User Manual: NexBot Robotics 232-001 Offline Programming & Simulation Software License

SKU: NXB-GEN-232-001 | Version: 1.0 | Brand: NexBot Robotics

Table of Contents

1. Safety Information

READ ALL SAFETY INSTRUCTIONS BEFORE OPERATION. Failure to follow safety procedures may result in serious injury or equipment damage.

WARNING: A program validated in simulation must be re-verified at low speed on the physical robot. Discrepancies between the digital twin and the real workcell can cause unexpected motion and lead to equipment damage or personnel injury.

CAUTION: Modifying core robot kinematic or dynamic parameters in the simulation without a thorough understanding can lead to highly inaccurate cycle time predictions and invalid program paths.

NOTICE: Always back up project files before making major changes or transferring programs to a controller. Data loss may occur due to file corruption or accidental deletion.

WARNING: The NexBot Robotics 232-001 license is node-locked to a single machine. Attempting to transfer or clone the license to

> another computer without using the official deactivation process will render the license invalid.

> **NOTICE:** The simulation environment is an ideal representation. It does not account for all real-world variables such as air resistance, cable wear, or temperature fluctuations which may affect robot performance.

2. Product Overview

The NexBot Robotics 232-001 Simulation Software License provides engineers and technicians with a comprehensive virtual environment for offline robot programming, system layout validation, and cycle time optimization. This powerful tool allows for the complete development, testing, and refinement of robotic applications in a digital twin environment, significantly reducing the need for physical robot access during the initial project phases and maximizing production uptime. Key features include a high-fidelity 3D kinematic simulation engine that accurately models the motion of NexBot robots, including the R-Series, C-Series, and S-Series. Users can import CAD models of tooling, fixtures, and workpieces to build a complete virtual workcell. The integrated collision detection feature prevents costly equipment damage by identifying potential interference between the robot and its surroundings before deployment. This proactive approach de-risks the commissioning process and accelerates project timelines. The software is essential for applications requiring precise path planning and efficiency, such as automated welding, complex assembly, and high-speed material handling. With the cycle time analysis module, engineers can predict operational throughput with an accuracy of +/- 2%, enabling data-driven decisions for process improvements. The license supports the simulation of up to 4 robots in a single workcell, making it suitable for designing complex, multi-arm automation systems. This single-seat license is delivered via digital download and activated with a unique key, providing full access to all simulation and offline programming features.

3. Getting Started

1. Navigating the User Interface

The main interface is divided into several key panels. The largest is the 3D Viewport, where you can see the virtual workcell. Other panels include the Program Editor for writing code, the Project Tree for managing assets, and the Output Window for system messages and cycle time results.

2. Creating a New Project

To begin, select 'File' > 'New Project'. This will open a wizard where you can name your project, select a primary NexBot robot model from the library, and define the basic layout of the workcell. You can add additional components like controllers and I/O devices later.

3. Importing Workcell Components

Use the 'Import 3D Geometry' function to add external CAD models of tooling, fixtures, and parts to your simulation. The software supports

common formats such as STEP, IGES, and STL. Once imported, these objects can be positioned and integrated into the simulation.

4. Using the Virtual Teach Pendant

The virtual teach pendant mimics the interface of a physical robot controller's teach pendant. Use it to manually 'jog' the robot in various coordinate systems (Joint, World, Tool) to test reachability and create target positions for your program.

4. Operation

Developing Robot Programs

Use the built-in Program Editor to write, edit, and debug robot logic. The editor features syntax highlighting, auto-completion, and direct integration with the simulation, allowing you to step through code and visualize the robot's actions line-by-line.

> **Tip:** Use the 'Comments' feature extensively to document your code. This makes it easier for you and others to understand the program's logic in the future.

Defining Tool Center Points (TCP)

Accurate path execution requires a correctly defined TCP. The software provides a utility to define the point, orientation, and mass of the robot's end-of-arm tooling. You can define multiple TCPs for applications involving tool changers.

