

Continue



Fanuc robot programming guide

Given text rewritten as follows: Update: 17 October, 2023 This manual tells us how to use RoboDK to make new programs for Fanuc robots and load them onto the robot. RoboDK works with all types of Fanuc robot controllers since RJ2, including RJ3, R-30iA and R-30iB. This manual is based on the R-30iA Fanuc controller. File format: PDF Size: - MD5 Checksum: 9AEF27750FE2D44B77B68E32934FC2B Publication date: 17 October, 2023 Downloads: - PDF Link: Fanuc robot programming manual PDF This TechShort explains how to expand the number of different Program Setup Items including UTools, UFrames, Digital I/O, Payload Schedules, Data Registers, Position Registers, Vision Registers, and more- without having to buy any extra software options. Robotic cells are complete systems that house your robot, controller, and other important things like positioners. Your robot cell also makes sure the safety of your employees and protects your investment. Efficiency, precision, and reliability are very important. So, before you start integrating, think about the following five tips for programming your FANUC integrated work cell to get the most out of it. Take some time to think carefully about the programming structure Your programming structure will be with you throughout the whole programming process of different applications of the cell. Spending some time setting it up correctly before starting programming could save you a lot of time and trouble later. Making a pseudo-code structure can help you think about the logical flow of events and give you a framework when writing the actual code. Drawing a flowchart or process diagram can also be very useful for visualizing the routine your robot will do. Save yourself some time and debugging by sketching out your programming structure in advance and making sure all workers know the routine. Set the robot's working tools correctly using "Tool Frames" FANUC's "Tool Frames" allow programmers to plot the robot's movement in a Cartesian coordinate system. The reference point, also called the Tool Center Point (TCP), is set by one of several methods, and all other positions are referenced relative to the TCP. The operator defines the TCP using one of four common methods : the three-point method, six-point method, direct entry, or two-point + Z method. With three-point and six-point methods, the TCP is defined by setting the robot tool to different reference positions. The direct entry method allows the programmer to enter the TCP coordinates manually. The 2-point + Z method uses two reference points to determine x and y coordinates but lets the user define the height of the tool. Be sure to enter the weight and inertia characteristics of the robot hand because this data plays a role in correctly determining the robot path and can speed up the cycle time when input correctly. Do not try to "patch" a programming problem When a problem arises in programming, a patch may provide a workaround and often appeals to users as a convenient and low-maintenance fix. However, patches may not address the underlying causes of errors that can persist and manifest themselves in other areas. Before applying a patch, check to see if you can identify the source of the error. Double-check To avoid errors and streamline your workflow, consider the following tips: Planning ahead is crucial. Before purchasing a robot, think carefully about the processes you want to automate. Research different models and consult with your team to choose the best fit for your automation needs. Leverage existing libraries and free resources to simplify programming tasks. The DIY-Robotics community offers valuable support through free consultations, forums, and training resources like GitHub and FANUC's official platforms. By understanding physics and engineering principles that govern robots, you can find universal solutions to common problems. Don't overlook programming opportunities in different languages or for other products. Additionally, the DIY-Robotics guide provides a comprehensive resource for executing your project effectively with Fanuc robots, covering topics such as Teach Pendant programming, motion commands, registers, logic structures, and more. The robot can be controlled using various commands, including motion commands such as Joint Move (J), Linear Move (L), and Circular Move (C). Joint Move allows the robot to move in joint mode at a faster but less precise rate. Linear Move enables the robot to move in a straight line with options for continuous motion and blending. Circular Move allows the robot to follow a circular path between two points. The robot also utilizes registers, including Registers (R) for storing numerical values and Position Registers (PR) for storing coordinates that can be modified during runtime. These registers enable calculations, loops, and conditional statements within programs. Logic and control structures are essential for programming the robot. Conditional execution allows the program to jump to specific labels based on conditions such as digital input states. Loops enable repetitive tasks by incrementing counters and jumping back to labels until a condition is met. Input/Output (I/O) and signal handling are crucial for interacting with external devices. The robot can wait for digital inputs, set digital outputs, and handle signals to control devices like grippers. Example programs demonstrate palletizing and pick-and-place tasks. These examples illustrate how to move the robot to specific locations, activate and deactivate grippers, and repeat tasks until a condition is met. Error handling and debugging are vital components of programming. The robot can check for part presence, signal