

UI ROBOTIC WORKCELL

SETUP, OPERATION, AND PROGRAMMING OF THE ENGINEERING DEPT. ROBOTIC WORKCELL

This instructional manual includes information on the DENSO Robotic arm and the modular work-cell designed for it. Information discussed in this manual includes general information on safety, manual operation, basic programming, and advanced programming using WINCAPS III software.

UI ROBOTIC WORKCELL

SETUP, OPERATION, AND
PROGRAMMING OF THE ENGINEERING
DEPT. ROBOTIC WORKCELL

INTRODUCTION

Project History

In the spring of 2013 The Boeing Company donated four DENSO robotic arms to the College of Engineering. That fall Team Roboshow, learned programming and created a work-cell which incorporated multiple safety features as well as a clear poly-carbonate enclosure for public demonstration. The team programmed the robot to use a dry-erase marker to create logos and patterns on a white board.

The following year Team Vandalbot designed a manufacturing process which could assemble various rivet and nut-plate assemblies in order to automate a repetitive task currently done by factory workers. The team was able to validate this process and provide records for future teams.

Our Contribution

In 2016 Team Do-All Robotics was tasked to design a 2nd generation work-cell to improve the cell's manufacturing capabilities. Our team designed a work-cell with a standalone pedestal for each arm that eliminated settling time and vibration during operation.

We also created a customizable grid system for a work surface for a customizable workspace to suit various different manufacturing capabilities. We retained a clear polycarbonate enclosure for safety and demonstration purposes.

Our team has also worked towards adding a second arm to the workcell. By allowing two controllers to interact and running two simultaneous programs that start at the same time, the robots will not collide so long as the user has tested the programs in a virtual environment to ensure the robots do not collide.

THANK
YOU TO
PROJECT
SUPPORT
AND
SPONSORS

The Boeing Company

DENSO Robotics

Steve Beyerlein

Team Roboshow

Team Vandalbot

TABLE OF CONTENTS

| | |
|---|----|
| Introduction | 1 |
| Project History | 1 |
| Our Contribution | 1 |
| Table of Contents | 2 |
| DENSO VS-577G Specifications | 3 |
| Modular Work-cell design | 4 |
| Previous Design | 4 |
| New Modular Design | 4 |
| Workcell Safety | 5 |
| OSHA Robotic Safety Standards | 5 |
| e-Gard Interface | 6 |
| Using The Teach Pendant | 7 |
| Powering Up the Workcell | 7 |
| Teach Pendant Controls | 7 |
| Motion Mode | 9 |
| Speed Key | 9 |
| Programming With The Teach Pendant | 10 |
| Running a Stored Program | 10 |
| Writing/Editing a Program Using the Teach Pendant | 10 |
| Importing Coordinates from the Arm's Current Position | 11 |
| Robot I/O | 12 |
| Interpolation Control | 13 |
| Figures | 13 |
| Programming with WINCAPS III | 14 |
| Starting a New Project | 14 |
| Writing a Program | 15 |
| 3D Simulation | 15 |
| Importing 3D Data | 15 |
| 3D View Pane | 16 |
| Uploading Data from WINCAPS to the Robot | 17 |
| Appendix | 18 |
| Error Code Tables | 18 |
| PAC Library | 18 |

DENSO VS-577G SPECIFICATIONS

VS SERIES 650-850 MM

DENSO
robotics

The DENSO VS-Series six-axis articulated robots offer longer reaches than the VP Series and can handle higher payloads. They are also available in configurations that allow operation in specific environments. Outstanding features include:

- Fast cycle times from 0.59 to 0.49 sec.
- Excellent repeatability from $\pm 0.03\text{mm}$ to $\pm 0.02\text{mm}$
- Payloads up to 7 kg
- Reaches from 650 to 850 mm
- High maximum allowable moment of inertia
- Compact design saves floor space, facilitates integration
- Floor- or ceiling-mount, with no special hardware required
- Operation in dusty and wet environments (optional IP65 configuration)
- Operation in cleanrooms (optional cleanroom class-10 and class-100 configurations)
- ANSI and CE compliance allows global deployment
- UL models available for the US and Canada



VS-6556G



VS-6577G



CEILING-MOUNT OPTION



| | VS-6556G | VS-6577G |
|---|---|-------------------------|
| Number of axes | 6 | |
| Overall arm length (mm) | 565 | 770 |
| Second-arm width (mm) | 110 | |
| Reach (to center of 5th axis, mm) | 653 | 854 |
| Minimum motion area (mm) | 234 | 242 |
| Maximum payload (kg) | 6 (7 with flange pointing down) | |
| Weight, approx. (kg) | 35 | 36 |
| Footprint (mm × mm) | 200 × 200 | |
| Cycle time (sec.) | 0.49 (with 1-kg weight) | 0.59 (with 1-kg weight) |
| Maximum composite speed (mm/sec.) | 8,200 | 7,600 |
| Maximum joint speed (°/sec.) | J1: 375 J2: 300 J3: 375 J4: 410 J5: 410 J6: 660 | |
| Repeatability ^{1,2} (mm) | ± 0.02 | |
| Maximum allowable moment of inertia (kgm ²) | J4, J5: 0.413 J6: 0.063 | |
| Range of motion (°) | J1 ± 170 J2 +135, -100 J3 +166, -119 J4 ± 190 J5 ± 120 J6 ± 360 | |
| Position detection | Absolute encoder | |
| Drive motors and brakes | AC servomotors for all axes; brakes on 2nd, 3rd and 4th axis (5th and 6th axis optional) | |
| Air piping (in-arm integrated) | 10 lines (max. 1 A per line) | |
| Wiring (in-arm integrated) | 7 systems ($\phi 4 \times 6$, $\phi 6 \times 1$), three 2-position double solenoid valves | |
| Protection class | Standard: IP40 Options: IP54/65 (body/wrist), IP65 (dust- and mistproof), cleanroom class 10 ($0.1 \mu\text{m}$) ³ , cleanroom class 100 ($0.3 \mu\text{m}$) ³ | |
| Ceiling-mount ³ | ✓ | |
| UL specifications ⁴ | ✓ | |



