

# **CR5 Automatic Launcher**

Operating System Technical Documentation and Network  
Configuration

Robotics team (that would be nice... but that's just me)

October 4, 2025

## Contents

<b>1</b>	<b>Detailed Script Operation Guide</b>	<b>3</b>
1.1	Script Architecture Overview . . . . .	3
1.2	Step-by-Step Process Analysis . . . . .	3
1.2.1	Step 1: Network Interface Detection and Configuration . . . . .	3
1.2.2	Step 2: Robot Connectivity Verification . . . . .	5
1.2.3	Step 3: ROS2 Environment Validation . . . . .	5
1.2.4	Step 4: Robot Driver Initialization . . . . .	6
1.2.5	Step 5: MoveIt Planning System Launch . . . . .	7
1.2.6	Step 6: System Readiness Verification . . . . .	8
1.2.7	Step 7: Movement Node Launch Options . . . . .	8
1.3	Process Management and Cleanup . . . . .	9
1.3.1	PID Tracking System . . . . .	9
1.3.2	Cleanup Function . . . . .	9
<b>2</b>	<b>Complete Usage Instructions</b>	<b>10</b>
2.1	Prerequisites . . . . .	10
2.1.1	System Requirements . . . . .	10
2.1.2	Software Dependencies . . . . .	10
2.1.3	Workspace Preparation . . . . .	10
2.2	Initial Setup Process . . . . .	11
2.2.1	Script Preparation . . . . .	11
2.2.2	Robot Hardware Setup . . . . .	11
2.3	Step-by-Step Execution Guide . . . . .	11
2.3.1	Method 1: Fully Automated Execution . . . . .	11
2.3.2	Method 2: Manual Step Verification . . . . .	12
2.4	System Shutdown and Cleanup . . . . .	13
2.4.1	Automated Cleanup . . . . .	13
2.4.2	Manual Cleanup (if needed) . . . . .	13
2.5	Troubleshooting Common Issues . . . . .	14
2.5.1	Network Configuration Problems . . . . .	14
2.5.2	Robot Connectivity Issues . . . . .	14
2.5.3	ROS2 Environment Issues . . . . .	14
2.5.4	Driver and MoveIt Issues . . . . .	15
2.6	Advanced Configuration Options . . . . .	15
2.6.1	Customizing Network Settings . . . . .	15
2.6.2	Timeout Adjustments . . . . .	15
2.6.3	Debug Mode . . . . .	16

# 1 Detailed Script Operation Guide

This section provides a comprehensive explanation of how the automated startup script works, including detailed analysis of each step and complete usage instructions.

## 1.1 Script Architecture Overview

The `dobot_startup.sh` script follows a modular architecture with clear separation of concerns:

- **Configuration Management:** Centralized variable definitions
- **Logging System:** Color-coded output with different log levels
- **Error Handling:** Robust error checking with graceful failure handling
- **Process Management:** PID tracking and cleanup mechanisms
- **Interactive Mode:** User guidance and confirmations

## 1.2 Step-by-Step Process Analysis

### 1.2.1 Step 1: Network Interface Detection and Configuration

**Function:** `detect_network_interface()` **Purpose:** Automatically detects available network interfaces and allows user selection.

**Process Flow:**

#### 1. Interface Discovery:

```
1 interfaces=$(ip link show | grep -E "[0-9]+:" | grep -v lo | awk -F':_' '{print $2}' | cut -d'@' -f1)
```

2. **User Selection:** Presents interactive menu with all available interfaces
3. **Manual Override:** Allows manual interface name input if needed
4. **Validation:** Stores selected interface in `NETWORK_INTERFACE` variable

**Error Conditions:**

- No network interfaces found (excluding loopback)
- Invalid user selection
- Manually entered interface doesn't exist

**Function:** `configure_network()` **Purpose:** Configures the selected network interface with static IP for robot communication.

