HD HYUNDAI ROBOTICS

Warning

THE INSTALLATION SHALL BE MADE BY QUALIFIED INSTALLATION PERSONNEL AND SHOULD CONFORM TO ALL NATIONAL AND LOCAL CODES

Hi5a Controller Function Manual

Noncontact Tool Calibration Function

HD HYUNDAI ROBOTICS

Printed in Korea –Jun, 2023, 2nd Edition

HD HYUNDAI ROBOTICS

Table of Contents

Table of Figures

1
Overview

1. Overview

This manual describes the noncontact tool calibration function that uses a LEONI sensor.

◆ **【Essential manual】** ◆

- Hi5a Controller Operation Manual

◆ **【For more information】** ◆

- This function manual is written based on the Hi5a V40.17-00 version.
- The tool calibration is an optional function. Execute the function after inputting the "Tool Data Automatic Calibration (ATDC)" license number.
- To use the built-in DeviceNet for communication with the sensor, a "built-in Fieldbus (DeviceNet)" license is required.

1.1. Main Specifications

1.1.1. Introduction of Tool Center Point (TCP)

In the robot, the distance and angle of the TCP should be set based on the position of the R1-axis flange to perform work using a tool. TCP is a reference point that the robot refers to for operation, and it is required to control each axis to designate a position and direction of the point as desired. Therefore, correct operation cannot be guaranteed if the TCP of the tool that is to be used is not set or incorrect.

In an industrial robot, the calibration is performed according to the following methods to designate the position of the TCP.

1. Prepare a sharp tip, as shown in the figure below, and fix it to a location outside.
2. Position the pointed part of the tool (in general, the end of the wire that protrudes by a preset amount from the end of the torch) onto the externally fixed tip, and record the position.
3. Repeat the recording of the position while changing only the direction of the tool and keeping the pointed part onto the externally fixed tip.
4. If you find both the axis origin and the tool length, record 7 points. If you find only the tool length, record 4 points or more.
5. Record the position of the TCP by performing the “axis origin and tool length optimization” function among the automatic calibration functions of the robot.

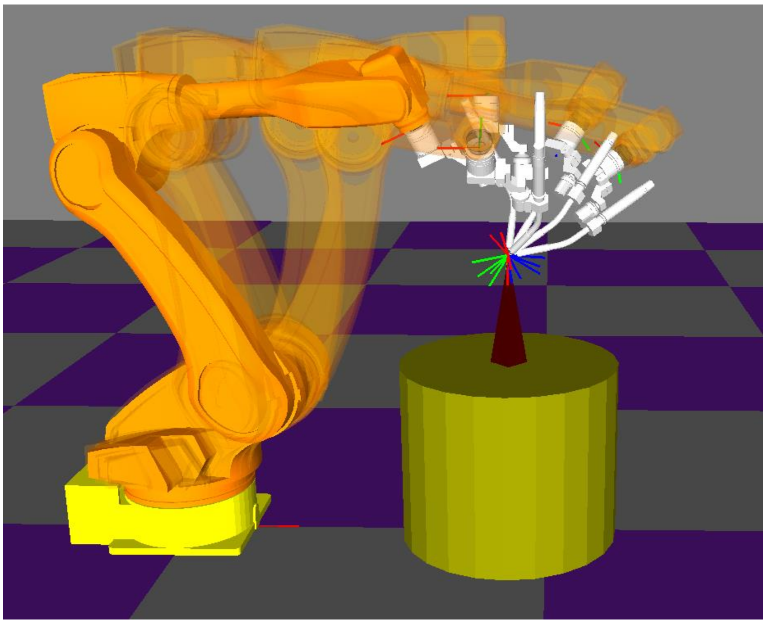


Figure 1.1 Method of Teaching for the Optimization of the Axis Origin and Tool Length

6. Set the angle of the TCP using the “tool angle calibration” function in the tool data menu.

1.1.2. Introduction of the Tool Calibration Function

To guarantee the quality of the works repeated by industrial robots, the position of the TCP should be constant. However, if the torch is damaged because of collision with a peripheral device or repeated thermal deformation, the TCP will move to a position different from the original teaching position, causing problems. In this case, the problem can be resolved either by physically restoring the deformed tool to its original state or by calibrating the changed position of the TCP. However, physically restoring the tool should be performed by a person using a separate jig for restoration and is difficult to actually execute in many cases. Therefore, the method of measuring the amount of deformation and calibrating the TCP as much as the measured deformation is more convenient and easier for automation.