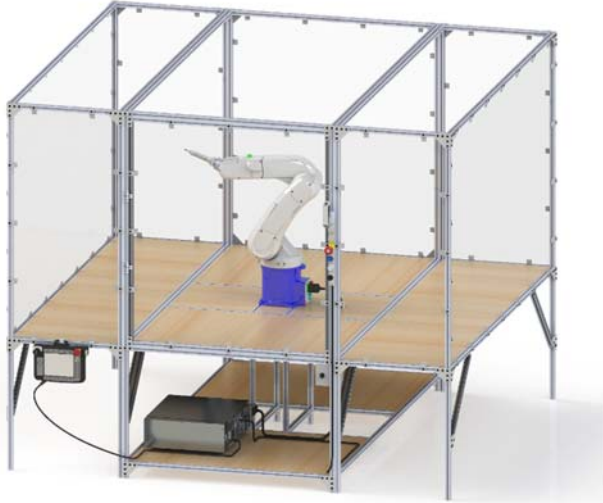
Scan this QR Code®
for more information

1. At constant ambient temperature. 2. In all directions. 3. All except cleanroom models, which can be floor-mounted only. 4. Only for dust- and mistproof model with brakes.

www.densorobotics.com

MODULAR WORK-CELL DESIGN

Our team decided to update the existing work-cell to a newer, more modular configuration. This will allow future users to easily arrange the work-cell into a configuration that allows for many different manufacturing capabilities.



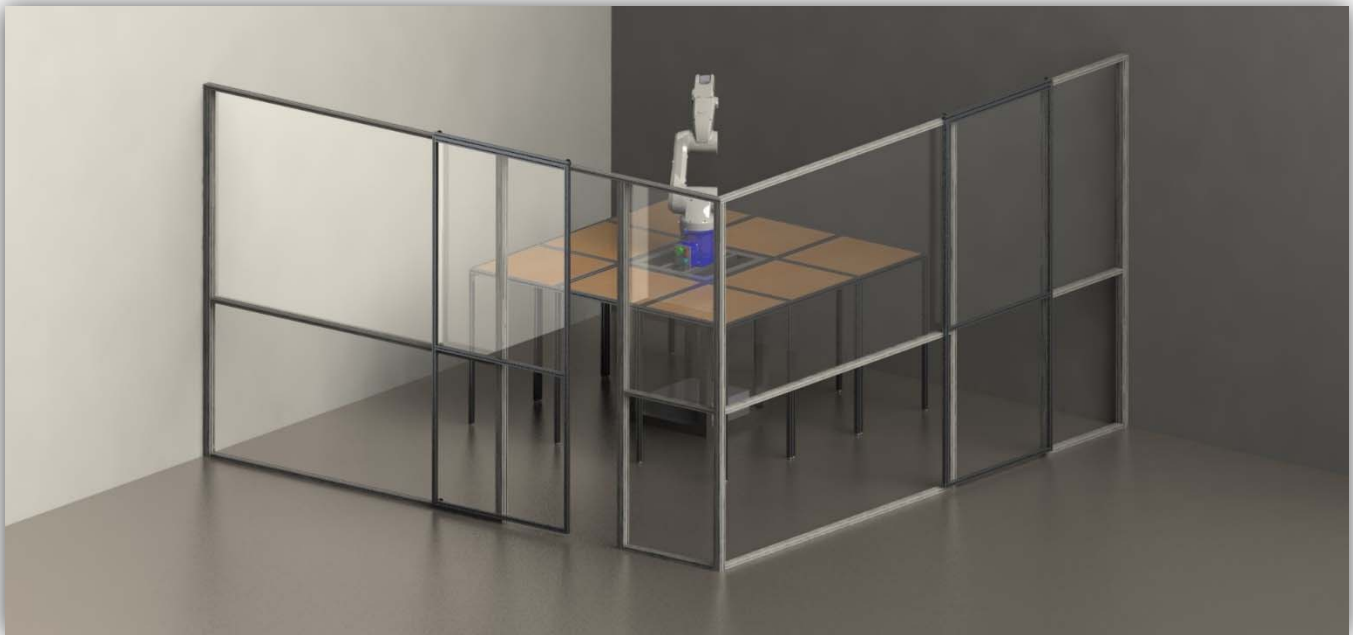
TEAM ROBOSHOW PREVIOUS WORKCELL DESIGN

Previous Design

In 2013 – 2014 Team Roboshow designed and fabricated a work-cell for one DENSO VS-G arm shown in Figure 1.1. Although this cell worked well for demonstration purposes two of its limitations were a slow settling time (being “wobbly”) and being difficult to reconfigure for different projects or worker entry.