**Technical Details:**

- **Target IP:** 192.168.5.100/24 (configurable)
- **Robot IP:** 192.168.5.1 (standard Dobot configuration)
- **Subnet:** 192.168.5.0/24 (Class C private network)

**Process Flow:**

1. Interface Existence Check:

```
1      if ! ip link show "$NETWORK_INTERFACE" &>/dev/null;
2          then
3      log_error "Network interface $NETWORK_INTERFACE does not exist!"
4      return 1
5      fi
```

2. NetworkManager Conflict Prevention:

```
1      # Disable NetworkManager management to avoid conflicts
2      if command -v nmcli &>/dev/null; then
3      sudo nmcli dev set "$NETWORK_INTERFACE" managed no
4      sudo systemctl restart NetworkManager
5      sleep 2 # Wait for NetworkManager to restart
6      fi
```

3. IP Conflict Detection:

```
1      if ip addr show "$NETWORK_INTERFACE" | grep -q "$LOCAL_IP"; then
2      log_warning "IP $LOCAL_IP already configured"
3      return 0
4      fi
```

4. IP Address Assignment:

```
1      sudo ip addr add "$LOCAL_IP/24" dev "$NETWORK_INTERFACE"
```

5. Interface Activation:

```
1      sudo ip link set "$NETWORK_INTERFACE" up
```

**NetworkManager Integration:**

- **Automatic Detection:** Checks if NetworkManager is installed using command `-v nmcli`

- **Management Disable:** Prevents NetworkManager from automatically managing the robot interface
- **Service Restart:** Ensures NetworkManager applies the new configuration
- **Graceful Fallback:** Continues operation even if NetworkManager commands fail
- **Conflict Prevention:** Eliminates interference between manual IP configuration and automatic DHCP

### 1.2.2 Step 2: Robot Connectivity Verification

**Function:** `test_robot_connectivity()` **Purpose:** Verifies network connectivity to the robot before proceeding with software initialization.

**Technical Implementation:**

```
1  if ping -c 3 -W 2 "$ROBOT_IP" &>/dev/null; then
2  log_success "Robot is reachable at $ROBOT_IP"
3  return 0
4  else
5  log_error "Cannot reach robot at $ROBOT_IP"
6  return 1
7  fi
```

**Parameters Explained:**

- `-c 3`: Send exactly 3 ping packets
- `-W 2`: Wait maximum 2 seconds for response
- `&>/dev/null`: Suppress ping output (only check return code)

**Failure Diagnostics:** The script provides comprehensive troubleshooting guidance:

- Robot power status check
- IP configuration verification
- Physical connection validation
- Network configuration review

### 1.2.3 Step 3: ROS2 Environment Validation

**Function:** `check_ros2_environment()` **Purpose:** Ensures ROS2 environment is properly configured and workspace is available.

**Environment Checks:**

#### 1. ROS2 Distribution Check:

```
1  if [ -z "$ROS_DISTRO" ]; then
2  log_error "ROS2 environment not sourced!"
3  return 1
4  fi
```

## 2. Workspace Discovery:

```
1 if [ -f "$HOME/dobot_v3/install/setup.bash" ]; then
2 source "$HOME/dobot_v3/install/setup.bash"
3 log_success "Workspace sourced successfully"
4 fi
```

### Required Environment Variables:

- ROS\_DISTRO: Must be set (typically "humble")
- ROS\_VERSION: Should be "2" for ROS2
- AMENT\_PREFIX\_PATH: Workspace package paths

### 1.2.4 Step 4: Robot Driver Initialization

**Function:** `start_robot_drivers()` **Purpose:** Launches the low-level robot drivers that communicate directly with the Dobot CR5 hardware.

#### Process Implementation:

##### 1. Duplicate Process Check:

```
1 if pgrep -f "dobot_bringup_ros2.launch.py" &>/dev/null;
   then
2 log_warning "Robot drivers appear to be already running"
   "
3 return 0
4 fi
```

##### 2. Driver Launch:

```
1 ros2 launch dobot_bringup_v3 dobot_bringup_ros2.launch.
   py &
2 DRIVER_PID=$!
```

##### 3. Initialization Wait:

```
1 sleep 30 # Allow 30 seconds for driver initialization
```

##### 4. Process Validation:

```
1 if kill -0 $DRIVER_PID 2>/dev/null; then
2 echo $DRIVER_PID > /tmp/dobot_driver.pid
3 return 0
4 fi
```

### Driver Responsibilities:

- Hardware communication protocol handling

- Joint state publishing (/joint\_states topic)
- Command interface for robot control
- Safety monitoring and emergency stops
- Robot status reporting

### 1.2.5 Step 5: MoveIt Planning System Launch

**Function:** `start_moveit_system()` **Purpose:** Initializes the MoveIt motion planning framework for high-level robot control.