When you use the tool calibration function of our robot, the robot's TCP calibration amount can be automatically measured without the operator's intervention, and thus, the measured amount can be reflected in the tool data, making it possible to continue the work according to the original teaching.

Figure 1.2 Position Taught Before Prior to Damage

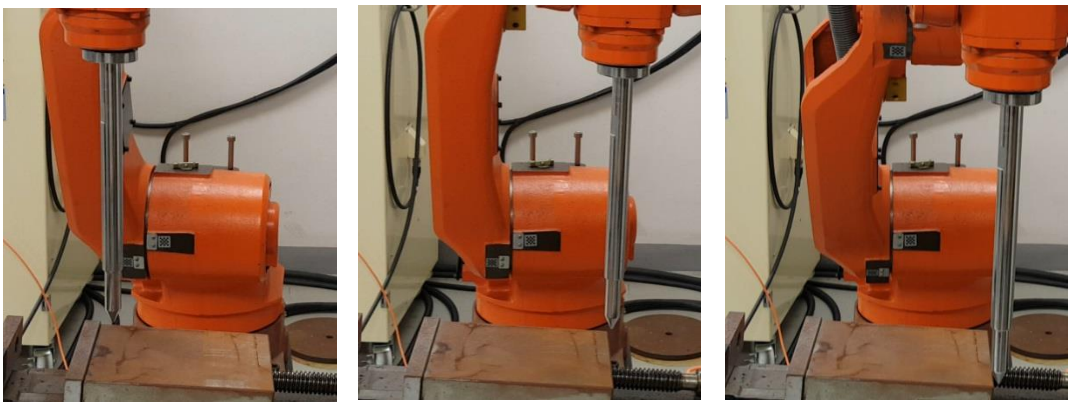


Figure 1.3 Tool Position Deviated Because of Damage to the Tool

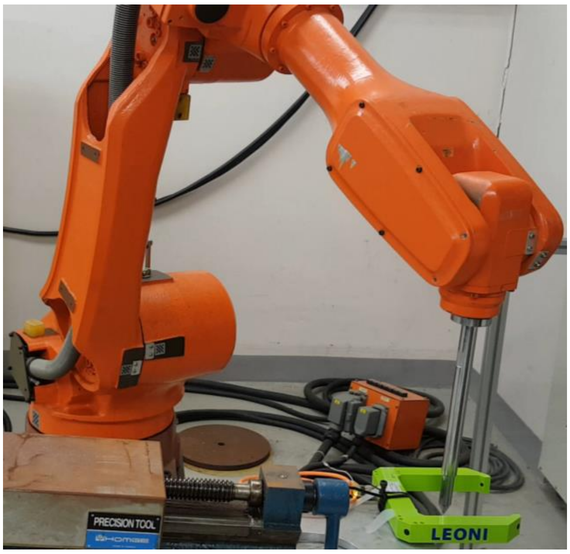


Figure 1.4 Execution of Tool Calibration

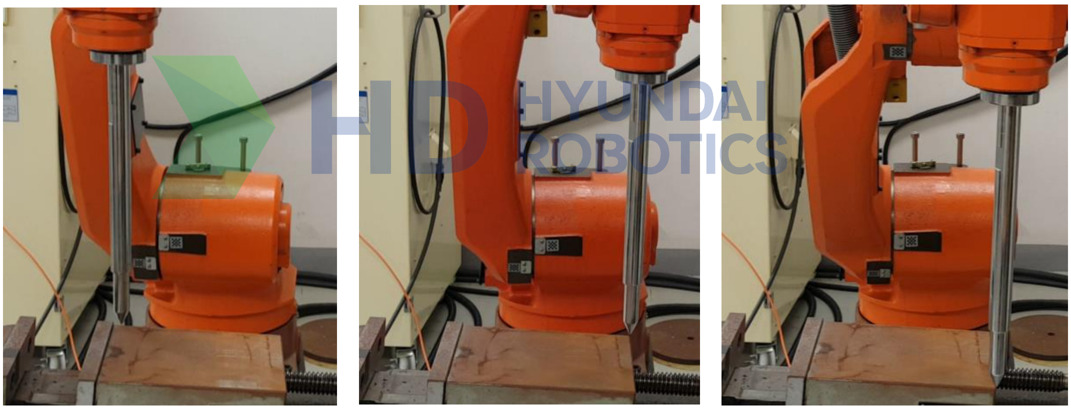


Figure 1.5 Calibrating and Restoring the Damaged Tool to the Original Position

1.2. Flow Chart of the Tool Calibration

Install the robot tool.	* Input the TCP data if the tool data model is checked. * Execute “Axis origin and tool length optimization” if there is no TCP data.
▼	
Move the operation position of the LEONI sensor.	* Move to the starting position to draw a circle around the center of the sensor. * Set the direction of the calibration program according to the direction of the starting position.
▼	
Execute the vector run.	* Operation to match the sensor’s coordinate system and the tool data’s coordinate system * Execute once after fixing the sensor position.
▼	
Execute the reference run.	* Operation for registering the TCP data of the current tool as the reference data of the sensor * When using multiple tools, register data by performing an operation for each tool.
