Arm and Gripper Programming Guide (LabVIEW™ for LEGO® MINDSTORMS®)

Introduction:
In this guide, the Ranger Bot will be programmed to follow a line, pick up an object, and put it in a box. This guide is for use with the LabVIEW for LEGO® MINDSTORMS® programming language.

Review:
• To review the basic principles of connecting to the NXT Brick and configuring LabVIEW for LEGO MINDSTORMS for use with various sensors and motors, review the Programming Guides and Tutorial Videos from Lessons 2 and 3.
• To review the basic principles of programming in LabVIEW for LEGO MINDSTORMS, review the Programming Guides and Tutorial Videos from the Line Follower Programming Extension.
• These materials contain detailed information about how to use the Schematic Editor and how to create the structures, functions, constants, and wires that are presented in this guide.
• This program makes use of the Screen Update SubVI, provided on the TETRIX® Getting Started Guide DVD. This function uses the NXT to display custom text as well as light and ultrasonic sensor readings.
• Use these steps to add the function to the program:
  a. Within the block diagram, right-click to bring up the Functions palette and choose Select a VI.
  b. Then navigate to where the Screen Update SubVI is saved.
  c. Once the Screen Update SubVI has been located, open it and place it on the block diagram.

Getting Started:
1. Create a new Ranger robot project and name it “Arm and Gripper.”
2. Using the Schematic Editor, configure the robot to have two TETRIX 180° Servo Motors attached to Servo Ports 1 and 2. The first servo will be called Gripper, the second servo will be called Arm.

3. Create a new VI.
Arm and Gripper Programming Guide (LabVIEW™ for LEGO® MINDSTORMS®)

Initialize the Servos:
4. The code involves first initializing the servo positions, delaying for 1 second, and then entering the main code While Loop.
   a. Create the code shown, beginning with the Move Servos function, then the Wait for Time (sec) function.
   b. Note that there is a large While Loop enclosing the remainder of the code. Create this While Loop and ensure that it is wide enough to contain all of the code that follows.

Check for an Object, Grip When Found Code:
5. Inside the While Loop, the code updates the display and then checks the ultrasonic reading to determine if an object is nearby. If the object is nearby, the code stops the robot and lowers the arm. To achieve this:
   a. Create the code shown, beginning with the Screen Update SubVI, followed by a Less? comparison function, and wire the result to the conditional terminal of a Case structure.
   b. The True case of the Case structure indicates that an object has been found. Inside the Case structure, first place another Screen Update SubVI and then a Move Motors and then a Wait For Time (sec) as shown.
   c. Next, a For Loop is used to lower the servo from 100 to 190 units in steps of 1 with a 10-millisecond delay between steps.
   **Note:** A +1 (increment) function is used because the value of the index terminal starts at zero.

6. After lowering the arm, the robot delays for 0.5 seconds, closes the gripper servo, delays again, and then raises the arm. To achieve this:
   a. Create the code shown, beginning with the Wait For Time (sec) function, followed by a For Loop to close the gripper servo from 180 to 80 units in steps of 1 with a 10-millisecond delay between steps.
   b. Then, create a second For Loop to raise the arm servo from 160 to 25 units in steps of 1 with a 10-millisecond delay between steps.
   **Note:** A +1 (increment) function is used because the value of the index terminal starts at zero.
Arm and Gripper Programming Guide (LabVIEW™ for LEGO® MINDSTORMS®)

7. The Case structure code completes with a verification that the object was successfully gripped, using the touch sensor.
   a. Use a **Read Touch Sensor (pressed)** function and connect the result to the conditional terminal of another Case structure.
   b. In the True case, use the “Screen Update SubVI” to display “Status: Got It!,” then play a tone and delay one second, as shown.
   c. In the False case, display “Status: Missed It!” and delay 0.5 seconds.

![Diagram](image)

**Note:** The **Read NXT Buttons** function is wired to the condition terminal of the outer While Loop. This will ensure that the code runs until the Enter button of the NXT is pressed.

### Line Follower When No Object Detected Code:

8. Finally, the False case of the outer Case structure executes a speed-adjusted line follower code when no object is detected in close proximity. To achieve this:
   a. Use a **Greater?** comparison function on the ultrasonic reading, a **Case structure** to determine speed, and then a **Less?** comparison function on the light sensor reading to determine which motor to drive, as shown.

![Diagram](image)