#### System Components:

- **Move Group Node:** Core planning and execution coordinator
- **Planning Scene:** 3D environment representation with obstacles
- **Planning Pipeline:** Motion planning algorithms (OMPL)
- **Controller Manager:** Interface to robot controllers
- **Reworked\_map:** Activation of `reworked_map_node` to translate zed messages into the desired format
- **RViz Integration:** Visualization and debugging interface

#### Launch Process:

##### 1. Conflict Detection:

```
1     if pgrep -f "full_bringup.launch.py" &>/dev/null; then
2         log_warning "MoveIt_system_appears_to_be_already_
           running"
3         return 0
4     fi
```

##### 2. System Launch:

```
1     ros2 launch cr5_moveit full_bringup.launch.py &
2     MOVEIT_PID=$!
```

##### 3. Extended Initialization:

```
1     sleep 45 # MoveIt requires longer initialization time
```

#### Initialization Timeline:

- **0-15s:** Parameter loading and node initialization
- **15-30s:** Planning scene setup and URDF parsing
- **30-45s:** Planning pipeline configuration and ready state

### 1.2.6 Step 6: System Readiness Verification

**Function:** `verify_system_readiness()` **Purpose:** Performs comprehensive system checks to ensure all components are operational.

#### Verification Tests:

##### 1. Robot Topics Check:

```
1     timeout 10 ros2 topic list | grep -q "/joint_states" ||  
2     {  
3         log_error "Robot_␣joint_states_␣topic_␣not_␣found!"  
4         return 1  
5     }
```

##### 2. Planning Service Check:

```
1     timeout 10 ros2 service list | grep -q "  
2         plan_kinematic_path" || {  
3         log_error "MoveIt_␣planning_␣service_␣not_␣found!"  
4         return 1  
5     }
```

#### Critical System Topics:

- `/joint_states`: Real-time joint position feedback
- `/robot_description`: URDF model information
- `/planning_scene`: 3D environment representation
- `/move_group/display_planned_path`: Trajectory visualization

#### Essential Services:

- `/plan_kinematic_path`: Motion planning service
- `/execute_trajectory`: Trajectory execution service
- `/get_planning_scene`: Environment query service
- `/clear_octomap`: Obstacle map management

### 1.2.7 Step 7: Movement Node Launch Options

**Function:** `start_movement_node()` **Purpose:** Provides user options for launching the actual movement control application.

#### Available Options:

##### 1. Modular Implementation:

```
1     ros2 run cr5_moveit_cpp_demo move_cr5_node_modular
```



## 2. Original Monolithic Version:

```
1  ros2 run cr5_moveit_cpp_demo move_cr5_node
```

### Interactive Launch:

```
1  read -p "Start movement node now? (y/N): " -n 1 -r
2  if [[ $REPLY =~ ^[Yy]$ ]]; then
3  ros2 run cr5_moveit_cpp_demo move_cr5_node_modular
4  fi
```

## 1.3 Process Management and Cleanup

### 1.3.1 PID Tracking System

**Purpose:** Maintains process identifiers for proper cleanup and monitoring.

**Implementation:**

- **Driver PID Storage:** /tmp/dobot\_driver.pid
- **MoveIt PID Storage:** /tmp/dobot\_moveit.pid
- **Process Validation:** kill -0 \$PID for existence check

### 1.3.2 Cleanup Function

**Function:** cleanup() **Purpose:** Ensures proper termination of all spawned processes on script exit.

**Cleanup Process:**

1. Read stored PIDs from temporary files
2. Validate process existence with kill -0
3. Send termination signals to active processes
4. Remove temporary PID files
5. Log cleanup operations

**Trap Implementation:**

```
1  trap cleanup EXIT
```

This ensures cleanup occurs on:

- Normal script completion
- User interruption (Ctrl+C)
- Script errors and early termination
- System signals (SIGTERM, SIGHUP)

## 2 Complete Usage Instructions

### 2.1 Prerequisites

Before using the automated startup script, ensure the following prerequisites are met:

#### 2.1.1 System Requirements

- **Operating System:** Ubuntu 22.04 LTS (recommended)
- **ROS2 Distribution:** ROS2 Humble Hawksbill
- **Network Interface:** Ethernet port for robot connection
- **Sudo Access:** Required for network configuration
- **Workspace:** Properly built ROS2 workspace with Dobot packages

#### 2.1.2 Software Dependencies

```
1  # Core ROS2 packages
2  sudo apt update
3  sudo apt install ros-humble-desktop-full
4
5  # MoveIt2 packages
6  sudo apt install ros-humble-moveit
7
8  # Additional utilities
9  sudo apt install net-tools iputils-ping
```

Listing 1: Install Required Packages

#### 2.1.3 Workspace Preparation

```
1  # Navigate to workspace
2  cd ~/dobot_v3
3
4  # Build the workspace
5  colcon build --packages-select cr5_moveit_cpp_demo
6
7  # Source the workspace
8  source install/setup.bash
```

Listing 2: Workspace Setup

## 2.2 Initial Setup Process

### 2.2.1 Script Preparation

1. Navigate to Package Directory:

```
1 cd ~/dobot_v3/src/cr5_moveit_cpp_demo
```

2. Verify Script Permissions:

```
1 ls -la dobot_startup.sh dobot_stop.sh
```

3. Make Scripts Executable (if needed):

```
1 chmod +x dobot_startup.sh dobot_stop.sh
```

### 2.2.2 Robot Hardware Setup

1. Physical Connections:

- Connect Ethernet cable between PC and robot
- Ensure robot is powered on
- Verify robot's teach pendant shows network connectivity

2. Robot IP Configuration:

- Access robot's network settings via teach pendant
- Set robot IP to 192.168.5.1
- Set subnet mask to 255.255.255.0
- Save configuration and restart robot if required

## 2.3 Step-by-Step Execution Guide

### 2.3.1 Method 1: Fully Automated Execution

Single Command Launch:

```
1 cd ~/dobot_v3/src/cr5_moveit_cpp_demo
2 ./dobot_startup.sh
```

Expected Interaction Flow:

1. Interface Selection:

Available network interfaces:

- 1) enp2s0
- 2) wlp3s0
- 3) Manual Input

Please select: 1

## 2. Network Configuration:

```
[INFO] Configuring network interface enp2s0...  
[SUCCESS] IP address configured successfully
```

## 3. Robot Connectivity Test:

```
[INFO] Testing connectivity to robot at 192.168.5.1...  
[SUCCESS] Robot is reachable at 192.168.5.1
```

## 4. System Launch Sequence:

```
[INFO] Starting Dobot CR5 drivers...  
[INFO] Waiting for drivers to initialize (30 seconds)...  
[SUCCESS] Robot drivers started successfully  
  
[INFO] Starting MoveIt and full robot system...  
[INFO] Waiting for MoveIt to initialize (45 seconds)...  
[SUCCESS] MoveIt system started successfully
```

## 5. Final Confirmation:

```
[SUCCESS] DOBOT CR5 SYSTEM READY!  
Start movement node now? (y/N): y
```

### 2.3.2 Method 2: Manual Step Verification

For debugging or learning purposes, you can manually execute individual steps:

#### 1. Network Configuration:

```
1      # Replace enp2s0 with your interface name  
2      sudo ip addr add 192.168.5.100/24 dev enp2s0  
3      sudo ip link set enp2s0 up  
4  
5      # Test connectivity  
6      ping -c 3 192.168.5.1
```

#### 2. ROS2 Environment:

```
1      # Source ROS2  
2      source /opt/ros/humble/setup.bash  
3  
4      # Source workspace  
5      source ~/dobot_v3/install/setup.bash
```

```
6  
7     # Verify environment  
8     echo $ROS_DISTRO
```