New Modular Design

Our new design was created by building a standalone pedestal for each robot and then attaching work “tables” that can be arranged in different patterns. The configuration shown below is a 3x3 square that allows a worker to walk around the workspace in the manufacturing cell.



RENDER OF NEW WORKCELL DESIGN WITH ONE ROBOT AND A 3X3 WORKSPACE

WORKCELL SAFETY

OSHA Robotic Safety Standards

Summary of Occupational Safety and Health Administration Safety Standards related to robotics and safety enclosures. The below list includes all standards relevant to the UI Robotic Work-cell.

1910.144

Color identification

- Red- Danger, fire protection, emergency stop
- Yellow- designating caution and marking physical hazards

1910.145

- Accident prevention signs and tags
 - Caution signs: warn against unsafe practices
 - Safety instruction signs
 - General operation instructions
 - Safety measures

1910.212

- Machine Guarding Methods
 - Barriers
 - Two-handed tripping devices
 - Electronic safety devices
- Workspace guarding (protecting the area of possible operation)
- Guarding device shall prevent operator from having any body part in the danger zone during operating cycle

PUB 8-1.3

- Protection Practices:
 - Interlocked Barrier: physical barrier around robot incorporating interlock gates
 - Fixed Barrier: fence that requires tools
 - Awareness barrier: railing or suspended chain
 - Presence sensing device: pressure mats, and light curtains
- Emergency Braking:
 - Dynamic braking systems cutting off all power
- Visible warning systems
 - Easily recognizable and enhance effectiveness of safeguards
- Control Devices:
 - Main Control panel is located outside robot work area and in sight of robot
 - Easy accessible emergency stops
- Controls that initiate power or motion are constructed and guarded against accidental operation
- All devices are clearly marked and labeled to purpose

e-Gard Interface

This device acts as an alternative interface to the controller along with providing extra safety precautions. The e-Gard system is connected to both the mini and safety I/O allowing for slight monitoring and external control of the robot.

Solenoid Lock

The features at the top of the e-Gard act as a door lock. It is equipped with a solenoid locking system for the front of the enclosure along with a built in safety relay, which detects when the door is open, instantly cutting power to the robot.

E-Stop

The emergency stop has a key reset adding extra security. This ensures that when non-authorized users try getting into the enclosure they cannot do so without a key.

Start/Re-Start

This feature is connected to the program start input of the mini I/O. With the first touch of the button the program is re-started and the second push starts the program.

Step Stop

When the stop button is pushed the robot will continue to the next coded in position allowing for a gentle stop. This button is also illuminated when power is being supplied to the robot. This is done by connecting the robot running indicator off of the mini I/O.

Selector Switch

This gives the ability to choose between two different programs pre-loaded on the controller. This is connected to data bit 0. The selector switch will be able to select between two programs determined on the controller by their designated name. To operate a program using the selector switch, the program file names must be pro0 and pro1. Only these file names will work with the selector switch.

E-GARD LOCKOUT

Deadman Switch

The dead man switch is needed to control the robot in manual mode. It must be held down in order for the robot to move during manual operation. Release the switch to stop the robot, or press harder to stop it. The switch has 3 states. Released and strongly pressed are off. Lightly pressed is the only state where it is on.

Motor Key

Each time the robot is powered up or the mode of operation is changed this switch must be pressed to manually turn on the motor. If the Deadman switch is released or if the emergency stop is activated, the motor will shut itself off and need to be restarted.

Mode Selector Switch

This switch changes the mode of operation for the robot, there are three different settings:

Auto – Allows the robot to run programs saved in the Teach Pendant or WINCAPS (the Enable Auto Switch on the enclosure must be “on” for the robot to run in this mode)

Manual – Allows the operator to control the robot manually using the Arm Transverse Keys (the Enable Auto Switch on the enclosure must be “off” for the robot to run in this mode)

Teach/Check – Allows the operator to manually run through programs step-by-step or at a slower speed, the Deadman switch must be used to operate in this mode (the Enable Auto Switch on the enclosure must be “off” for the robot to run in this mode).



ENABLE-AUTO SWITCH

Arm Transverse Keys

These are the keys the operator uses to manually operate the robot. There are three different motion modes but most often used is the “Joint” mode where each key performs the following operation:

J1 – Rotates the robot on its base

J4 – Rotates the wrist of the robot arm

J2 – Moves the shoulder of the robot arm

J5 – Moves the head the tool face is attached to

J3 – Moves the elbow of the robot arm

J6 – Rotates the tool face or end effector

MOTION MODE

There are three motion modes the robot is able to be operated in during manual mode: Joint mode, X-Y mode, and tool mode. The operation in Joint mode is shown above. Operation in X-Y mode and Tool mode assigns the buttons the following operations but with different origins:

J1 – Moves parallel to X-axis

J2 – Moves parallel to Y-axis

J3 – Moves parallel to Z-axis

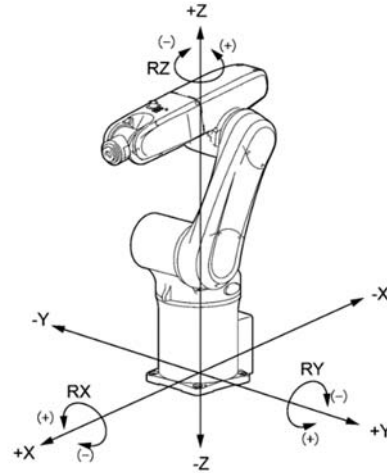
J4 – Rotates with X-axis as rotation center

J5 – Rotates with Y-axis as rotation center

J6 – Rotates with Z-axis as rotation center

When the robot is in X-Y mode the center of the bottom of the base is used as the origin

When the robot is in Tool mode the center of the flange surface on the tool face is used as the origin.