▼	
Inspect the tool during the normal operation of the robot.	* Inspect the tool at a certain cycle. * Inspect when a collision occurs or a tool is changed.
▼	
Execute the measurement run.	* Check the amount of error compared to the reference data and reflect it into the tool data. * If the calibration range is exceeded, the robot will stop because of an error. Reference for error: 10 mm (position), 10° (posture)

2

Installation

2.1. Requirements for the System

2.1.1. Requirements for the Sensor

(1) A tool in the form of rotational symmetry such as a welding torch

(2) It should be made possible to move the tool in a circular or vertical trajectory.

(3) Function of communication between the robot and the tool through DeviceNet

2.2. Installing the System

2.2.1. Installing the Sensor

Install the light sensor unit at a proper position, as shown below. The recommended position is one within the robot's operation range where the robot can move easily, and there should be no change in the position once it is fixed.

The opened section of the sensor should be arranged to face toward the X-axis or Y-axis direction of the robot coordinate system to make it easy to perform an operation in the circle trajectory.

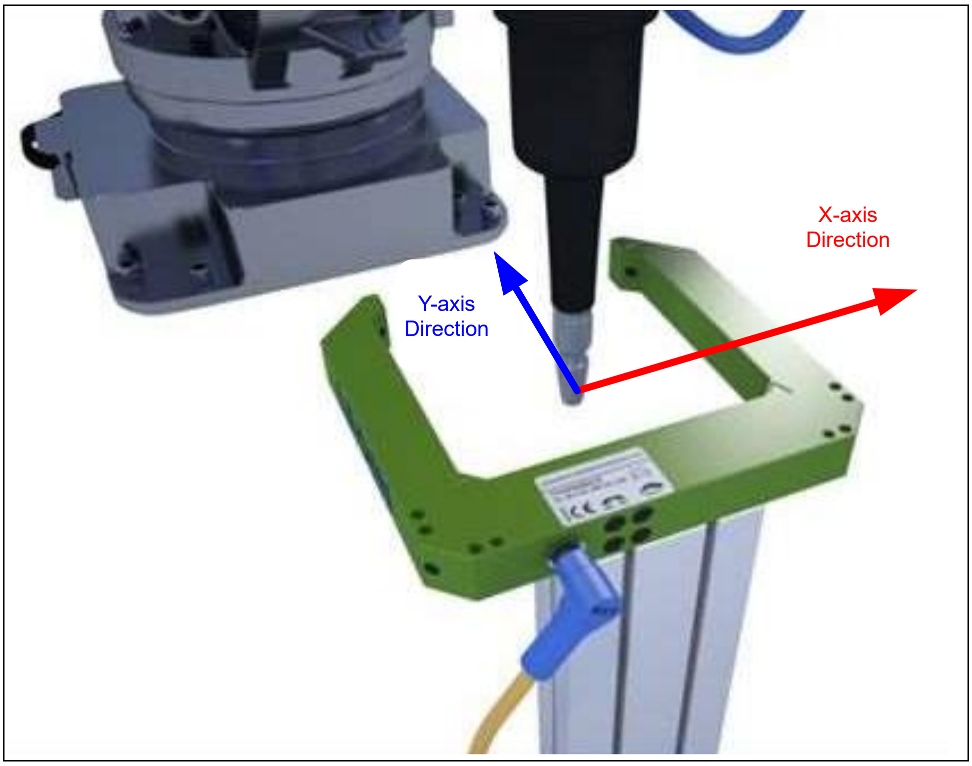


Figure 2.1 Sensor Coordinate System

2.2.2. Setting the Communication

- Set the controller DeviceNet as the master.
- Set the LEONI sensor as the DeviceNet slave #19 node.

