OzoBlockly Editor Challenge

ORA Intelligent Currency Challenge

Student Guide

by Richard Born, Associate Professor Emeritus, Northern Illinois University

Introduction

Welcome to the *ORA Intelligent Currency Challenge*. You will be stepping beyond basic automation and into the world of intelligent, vision-guided robotics. While many industrial robots are programmed to perform repetitive sorting tasks, your mission requires a higher level of integration. You will be combining ORA's 6-axis precision with real-time logic and artificial intelligence.

Using the ORA cobot and an LLM-powered vision system, you will develop a solution that does more than just recognize objects—it must understand value. The task is to scan a sequence of random coins and identify their denominations. However, instead of simply sorting them into piles, you will be building a "*Smart Deposit*" system.

The challenge is centered on a specific financial threshold. Your robot must pick up and drop coins into a collection cup, maintaining a running digital total of the value inside. The operation continues until the total value in the cup reaches exactly one dollar—or the very moment it exceeds a dollar.

To succeed in this project, you need to concentrate on three things—computer vision, kinematics, and conditional logic. You will leverage the LLM to accurately identify pennies, nickels, dimes, and quarters while programming ORA for smooth transitions between the coin source and the deposit cup. Implementing precise conditional logic ensures the system tracks the cumulative sum and stops exactly when the one-dollar goal is reached. This challenge mirrors real-world applications in automated retail and logistics, where precision and visual data drive decisions.

Lab Setup

Figure 1shows the setup for this challenge. A camera is mounted so that it can get a good closeup view of the coin, *directly above the coin* to avoid distortion. With the camera in the way, ORA cannot pick up the coin from the camera location. Therefore, the coin is *manually* placed on the white dot with a black background that helps the LLM distinguish the coin from the background. The coin is then *manually* moved to the ORA pickup location. ORA then picks up the coin with the vacuum gripper. The vacuum gripper works well as it is soft and flexible enough to conform to the shape of the coin's face. Finally, ORA moves the coin and drops it into the collection cup. Strong

double-stick tape is used to keep the collection cup from moving when ORA drops the coin. The lamp on the far left is located so that it provides nice shadows of the details on the face of the coin, improving the LLM's chances of correctly identifying the coin's denomination. It is important that the lamp is not too bright or that overhead lights wash out any of the features on the face of the coin.

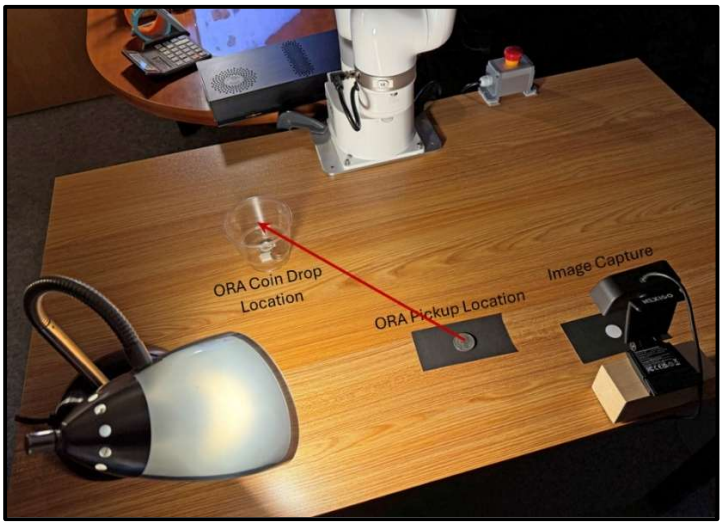

Figure 1

To get a better understanding of how the challenge works, you should view the accompanying 20-second time-lapse video that shows coins being dropped until the one-dollar goal has been reached. There is also a 2-minute video at standard speed that shows the first three coins being identified and dropped into the collection cup. This video makes it easier to see the manual and ORA movements of the coin. It is difficult to hear in the video, but as soon as the coin is dropped, Evo speaks the value of that coin followed by the total value of all the coins dropped so into the cup.

Two <u>VERY IMPORTANT</u> Notes Regarding Safety

1. You should include the Ozobot Editor block shown in Figure 2 at the beginning of your program, perhaps right after the *Program start* block or in a function where you initialize a number of things required by your solution to the challenge. Why? When ORA is moving at a slower speed, say 20% or less, you have more time to prevent an impending collision. Of course, this also allows a person more time to move out of the way of ORA's moving arm! This is particularly important throughout the development process while creating your program, when you are more apt to make errors in coordinate values in *move to coordinates* blocks.

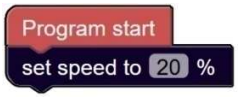

Figure 2

2. Make sure that the **Emergency Stop Button**, also shown in the upper-right side of Figure 1, is within easy reach. It might even be a good idea to have one student with his or her hand above the button when the program is started, in the event that it becomes necessary to instantly terminate all motion of ORA's arm and cancel any commands.

Dive into the future of automation with the ORA Intelligent Currency Challenge! You will combine 6-axis precision with AI-driven vision to build a "Smart Deposit" system. This hands-on project teaches you how to identify currency denominations, manage real-time logic, and master robotics kinematics. It's your chance to use cutting-edge technology to solve a real-world financial task. Step up, start programming, and gain the skills needed for the fast-paced world of advanced robotics!