X-Y MOTION MODE COORDINATE SYSTEM

SPEED KEY

The speed key allows the user to set the speed at which the robot runs either during manual operation or while running a selected program.

In Auto mode, the robot will run at the (programmed speed x reduced ratio (%)). If set to 80%, the robot will run at 80% of the programmed speed.

In Manual mode or Teach check mode, the robot controller is designed to run the robot at 10% of the programmed speed. If set to 80% speed, therefore, the robot will run at 10% x 80% of the programmed speed, that is, at 8% ($0.1 \times 0.8 = 0.08$).

At the start, set the speed to 20% or less. By running the robot manually at high speeds from the beginning, it may mistakenly strike against the surrounding objects.

PROGRAMMING WITH THE TEACH PENDANT

Running a Stored Program

In order to run a program that is stored on the Teach Pendant, first the program needs to be enabled.

1. Switch the Teach Pendant to Manual mode.
2. Touch the "Pgrm" button on the bottom of the Pendant screen and scroll to the desired program to ensure it is Enabled.
3. If it is listed as Disabled, touch "Config" at the bottom of the screen.
4. Select "Make Specified Program Active".

An error message (73F5 "Local variable initialized") will pop up. This error is not an issue; it simply halts any currently running program.)

5. Once it is confirmed the program is initialized switch the Teach Pendant to Auto mode.
6. Touch the "Pgrm" button on the bottom of the screen and select the program to be run.
7. The program can be run via Start or Step-Start.
 - a. Start allows the program to be run in its entirety from start to finish.
 - b. Step Start allows the operator to run the program line-by-line manually.

Writing/Editing a Program Using the Teach Pendant

The Teach Pendant has the capability to create new and edit existing programs. It is recommended that for writing a new program WINCAPS is used on a PC, this is much easier than using the Pendant's on-screen keyboard. Below is an example program, and on the following page is a table of the syntax used.

```
0001 '!Title "Test1"
0002 PROGRAM TEST1
0003 MOTOR ON
0004 TAKEARM
0005
0006 'LIST OF COORDINATES
0007 j00 = (0.00, -6.53, 155.37, 0.00, -60.54, 0.00)
0008 CENTER
0009 P1 = (500, 300, 300, 0.00, 90, 0.00)
0010 'OUTSIDE
0011 P2 = (500, 300, 500, 0.00, 90, 0.00)
0012 P3 = (500, 100, 300, 0.00, 90, 0.00)
0013 P4 = (500, 300, 100, 0.00, 90, 0.00)
0014 P5 = (500, 500, 300, 0.00, 90, 0.00)
0015 'MOVE COMMANDS
0016 move p, @E j00, S=100
0017 'DELAY 2000
0018 move p, @E P1, S=100
0019 move p, @E P2, S=100
0020 move c, P5, @P p4, S=50
0021 move c, P3, @E p2, S=50
0022
0023 GIVEARM
0024 END
```

| Command | Description |
|----------------------------------|---|
| PROGRAM | Declares the program name to be run, this must match the file name |
| MOTOR ON | Turns on the robot's motor so the operator does not have to do this manually |
| TAKEARM | This command specifies the program is taking control of the arm |
| ` | This symbol at the beginning of a line turns it into a comment |
| j00 | Joint equivalent for a point. Sets the exact position of the arm based on joint positioning, useful for setting home position of robot before starting movement |
| p# = (x,y,x,Rx,Ry,Rz) | Sets coordinate positions as a variable to be used later in the program, these are formatted in the same style as X-Y and Tool Motion Modes. When setting these coordinates, the arm will position the center of the flange surface to the specified location |
| move p | Moves the robot to the specified coordinate variable in a P2P fashion |
| move c | Moves the robot in an arc with the first variable as the arc center-point and the second variable as the destination |
| @E | Specifies the movement as an encoder value check motion. The robot will pause and confirm its location before continuing on with the program |
| @P | Specifies a pass-through point. The robot will continue through this point to the next command without stopping at all |
| DELAY ##### | Tells the robot to pause for a specified amount of time in milliseconds before proceeding to the next command |
| GIVEARM | This usually goes at the end of the program and gives control of the arm back to the operator or allows another program to take control |