For more information on how to set the node number of the sensor, refer to the LEONI sensor manual.

- Connect the Advintec TCP-Controller, as shown in the figure below.

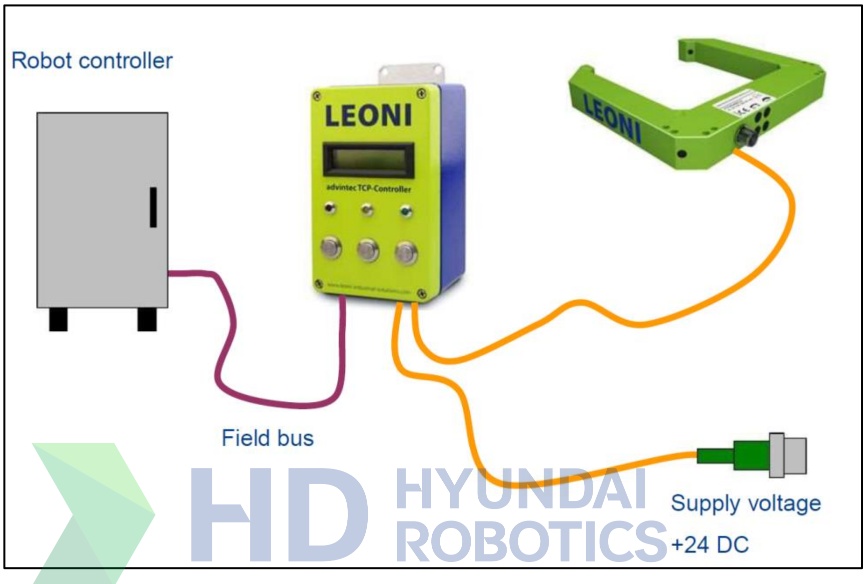


Figure 2.2 Configuration of the System Network

- The terminals at the bottom of Advintec TCP-Controller are as follows. Among them, “FB in” should be connected to the robot controller. If there is an external device, such as a welding machine other than the sensor, connect it using the “FB out” port. “LS” should be connected to the light sensor, and 24 V should be supplied to “PWR.”
- After connecting the communication line and power line, set the device built-in DeviceNet master. The CAN port should be selected according to the type of communication board to connect.
- Execute “F1: Node search” to check whether the sensor is loaded on #19.

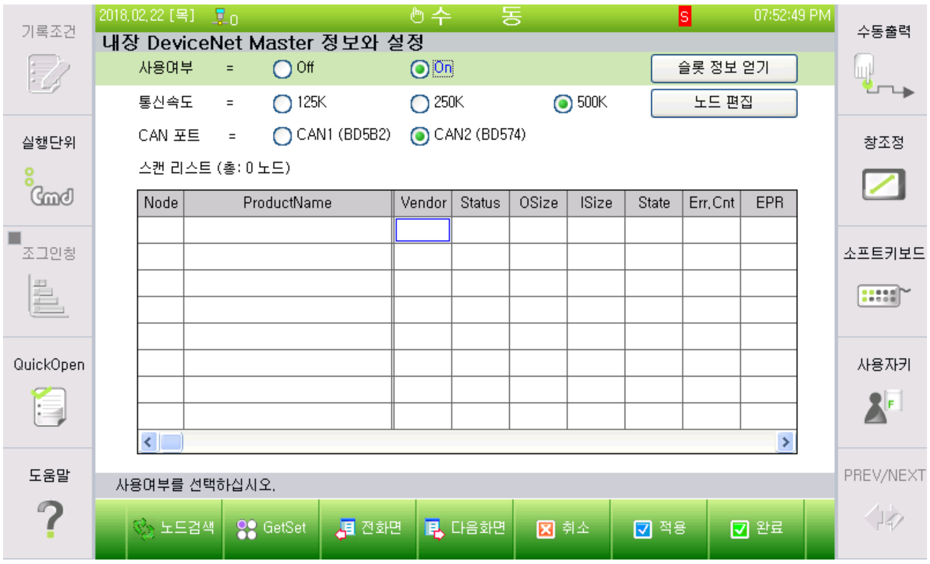


Figure 2.3 DeviceNet Master Setting and Scan Execution Screen

2.2.3. Copying the Robot Program for Tool Calibration

Insert the tool calibration program files provided by our company into the controller.

