OzoBlockly Editor Challenge:

ORA Expansion Gripping: Getting a Grip from Within

Grades 9-12

Teacher Guide

by Richard Born, Associate Professor Emeritus, Northern Illinois University

Introduction

Welcome to the Teacher Guide for the ORA Expansion Gripping challenge. This lesson is designed to introduce students to advanced material-handling concepts through the use of the OzoBlockly Editor and the ORA robotic arm. Unlike traditional external pinching, expansion gripping requires students to understand internal geometries and the importance of outward pressure. This technique is an important element in industrial manufacturing, particularly when handling delicate components like gears, where external contact could damage sensitive teeth.

As an instructor, your role is to guide students through the complexities of spatial coordination and precision programming. The challenge emphasizes the "Pick and Place" workflow but adds a layer of difficulty by requiring a vertical mount. This necessitates careful calibration of joint angles and movement speeds. Safety is paramount in this lab. Ensure that all students prioritize the inclusion of speed-limiting blocks and maintain access to the Emergency Stop Button. By facilitating this challenge, you are helping students bridge the gap between basic block coding and professional automation engineering. This fosters critical thinking regarding hardware protection and operational efficiency in a controlled robotic environment.

If you have not already done so, you should now read through the *Student Guide* for this challenge for details on what is expected from the students. Also, view the two videos that accompany this lesson. One is a 7-second time-lapse video showing ORA pick and place four gears. The other is a 26-second standard speed video showing ORA pick and place the first gear only.

Share Code for the Solution – ynb2n4o

Link to 3D STL File – Vertical Gear Holder
Link to 3D STL File – 8-tooth gear X4

Example Solution to this Challenge

The solution to this challenge provides the opportunity for your students to develop an OzoBlockly Editor program that is modular, dividing the program logic into functions each having a specific purpose. Images of the modules follow with a discussion explaining the purpose of the Ozobot

Editor blocks that make up each module. A separate included PDF file also contains images of these modules, in the event that you want to discuss or share any of them with your students.

Figure 1 shows the main program and the "*Initialization*" function. The first four blocks of the initialization module are self-explanatory. The *pickUpZValue* variable is the height in mm where the top gear in the vertical stack is to be extracted. The *gearThickness* variable is the thickness of each gear in mm.

The main program begins by calling the "*Initialization*" function. Then a "count with" loop varies the loop variable *x* from 0 to 3, allowing for picking up a total four gears from the stack location and placing them on the vertical wall. For each loop iteration, the "*Pick Up Gear*" function picks up the top gear from the stack of gears. The "*Place Gear on Peg Board*" function is then called, placing the gear on the peg on vertical wall. After the pallet has been filled, ORA's arm moves to the ZERO location and the program terminates.

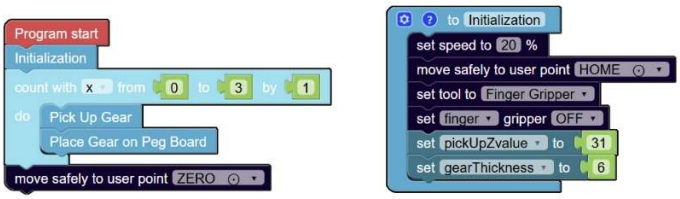

Figure 1

Figure 2 shows the "*Pick Up Gear*" function. This function directs ORA to pick up the gear that is currently at the top of the stack of gears at (*x, y*) = (241, -243). The first *move to coordinate* block moves ORA's vacuum gripper to a height of 60 mm above the pick-up location. The finger gripper is then closed. The second move block lowers the gripper so that it can grip the gear at a height of *pickUpZvalue* mm. The gripper is then opened to grip the gear from within via expansion gripping, and the gear is lifted back to a height of 60 mm above the pick-up location. The *pickUpZvalue* variable is then decreased by the thickness of a gear so that it is ready for the *next* gear to be picked up. Finally, ORA's arm is moved to the Home location, and the gripper is turned off.

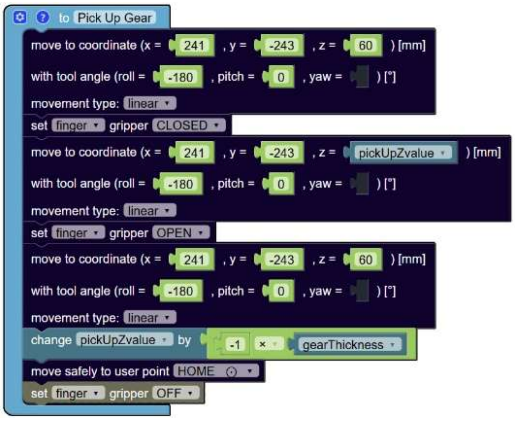

Figure 2

Figure 3 shows the "*Place Gear on Peg Board*" function. The gear is to be placed on the peg at location (x, y, z) = (230, 143, 85). The first *move to coordinate* block moves the gear to y = 0 and adjusts the pitch and yaw by 90°, orienting the gear so that it is a vertical plane at y = 0. The second *move to coordinate* block moves the gear so that its hole is centered over the end of the peg at y = 143. Then the finger gripper is CLOSED, releasing the gear onto the peg. The gripper is turned off. The third *move to coordinate* block backs the gripper away from the pegboard to y = 115 mm. Finally, ORA's arm is moved to the Home location.

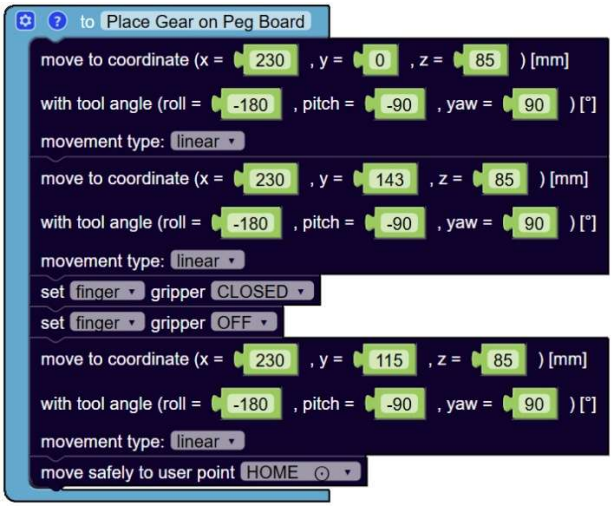

Figure 3

ISTE Standards (International Society for Technology and Education)

Empowered Learner 1.1.d: Students understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.

Knowledge Constructor 1.3.d: Students build knowledge by actively exploring real-world issues and problems, developing ideas and theories and pursuing answers and solutions.

Computational Thinker 1.5.c: Students break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.

CSTA (Computer Science Teachers Association) Standards

2-CS-02 Design projects that combine hardware and software components to collect and exchange data.

2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.

2-AP-12	Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
2-AP-13	Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.
2-AP-19	Document programs in order to make them easier to follow, test, and debug.
3A-DA-11	Create interactive data visualizations using software tools to help others better understand real-world phenomena.
3A-AP-17	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
3B-CS-02	Illustrate ways computing systems implement logic, input, and output through hardware components.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.