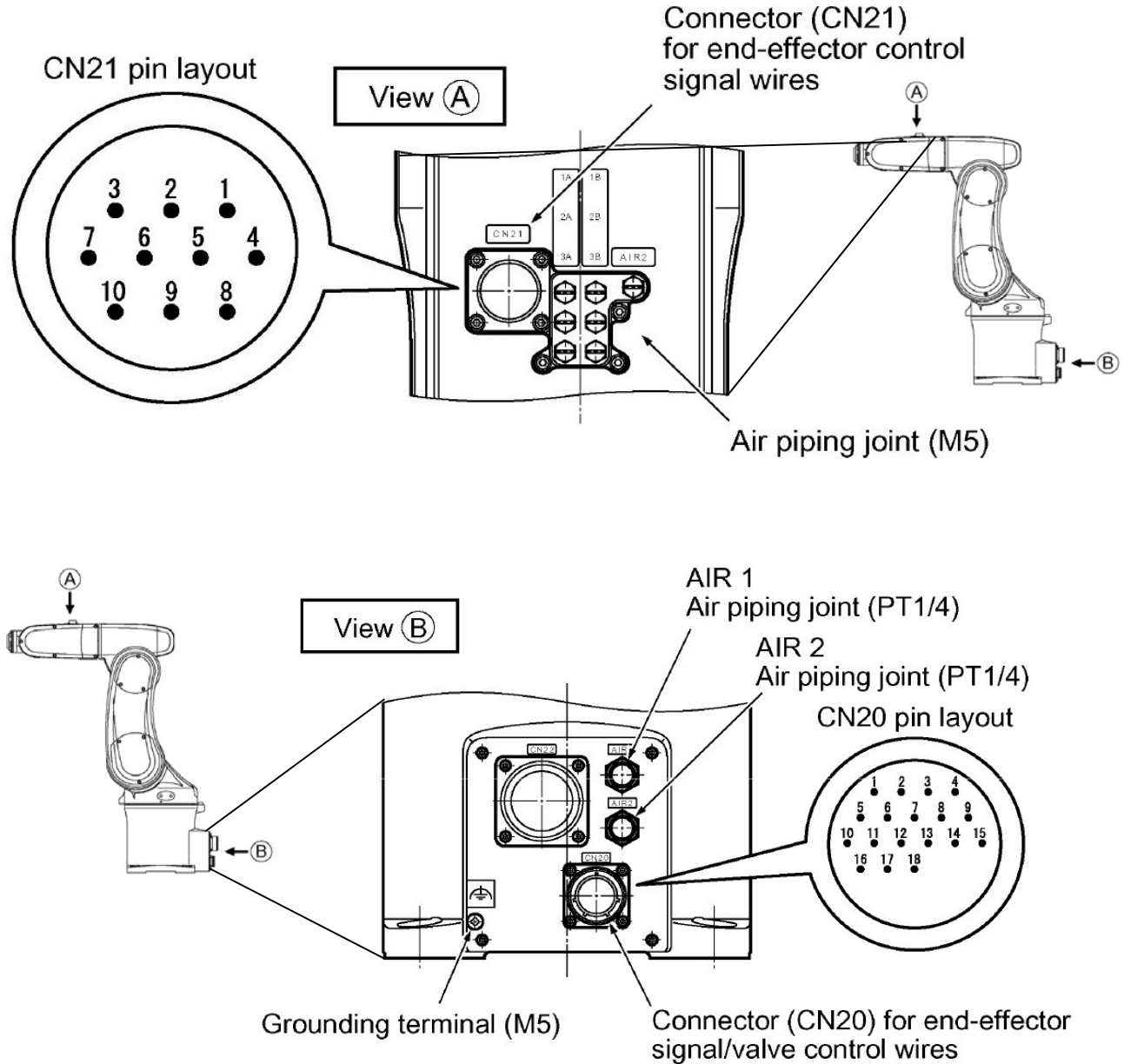
IMPORTING COORDINATES FROM THE ARM'S CURRENT POSITION

Sometimes when creating a process, it is useful to move the arm manually, then use the current position in a program. This is a simple process using the Teach Pendant:

1. On the home screen of the Pendant select the Arm button at the bottom (F2)
2. Select the Var. button (F4)
3. Select the POS button (F4)
4. Select Get Pos. (F6) and the arm's coordinates will be imported into the selected variable. This will not update variables explicitly specified in existing programs.

ROBOT I/O

The robot is equipped with 2 air inputs, 10 signal lines, and 3 solenoid valves. One of the air inputs feeds the three solenoid valves and the other one is pass-through. The signal lines are able to electronically operate an end effector in the robot via logic programming. Below are two diagrams that show the location and layout of the robot's air routing and signal lines.



Setting I/O in WINCAPS is simple. There are two ways to program I/O. The first is using semaphores with the TAKESEM command or directly setting an I/O pin using the SET IO command:

SET IO(###)

RESET IO(###)

SET IO will signal a port on and RESET IO will signal a port off. It is also possible to read an input from an I/O port

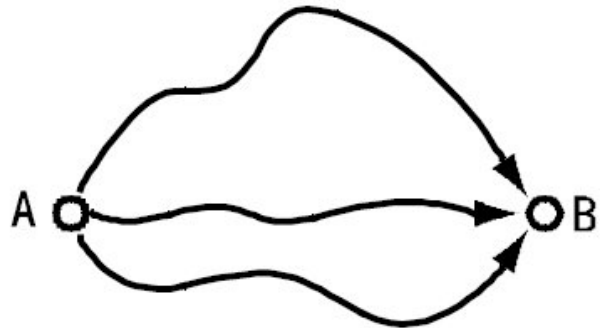
INTERPOLATION CONTROL

There are three different methods of movement that can be specified when telling the robot to move between two different points. Two of these were referenced in the example program that was run on the teach pendant (move p & move c). Below is a description of each movement type.

PTP: Point to Point

This can be defined as the movement from one point to another point. The path on which the robot moves depends on the robot posture and is not always a straight line.

To designate this path type use “move P” before the variable. It is suggested to use this as the first motion command in a program as the other paths may not be available due to the robot’s position.



CP: Linear Path

This control manages interpolation so that the path to reach the motion destination position will be a straight line.

