

Experiment Guidelines

Read our guidelines carefully before proceeding to the experiment, and feel free to ask us any questions that you may have regarding our study.

In this experiment, you will implement a robot application using the ABB Wizard Easy Programming Tool, a block-based programming language for collaborative robots. You will be requested to write a software solution that allows customers to sort coffee cans among two can organizers using a robot. The cans, organizers, and the robot will be placed on the same table. The robot will be configured with pre-defined robot positions to grab and release the cans in both organizers, and such positions will be available for use in the programming language. Next to the robot, a FlexPendant (a screen similar to a tablet) will be available for you to program in the block-based language, and a desktop computer will be available for you to request assistance when necessary. Two flavors of coffee will be randomly placed by a stocker in each organizer: Kona Coffee and Original Coffee. The organization of the cans by the robot over the two organizers must be guided by customers via touchscreen interactions on the FlexPendant, as the order of the cans in each organizer will be random. Everything that you need for this task is available in the programming language. If you face any questions, use the desktop computer to get assistance. You may also ask the proctor for clarification.

Follow the software requirements below to guide your software implementation. Your block-based solution will be considered valid if it follows all the requirements.

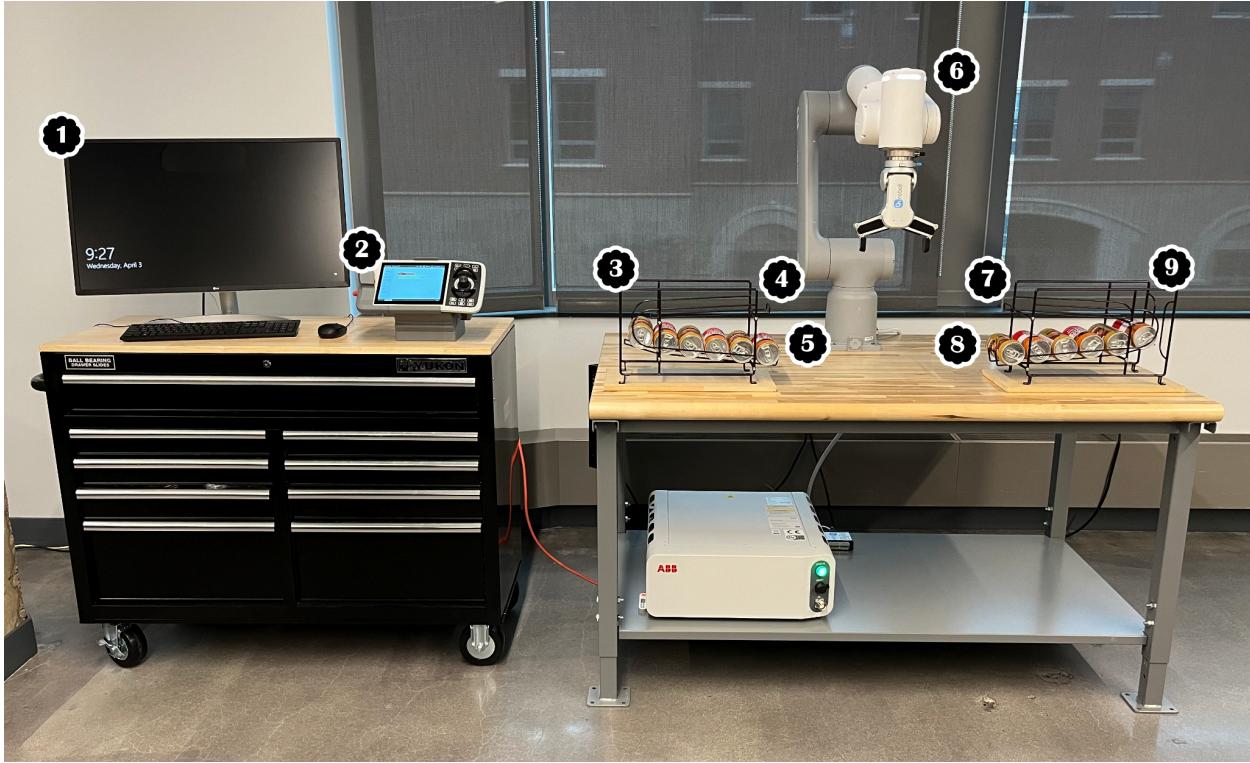
Software requirements:

- Your software must start calling the *StartGripper* procedure to activate the robot gripper.
- Your software should only use the features available in the given block-based language.
- Your software should only use the robot positions available in the block-based language. You should not create new robot positions for your solution, but you can instantiate as many other types of variables as you want.
- Your software must receive the customer's input to decide the next actions of the robot, and the customer's input must be received as touchscreen interactions on the FlexPendant screen.
- In terms of actions, the robot must be able to move the coffee cans from one organizer to the other and to move the first can of each organizer to the last position of the same organizer, based on the customer's input, and while your application is running.
- Customers should be informed about which one of the actions above the robot is performing.
- Customers should not interact directly with your code, including but not limited to moving your blocks or editing your variables.
- Your software must allow customers to decide when to stop the application. This decision must be received as a touchscreen interaction on the FlexPendant screen.
- Your software must be written within a single file, provided for you beforehand by the experiment. You should not create or load other files in the programming environment, or even rename the one opened for you.