error conditions, and wait for specific times to handle errors effectively. Advanced functions include using macros for repetitive tasks and timers for waiting specific periods. Best practices recommend utilizing CNT blending for smooth transitions, Position Registers for dynamic locations, and optimizing motion paths to reduce cycle time. To load a program from a USB drive into the Fanuc robot controller, follow these steps: plug in the USB drive, select Menu→File, choose Utilities→Set device→USB, select DIR, and pick the program file (e.g., BALLBARTEST.TP). The file will be saved to the FR memory of the controller. Then, select the program in the FR memory and click LOAD. The documentation covers typical operations using a Fanuc robot with RoboDK and transferring programs to the robot. It supports all Fanuc robot controllers since RJ2, including RJ3, R-30iA, and R-30iB, with this specific guide based on the R-30iA Fanuc controller. Given text here The ability to load Linear Safety (LS) files directly onto the controller enables streamlined operation. Additionally, LS programs can be compiled into Fanuc Tool Program (TP) files using WinOLPC tools. It's worth noting that RoboDK-generated programs' linear speeds are set by default in register R[10]. If not specified, setting speed manually is necessary. To start a robot program on the controller: 1. Select FCN and abort all. 2. Select SELECT and the program (e.g., BALLBARTEST). 3. Press and hold the Deadman switch. 4. Select RESET to clear alarms. 5. Hold the SHIFT button and select FWD to initiate the program. The teach pendant allows retrieving real robot joints by: 1. Selecting POSN or JNT on the screen. 2. Adjusting tool parameters as needed. Robot tools are stored in registers (UTOOL) and can be created or modified using these steps: 1. Select MENU, Setup, Frames. 2. Choose a tool and modify X,Y,Z,W,P,R values. Note that RoboDK-generated programs may set up tools according to the simulation settings. Fanuc's FTP server is enabled by default on recent controllers, allowing for secure file transfers. To connect via FTP: 1. Select MENU, Host Comm. 2. Use an anonymous username if available. Establishing a connection between RoboDK and the Fanuc robot controller enables online programming and debugging using the Run on Robot option. Robot drivers provide an alternative to offline programming and allow running simulations directly on the robot (Online Programming). To set up the RoboDK driver for Fanuc, follow these steps: 1. Download the Fanuc driver program files from the customer area and transfer the required robot programs to the Fanuc robot controller. 2. On the robot teach pendant: * Select Menu-(Next)-System-[TYPE]-Variables * Set \$HOSTS_CFG[3]\$_SERVER_PORT = 2000 * Set Protocol Name to SMB * Set Port to P3 (or the corresponding port) * Set Inactivity Timeout to 9999 * Set Startup state to [CHOICE] START * Start the program RDK_S3 3. In RoboDK, provide the IP of the robot and select Connect to connect to the robot. 4. If the communication fails or the robot enters error mode, restart the program RDK_S3 and reconnect from RoboDK. Note: Ensure that the port specified in section 1.c is used consistently across both RoboDK and Fanuc, as the default port 2000 may need to be adjusted for specific configurations. Settings File Configuration for RoboDK and Fanuc Robots A settings file can be found at the beginning of the driver's log on the line that starts with Loading settings. To customize configurations, make a copy of the driver with a unique name. Open this file with any text editor to find a list of axes used by RoboDK and corresponding axes on the Fanuc robot controller side. You can enable extra debug printing and change the number of motion groups used. For example, if axis 8 in RoboDK corresponds to axis 9 on the robot controller, simply change the number on the right-hand side to the correct value, as seen in the startup message below the Loading settings message. The configuration explains the difference between Fanuc LS and TP program files and how to automatically compile programs for Fanuc robots using RoboDK. Fanuc robots require a binary file to run robot programs. Programs generated by offline programming software are in LS format (ASCII, text-readable), while TP formats are compiled. There are two options to convert LS robot programs to TP programs:1.Having the ASCII Upload software option on the robot;2.Compiling the program using the WinOLPC tools from Roboguide. To compile a program automatically, use the "maketp.exe" compiler tool from Fanuc. This requires a <robot.ini> file, which can be generated with the setrobot.exe utility from WinOLPC if Roboguide was installed and a workstation was saved on the computer ("Work Cell"). 1.Select Program→Add/Edit post processor 2.Select Fanuc RJ34 (or other required post processor). Check if you are using compiled post processors. 3.Locate variable PATH_MAKE_TP, enter path of maketp executable. Post Processor for Fanuc RJ34 needs to be updated after calibration with RoboDK's Calibration and Performance package. This is due to changes in kinematic parameters (DH table). It may impact programs manually created on the robot controller. Keeping original DH parameters will allow reverting back to factory settings. Steps: 1.Open Calibration window, open Calibration parameters 2.Select Geometric simplified 3.Run calibration sequence or Update existing one 4.Retrieve calibrated DH parameters from Robot panel and select Export Table 5.Use Accurate kinematics (nominal is default) 6.Use Danavit Hartenberg Modified model in System variables to modify controller parameters. 7.Modify DH values with a and d values from Export Robot Kinematics window.