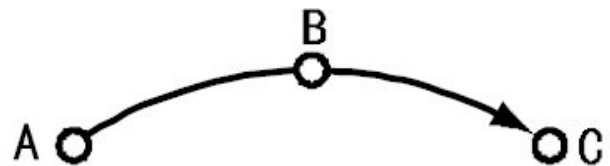
Use “move L” for this interpolation method. If an error occurs, visually check to see if it is possible for the robot to move between the two points.



ARC: Circular Path

This control manages interpolation so that the path to the motion destination position will be an arc.

By designating “move C” the robot executes the ARC motion. The robot will use the first variable as the center-point of the arc, and the second variable as the destination point.



Figures

A 6 axis robot has many available figures to reach one point. It can take a path to the right or left, above or below to get to the designated point in the work space. Figures can be used to influence the method in which the robot moves from one point to the other and how it behaves during a program.

WINCAPS has 32 predefined figures to select from to use for a point in space:

- The shoulder (axes 1, 2, 3) of the robot can be designated **lefty** or **righty**.
- The elbow (axes 2 & 3) of the robot can be designated **above** or **below**.
- The wrist (axes 4 & 5) of the robot can be designated **non-flip** or **flip**.
- The tooling end (axis 6) of the robot can be designated **single** or **double**
- The wrist (axis 4) of the robot can also be designated **single** or **double**

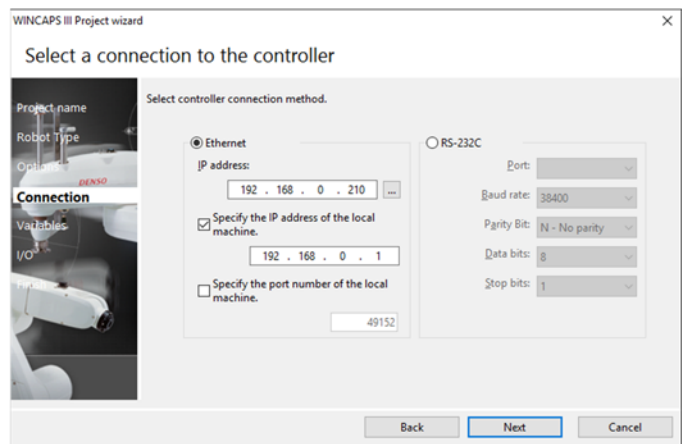
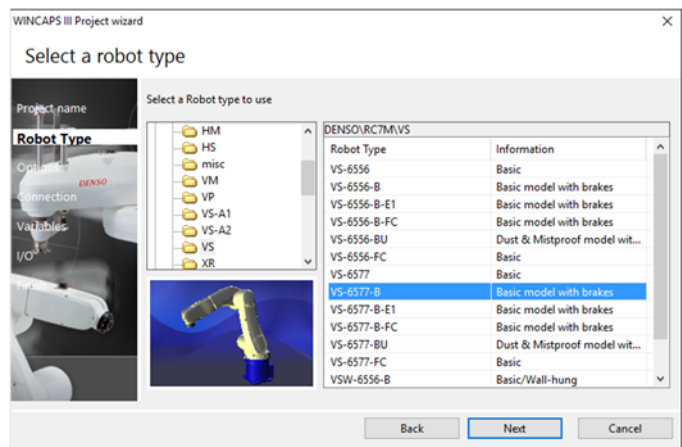
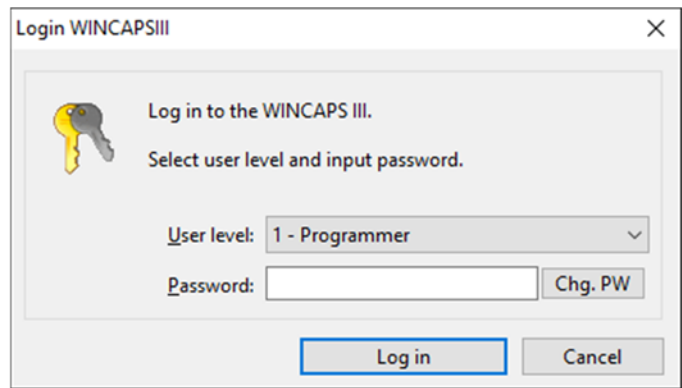
PROGRAMMING WITH WINCAPS III

WINCAPS III is a powerful program used to do offline programming for DENSO Robotics. WINCAPS is capable of creating, editing, and simulating any program that can be performed by many of DENSO's different robots. The program also offers extensive help and has built in tasks, such as pallet stacking, which can be tailored to the user's specifications.

Starting a New Project

The robotic workcell has a laptop computer with WINCAPS III already installed. This computer has been preconfigured with the proper network setting to communicate with the robotic workcell. Start by opening WINCAPS III on the computer.

