

Bit/Evo OzoBlockly Lesson: User Input with “Button Press Count”

By Richard Born
Associate Professor Emeritus
Northern Illinois University
rborn2@niu.edu

Introduction

In addition to its optical and color sensors, Evo has two ways of accepting input from users of an OzoBlockly program while the program is running. These relate to (1) the ability of Evo to respond to proximity sensor readings from its four IR (infrared) sensors, and (2) the ability to respond to presses of its button. Bit, of course, can accept input only through its optics and button. Therefore, in this lesson, the student will learn how to program both Bit and Evo for the latter of these two options.

Once an OzoBlockly program is running, a short press of the button normally turns Ozobot off. However, the function of the button can be altered programmatically, disabling this automatic power-off feature. The first thing the programmer must do to accomplish this is to use a special block to “set button capture events” to true. As long as button capture events are allowed, any press of the button increases the value of a special built-in variable called “Button Press Count” by 1. Then, depending on the exact value of the *Button Press Count* variable, different groups of blocks can be executed. In effect, the user clearly has the ability to supply input *while the OzoBlockly program is running*.

Ozobot’s Predicament

Decisions, decisions! Ozobot is out for a joy drive, but can’t decide whether he wants to navigate the winding roads on the nearby hills, take to the open highway for a trip, or just take a relaxing ride to view homes in the neighborhood. Because of this uncertainty, Ozobot has decided to let **you** make this decision for him.

Figure 1 shows the Ozomap that you will use to help him decide. After placing Ozobot on the start location facing the direction shown and then starting the stored OzoBlockly program, he will give you several seconds to let him know which route to take. He has told you to press the button once if you want to take the mountain view, press the button twice if you want to visit the neighborhood, or press the button three times if you want take a highway trip. He has asked that you observe all speed limits, as he does not want to be stopped by the police for a traffic violation. The neighborhood speed limit is 20, the mountain speed limit is 40, and the highway speed limit is 65. Finally, he wants to display a red top light for the mountain trip, a green top light for the neighborhood drive, and a blue top light for the highway trip. After reaching the end of a route, he wants to stop.