Setting Up Collision Detection

To prevent virtual crashes, you must define which objects should be monitored for collisions. In the Collision Detection menu, create pairs or groups of objects (e.g., robot arm vs. fixture) that the software will monitor during program execution. A detected collision will halt the simulation and highlight the involved components.

> **Tip:** For faster simulations, only include essential objects in the collision detection sets. Avoid monitoring non-critical items like small bolts or cosmetic features.

Analyzing Cycle Time

After a program is complete, run the Cycle Time Analysis tool. The software will execute the entire program at maximum virtual speed and generate a detailed report, breaking down the time spent on motion, logic, and I/O waits. This is crucial for optimizing the application for throughput.

Transferring Programs to a Controller

The NexBot Robotics 232-001 software can connect directly to a physical or virtual network of NexBot controllers. Use the 'Controller Manager' to establish a connection and transfer your completed and validated offline programs for real-world execution.

> **Tip:** Always perform a full backup of the robot controller before loading new programs from the simulation software.

5. Maintenance Schedule

Interval	Task	Notes
Weekly	Create backups of all active project files.	Store backups on a separate physical drive or secure network location to protect against local hardware failure.
Monthly	Check for software updates and patches via the application's built-in update utility or the NexBot customer portal.	Review the release notes for any updates to understand new features, bug fixes, and potential compatibility changes.
Quarterly	Archive completed or inactive projects to long-term storage.	This helps keep the primary workspace organized and reduces the size of local backups.
Quarterly	Review and clean up the library of imported CAD models, removing duplicates or obsolete versions.	A well-organized library improves project loading times and reduces errors.
Annually	Review the status of the software license. If using a subscription-based license, verify the renewal date.	Contact NexBot sales for information on renewals or upgrading license tiers.
As Needed	Update the local robot and controller model libraries when new NexBot hardware is purchased or firmware is updated.	Using the correct model version is critical for an accurate simulation.

6. Troubleshooting

Symptom	Possible Cause	Solution
Software fails to start with a 'License Not Found' error.	The license was not activated, the license service is not running, or the license has expired.	Run the License Manager to activate your key. If already activated, reboot the PC to restart the licensing service. Check the 'About' dialog for expiration dates.

Symptom	Possible Cause	Solution
The 3D simulation environment is slow, lagging, or displays graphical artifacts.	The workstation's graphics card driver is outdated, or the PC does not meet the recommended system requirements.	Update the graphics card driver to the latest version from the manufacturer's website. Close other resource-intensive applications. If the problem persists, consider upgrading the workstation hardware.
An imported CAD model is missing, incorrectly positioned, or has broken geometry.	The source CAD file may be corrupted, exported with an incorrect origin point, or in an unsupported format version.	Re-export the model from the source CAD software, ensuring the origin is correctly placed. Try exporting in a different neutral format like STEP or Parasolid XT. Use the software's 'Geometry Tools' to attempt a repair.
The simulated robot motion does not accurately match the physical robot's motion.	Mismatch in Tool Center Point (TCP), payload data, kinematic model, or controller firmware version between simulation and reality.	Carefully re-measure and verify all TCP and payload data on the physical robot and enter the exact same values into the simulation. Ensure the robot model and controller version selected in the software match the physical hardware.
Collision detection fails to register an obvious collision between two objects.	One or both of the objects were not included in the active collision detection set.	Open the Collision Detection settings menu. Verify that a rule exists for the pair of objects in question and that it is enabled.
Cannot establish a connection to the physical robot controller.	Incorrect IP address, subnet mask, or a network firewall is blocking the connection.	Verify the network settings on both the workstation and the robot controller. Ping the controller's IP address from the workstation to test basic connectivity. Consult your IT department to ensure necessary ports are open on the firewall.
The application crashes when running a complex simulation.	The workstation is running out of available RAM, or the project contains a corrupted element.	Try closing all other applications to free up system resources. Monitor RAM usage during simulation. If the crash is repeatable, try removing recently added

Symptom	Possible Cause	Solution
		components from the simulation to isolate the problematic element.

7. Technical Specifications

Parameter	Value	Unit
Country of Origin	US	