### 3. Robot Drivers:

```
1     # Terminal 1: Start drivers  
2     ros2 launch dobot_bringup_v3 dobot_bringup_ros2.launch.  
        py
```

### 4. MoveIt System:

```
1     # Terminal 2: Start MoveIt (after drivers are running)  
2     ros2 launch cr5_moveit full_bringup.launch.py
```

### 5. Movement Node:

```
1     # Terminal 3: Start movement application  
2     ros2 run cr5_moveit_cpp_demo move_cr5_node_modular
```

## 2.4 System Shutdown and Cleanup

### 2.4.1 Automated Cleanup

```
1     cd ~/dobot_v3/src/cr5_moveit_cpp_demo  
2     ./dobot_stop.sh
```

#### Cleanup Output:

```
[INFO] Stopping movement nodes...  
[SUCCESS] Movement nodes stopped
```

```
[INFO] Stopping MoveIt system...  
[SUCCESS] MoveIt system stopped
```

```
[INFO] Stopping robot drivers...  
[SUCCESS] Robot drivers stopped
```

```
[SUCCESS] All Dobot processes stopped successfully
```

### 2.4.2 Manual Cleanup (if needed)

```
1     # Kill all Dobot-related processes  
2     pkill -f dobot  
3     pkill -f moveit  
4     pkill -f move_cr5  
5
```

```
6 # Remove network configuration (optional)
7 sudo ip addr del 192.168.5.100/24 dev enp2s0
```

## 2.5 Troubleshooting Common Issues

### 2.5.1 Network Configuration Problems

**Issue:** Permission denied when configuring network

```
1 # Solution: Ensure sudo access
2 sudo -v # Verify sudo access
3 ./dobot_startup.sh # Try again
```

**Issue:** Network interface not found

```
1 # List all interfaces
2 ip link show
3
4 # Use correct interface name in script or select manually
```

### 2.5.2 Robot Connectivity Issues

**Issue:** Robot not responding to ping

- Verify robot power and network LED status
- Check Ethernet cable connection
- Confirm robot IP configuration (192.168.5.1)
- Test with different Ethernet port if available

**Issue:** Intermittent connectivity

```
1 # Disable NetworkManager for the interface
2 sudo nmcli dev set enp2s0 managed no
3 sudo systemctl restart NetworkManager
4
5 # Reconfigure network manually
6 sudo ip addr add 192.168.5.100/24 dev enp2s0
```

### 2.5.3 ROS2 Environment Issues

**Issue:** ROS\_DISTRO not set

```
1 # Source ROS2 environment
2 source /opt/ros/humble/setup.bash
3
4 # Add to bashrc for persistence
5 echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc
```

**Issue:** Workspace not found

```
1  # Rebuild workspace
2  cd ~/dobot_v3
3  colcon build --packages-select cr5_moveit_cpp_demo
4
5  # Source workspace
6  source install/setup.bash
```

## 2.5.4 Driver and MoveIt Issues

**Issue:** Drivers fail to start

- Check robot connectivity first
- Verify no conflicting processes are running
- Review driver logs for specific error messages
- Ensure all required packages are installed

**Issue:** MoveIt initialization fails

```
1  # Check for missing dependencies
2  ros2 pkg list | grep moveit
3
4  # Verify robot description
5  ros2 param get /robot_description robot_description
```

## 2.6 Advanced Configuration Options

### 2.6.1 Customizing Network Settings

Edit the script variables at the top of `dobot_startup.sh`:

```
1  # Custom robot IP
2  ROBOT_IP="192.168.1.100"
3
4  # Custom local IP
5  LOCAL_IP="192.168.1.50"
6
7  # Custom interface (bypass detection)
8  NETWORK_INTERFACE="eth0"
```

### 2.6.2 Timeout Adjustments

Modify initialization wait times for slower systems:

```
1      # In start_robot_drivers()  
2      sleep 45  # Increase from 30 seconds  
3  
4      # In start_moveit_system()  
5      sleep 60  # Increase from 45 seconds
```

### 2.6.3 Debug Mode

Enable verbose output for troubleshooting:

```
1      # Add at beginning of script  
2      set -x  # Enable debug output  
3      set -v  # Enable verbose mode
```