratchX plotting area. To avoid drift in the mapping of the “room”, the forever loop needs to be stopped manually in ScratchX when one complete rotation of the turntable is completed.

Figure 4 contains the code for the Arrow1 sprite. This code is started upon receiving the broadcasted message1 from the calibration sprite. With the turntable taking 20 seconds to rotate once, the author decided to adjust the angle theta every 0.5 s, providing about 40 Voyager Range Finder observations for a single rotation of the turntable. This explains the *wait 0.5 secs* blocks in the forever loops. The variable *halfSecAngle* is the number of degrees the turntable rotates every half of a second.

```

when I receive message1
pen up
set scaleFactor to 180 / maxDistance + 0.05
set R to scaleFactor * Get Range Finder Value
clear
go to x: R * cos of theta y: R * sin of theta
set pen size to 5
pen down
set pen color to
forever
set R to scaleFactor * Get Range Finder Value
go to x: R * cos of theta y: R * sin of theta
wait 0.5 secs

```

Figure 3

```

when I receive message1
set x to 0
set y to 0
point in direction 0
set theta to 90
set halfSecAngle to angVel / 2
forever
turn -1 * halfSecAngle degrees
set theta to halfSecAngle + theta
wait 0.5 secs

```

Figure 4

Link to the PocketLab Support BB-8 mapping project:

<http://www.thepocketlab.com/support/comment/47#comment-47>

Link to a detailed lesson of LIDAR concepts by the author:

<http://www.thepocketlab.com/support/lesson/voyager-ozobot-stem-team-determine-dimensions-cardboard-box>