User Input with Button Press Count

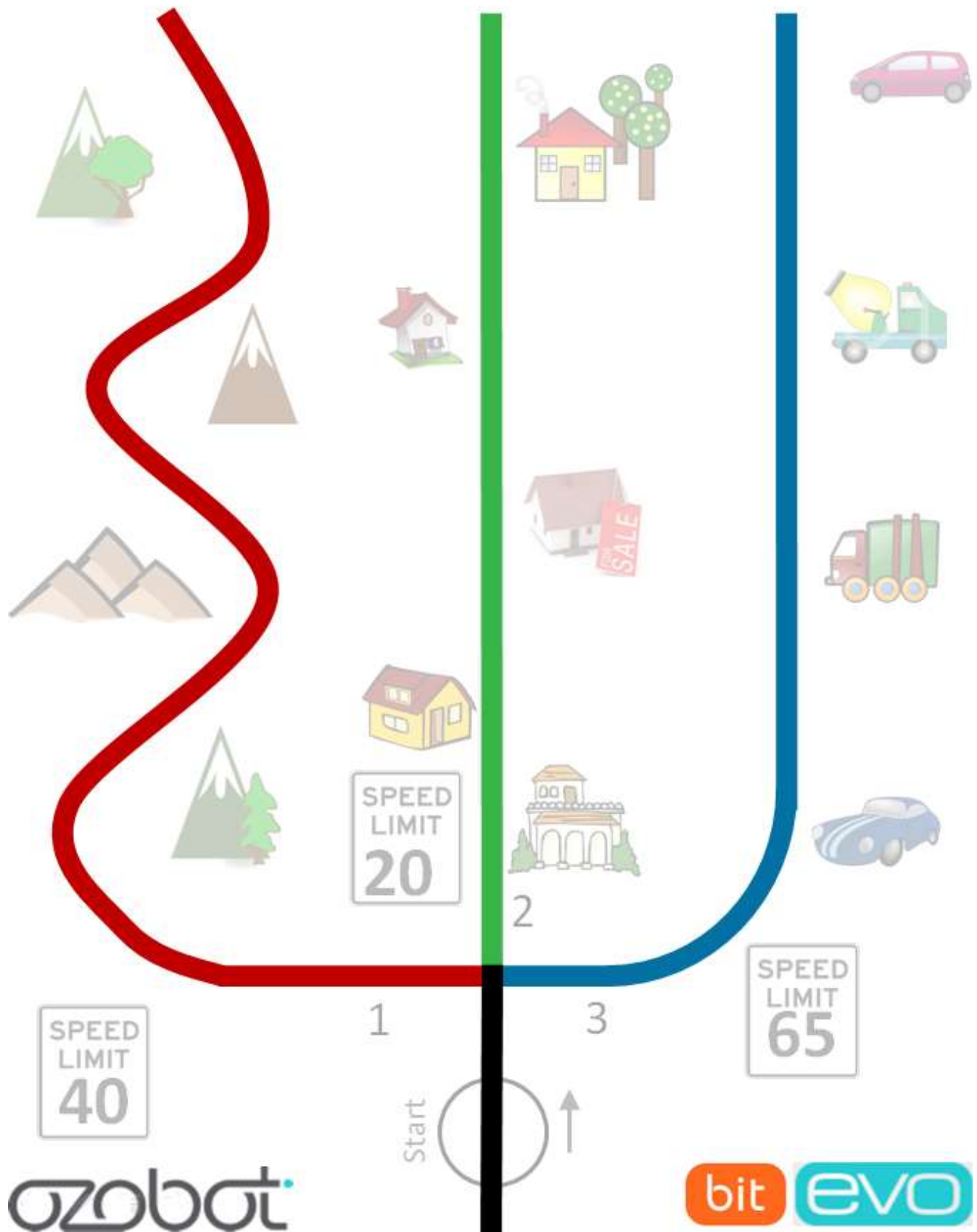


Figure 1

The OzoBlockly Program

The main program consists of calls to three subroutines as shown in Figure 2. These subroutines allow us to break the code up into smaller, more manageable functional pieces.

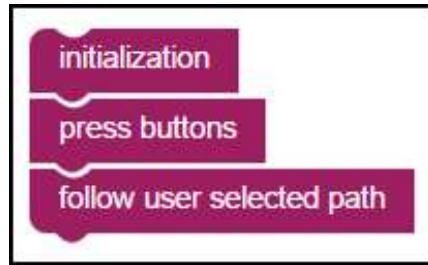


Figure 2

The initialization subroutine is shown in Figure 3. The only thing done in the initialization subroutine is to set the top LED to the color red and then wait three seconds. It should be noted that if the button is pressed during this time, then Ozobot will turn off, as we have not yet disabled the automatic power-off feature.

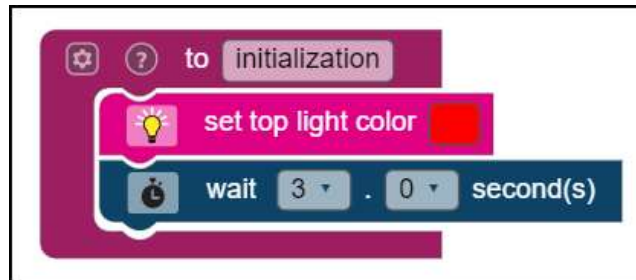


Figure 3

The “Press Buttons” Subroutine

Figure 4 shows the blocks making up the “press buttons” subroutine. This subroutine allows button presses to be recognized and counted. It begins with a block that sets the presses to zero. The button press count is then explicitly set to presses. Knowledge of this block is important as you may need to reset the button press count to zero at various points in a program if program logic requires you to do so. Capturing button press counts is set to true, and the set timer value is set to zero.

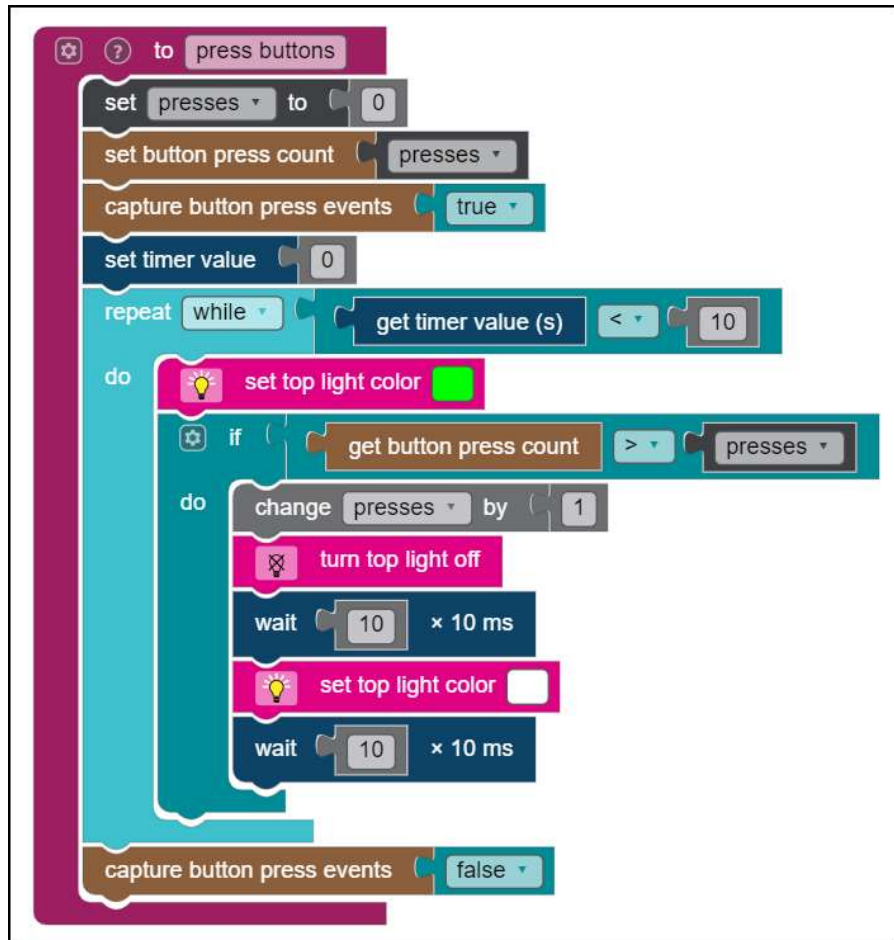


Figure 4

The next block is a repeat while loop, which repeats as long as the timer is less than 10 seconds. Inside the repeat while loop, the top light is set to the color green. If the button press count is greater than presses, presses is increased by 1 and the top light blinks white quickly to let us know that the press was recognized.