To guide your experiment, consider the following details:

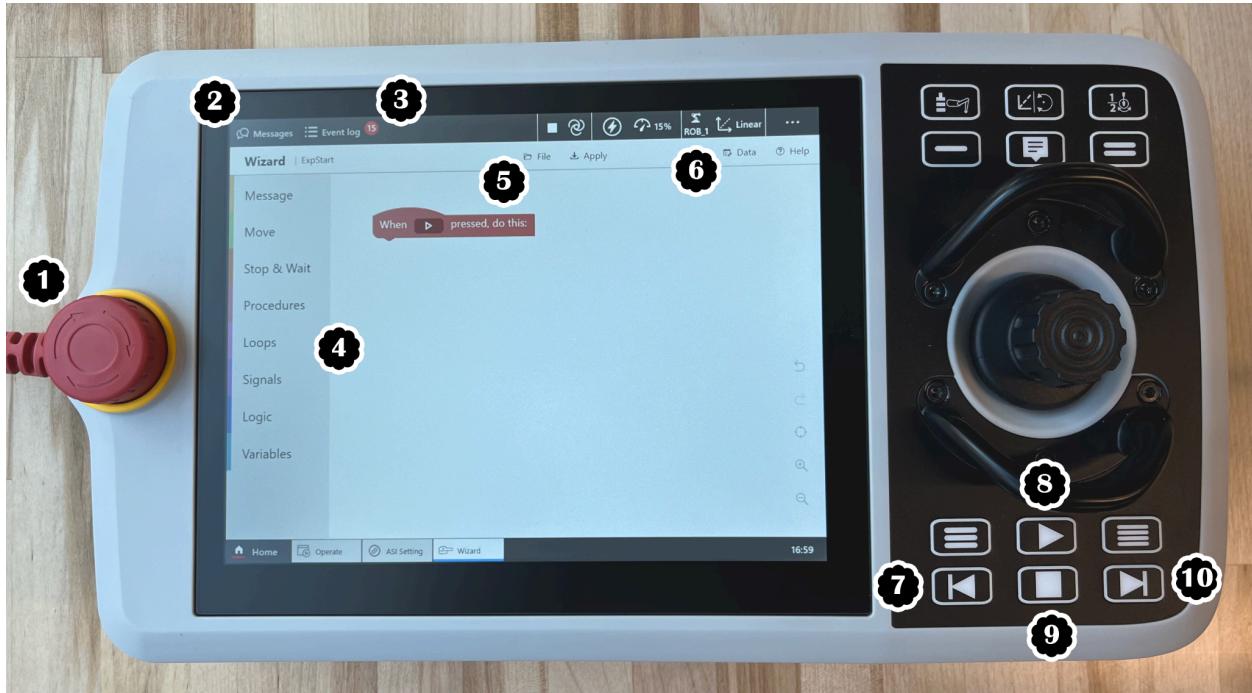
- You are expected to be able to complete the experiment within one hour, but you may have up to twenty minutes extra if you judge as necessary.
- You are free to end your participation in our experiment at any moment, even if you do not complete the task given.
- You should not physically interact with the robot or any other software application installed on the FlexPendant. Your focus should be concentrated only on the block-based language and the task assigned to you. However, you are free to use the desktop computer to request assistance as many times as you want.
- The robot is configured with pre-defined robot positions that you can use to move it around the workspace: *LeftDispenser_Bottom*, *LeftDispenser_Top*, *RightDispenser_Bottom*, *RightDispenser_Top*, and *Center*.
- Each position comes with a descriptive name. For example, *LeftDispenser_Bottom* stands for the bottom part of the left dispenser, *RightDispenser_Top* stands for the upper part of the right dispenser, and *Center* stands for the center point between the two dispensers.
- **When moving a can around the workspace, always move it to the Center position before moving it to the next desired position.** This will protect the gripper from colliding with the environment, avoiding potential damage to our equipment.
- Every time you run your solution, make sure you have one of your hands close to the stop button on the FlexPendant. If you notice that the robot will collide with itself or with the environment, use the stop button to stop the program execution and fix any problems in your solution before proceeding.
- On the desktop computer, you will find an application running with text materials, videos, and a chatroom with an expert. Feel free to use these resources as much as you want. Don't close the application running on the desktop, nor look for external resources on the computer.
- Before getting access to the resources page on the desktop application, you will be requested to inform what type of question/problem you are facing. Please, provide as much detail as you can about your current question/problem, your input will be valuable to us. Close the request every time you solve a question/problem, and start another request when necessary.
- Once you complete your participation, you will be invited to complete a post-experiment questionnaire about your experience in our study. The proctor will also inform you about the completeness of your solution.

Robot Workspace



- 1) Desktop computer
- 2) FlexPendant
- 3) Left Dispenser
- 4) *Left_Dispenser_Top* (Robot Position)
- 5) *Left_Dispenser_Bottom* (Robot Position)
- 6) Robot
- 7) *Right_Dispenser_Top* (Robot Position)
- 8) *Right_Dispenser_Bottom* (Robot Position)
- 9) Right Dispenser

Programming Environment



- 1) Emergency Button
- 2) Messages Window
- 3) Event Logs Window
- 4) Block Category
- 5) File and Apply Buttons
- 6) Data and Help (Language Manual) Buttons
- 7) Play Previous Block/Instruction
- 8) Play Program
- 9) Stop Program
- 10) Play Next Block/Instruction