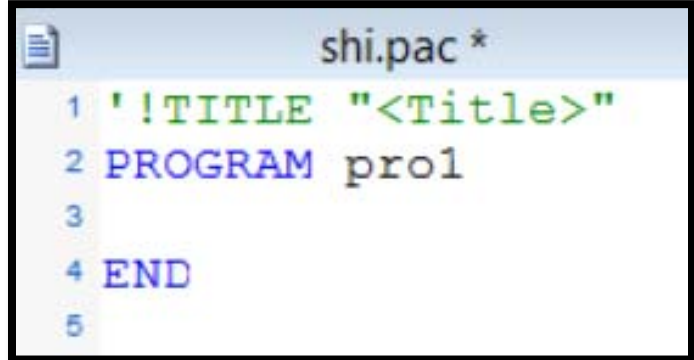
1. Login to WINCAPS as a programmer, use the password "denso". Go to File>New Project
2. The New Project Wizard will appear, select the robot type manually.
3. Project Name – After this there will be a prompt to save the project, do not save programs to the desktop as this will cause the computer to run slowly
4. Robot Type – Next the computer will ask to specify the robot type, use VS-6577-B.
5. Options – The controller specification is "ANSI (Global)", select extra joints as NOT USE. Output code is Ver2.80.
6. Connection – Each controller has a local IP address specified on the side of it. Use the Ethernet option and type this IP into the IP address bar. Make sure the box to "Specify the IP address of the local machine" is checked and is populated with 192 . 168 . 0 . 1.
7. Variables – Do not make any changes to Set Variable Size, unless certain more are needed.
8. I/O – Do not make any changes. The device used is "Mini I/O" and the mode is "mini I/O".
9. Finish – Review the project specifications and click Finish.



Writing a Program


After creating a project in WINCAPS a new program can be added. Do this by using the drop down menu under "Project" on the menu bar and select "Add Program". A "Create New Program Prompt" will appear. The robot operates using *.PAC files, so choose to save the program in this format. It is best practice to give the program the same name as the project to prevent confusion later.

After the program is created the program will have a commented title, the program name declaration, and an end statement. Remember to use the TAKEARM command before writing anything else in the program. It is also useful to use a MOTOR on command so this does not need to be done manually. Writing a program using WINCAPS can be done exactly how it is done using the Pendant.



```
shi.pac *
1 !TITLE "<Title>"
2 PROGRAM pro1
3
4 END
5
```

3D Simulation

WINCAPS has the capability of 3D simulation within the program. This can be used to teach the robot new points and to check existing programs for any possible issues before testing them in the actual workcell. In order to use 3D simulation a project must be active, to engage the robot arm view click the arm view icon  in the command bar.

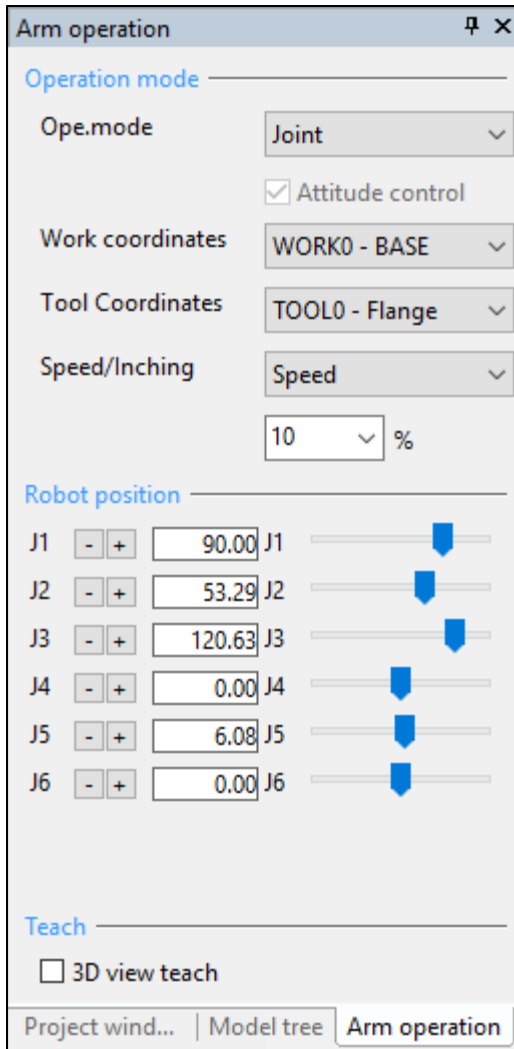
In the 3D view use the left mouse button to rotate the screen, the scroll wheel or the right mouse button to zoom and the center button to pan.

IMPORTING 3D DATA

Although WINCAPS can model simple objects within its 3D simulation window, sometimes it may be useful to include parts with more complex geometry. The best way to do this is by using SolidWorks parts.

Open the SolidWorks file and select which face will be used as the front of the object. When prompted to save the file select the VMRL format (*.wrl) with Version: 97 and whichever unit system will work best for the specified project. Select a node to insert the image into WINCAPS and import the 3D data, then scale accordingly.