Once we exit the *repeat while* loop, we set "capture button press events" to false and are ready for Ozobot to follow the path selected by the user in the button presses.

The "Follow User Selected Path" Subroutine

Figure 5 shows the blocks that comprise the "follow user selected path" subroutine. The first thing done in this subroutine is to set the line-following speed to 30 mm/s, which is the speed Ozobot will travel on the black line at the bottom of the Ozomap. Then Ozobot is directed to follow the line to the next intersection, *i.e.*, where the red, green, and blue lines intersect.

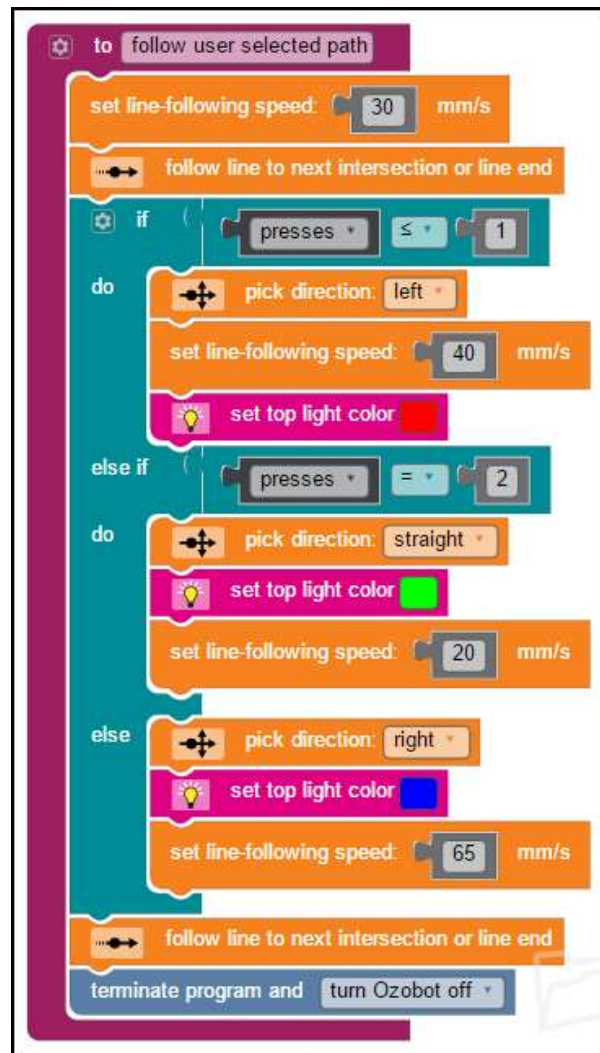


Figure 5

An *if...else if...else* block is then encountered that looks at the number of times the user pressed the button. Since we want to make sure that **all integer values** produce reasonable results, we have arbitrarily decided that in the event that the user did not press the button at all (*presses* would then be 0), Ozobot would turn left to the mountains. Similarly, if the user pressed the button more than three times, Ozobot would make a right turn toward the highway. Of course, if the button was pressed 1, 2, or 3 times, then Ozobot would turn left, go straight, or turn right, respectively, as required in the problem statement at the bottom of page 1. Each of the three groups of blocks in the *if...else if...else* also sets the top LED and Ozobot's speed in accordance with the problem statement.

After the *if...else if...else* block has been executed, Ozobot is instructed to follow the line to the next intersection or line end. It turns out to be a line-end. The program is terminated and Ozobot turns off.

Instructions for Running the OzoBlockly Program

1. Load your program onto either Bit or Evo. Details for loading programs can be found at the ozoblockly.com Web site.
2. Starting with Ozobot powered down, to run the program:
 - a. *on Bit*: Double-press the button.
 - b. *on Evo*: Press the button once to power-up. The front lights will flash blue and Evo will make a sound. Then double-press the button.
3. The top LED will be red for about three seconds.
4. When the top LED changes to green, press the button either once, twice, or three times to let Ozobot know which of the numbered paths to take on the Ozomap. ***The top LED will make a quick white flash if the button press is understood by Ozobot.*** If you press the button two or more times too quickly, Ozobot may not recognize the press. You have 10 seconds during which you can press the button the desired number of times.
5. Place Ozobot at the start location on the Ozomap. He will then follow the path that you selected to the end of the path, and then automatically turn off.