3D VIEW PANE



Arm Modelling

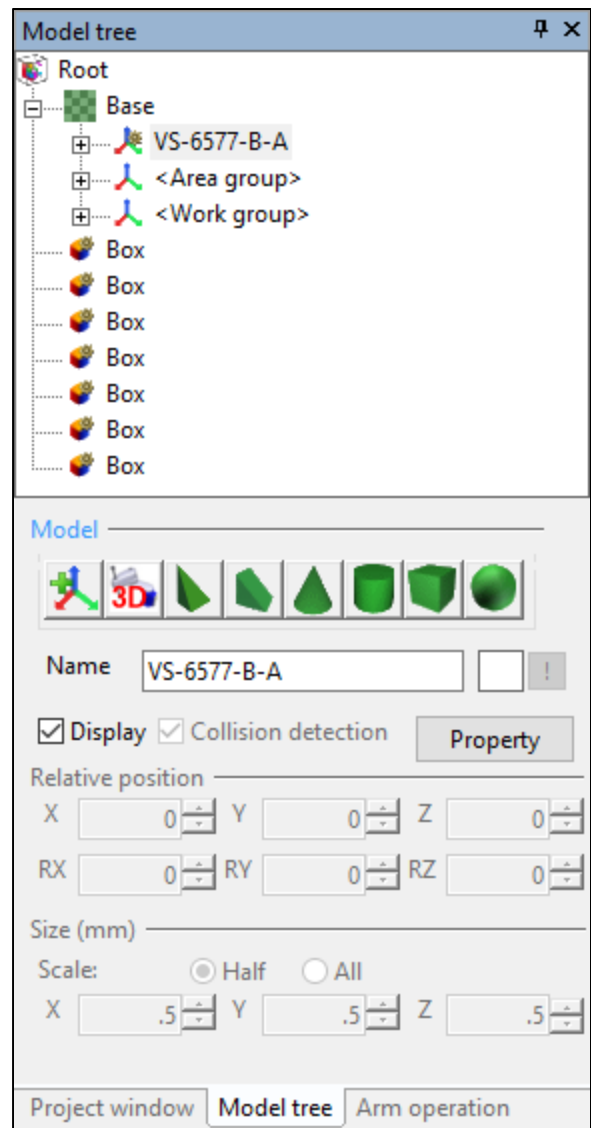
The Model Tree is for adding tools, work pieces, obstacles, and other objects to the equipment layout in the Arm 3D view window. This window allows creation of basic figures and objects in the 3D view window to be used during simulation.

It is possible to add and remove objects from the simulation as well as change their colors. "Collision detection" can also be used to check the program for any possible problems so they can be resolved before uploading data to the robot.

Arm Operation

The 3D view pane includes a project window with tabs for operating the simulated arm and populating the workspace with 3D parts.

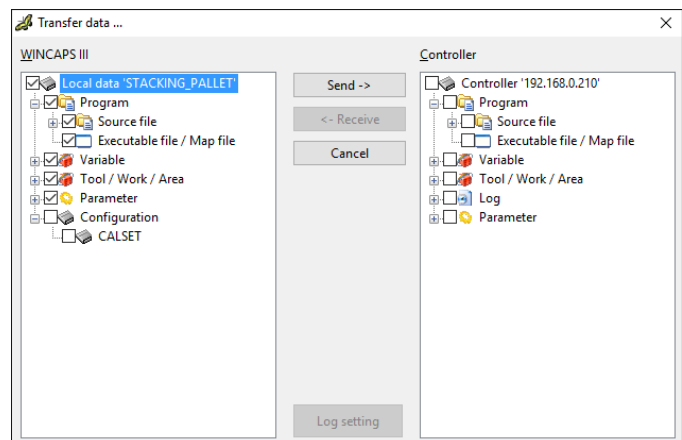
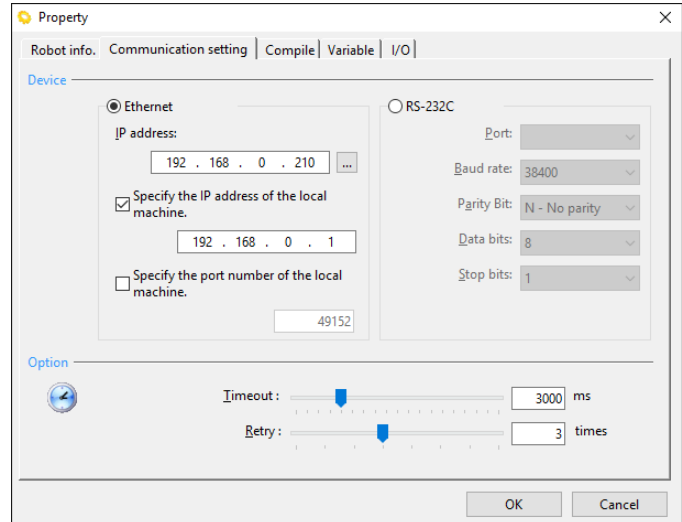
Using the arm operation window is similar to programming with the Teach Pendant, the same motion modes are used with the same coordinate systems. In this mode it is possible to select either work coordinates or tool coordinates for programming the robot. "3D view teach" teaches the posture symmetry to the surface when clicking an object within the Arm 3D view window.



Uploading Data from WINCAPS to the Robot

To send data from WINCAPS to the RC7M controller (the robot's brain), only the Toshiba laptop that comes with the workcell should be used due its specific network configuration settings. Below are the steps to transfer a project from the controller to the robot.

1. Ensure the computer and the controller are both ON with the robot in Manual and the motor OFF.
2. Ensure the computer is disconnected from any wireless networks.
3. Connect the Ethernet cable to the controller and the other end to the computer via the USB converter, it does not matter which port is used
4. Open WINCAPS if it is not already open and check the Connect Settings located under "Connect" in the toolbar to make sure the IP address of the controller is set properly and the IP address of the local machine is specified as 192.168.0.1
5. Once the Connect Settings have been verified select Transfer data (ctrl+t), also located under "Connect"
6. In the Transfer data window select the very top checkbox on the left side and press Send. Accept any dialogue boxes and wait for the computer to send the project to the controller
7. Verify the project transferred successfully.



Once this process is complete the project will be available via the Teach Pendant in the same location as any other previously stored projects or ones that were created using the Teach Pendant.

To copy any projects from the controller to the computer work up to step 6 then, on the right side of the Transfer data window, select the desired project to transfer under the "Source File" folder.

APPENDIX

Error Code Tables

PAC Library