- G00.IDV: ID variable designation file for the tool calibration program
- 1000_GETMESVAL.JOB: Read the measured amount inputted from the sensor and reflect it into the tool.
- 1001_VECRUN.JOB: Program for executing Vector run
- 1002_REF2D.JOB: Program for executing 2D Reference run
- 1003_REF3D.JOB: Program for executing 3D Reference run
- 1004_REF5D.JOB: Program for executing 5D Reference run
- 1006_MES2D.JOB: Program for executing 2D Measurement run
- 1007_MES3D.JOB: Program for executing 3D Measurement run
- 1008_MES5D.JOB: Program for executing 5D Measurement run
- 1020_INIT_SET.JOB: Setting of the parameters to measure and move
- 1021_CALC_C1.JOB: Creation of the first double-circle pattern
- 1022_CALC_OS.JOB: Creation of the oscillate pattern
- 1023_CALC_C2.JOB: Creation of the second double-circle pattern

2.2.4. Checking the Byte Swap

Set the byte swap of the word data, inputted from the LEONI sensor, to On. If set normally, the position and angle data for each direction will be inputted as 30000. When the byte swap is off, the calculated coordinate data will be inputted incorrectly as 12405.

2.2.5. Designating the Pattern for Measuring and Moving

The operation for tool calibration consists of the following two patterns.

(1) Double-Circle

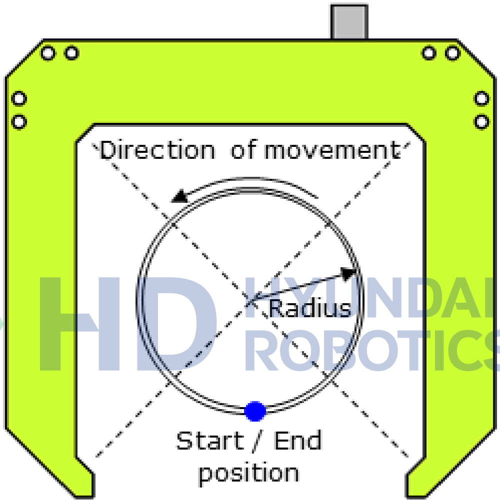


Figure 2.4 Pattern of the Robot Moving Within the Sensor: Circle

Rotate the tool by two turns counterclockwise from the start/end position, as shown in the figure. It is possible to set the diameter of the rotating circle inside the sensor. The measurement and movement patterns should be designated to operate the robot in the same circle trajectory as the set radius. The start point should be recorded in a way that the center of the circular trajectory is perpendicular to the intersection of the sensor lines.

(2) Oscillate

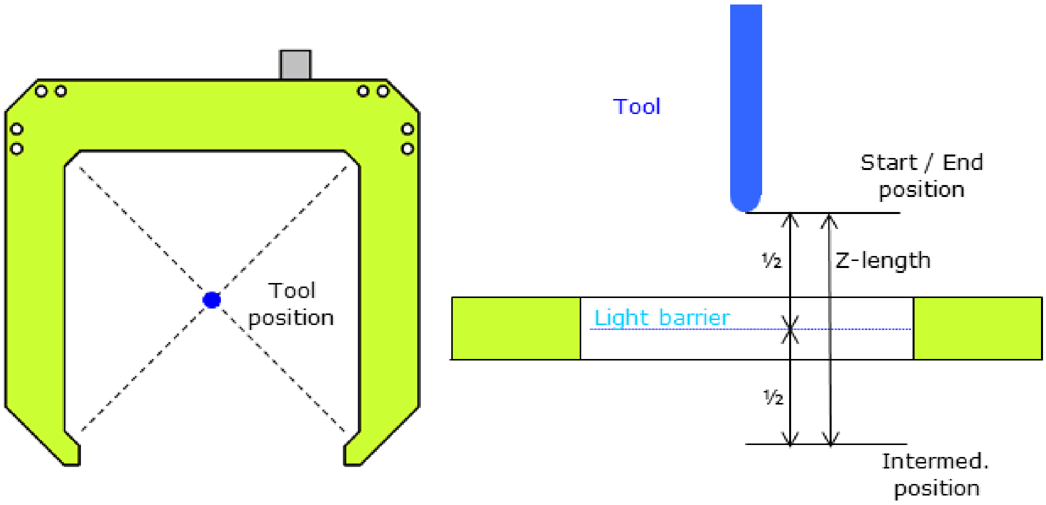


Figure 2.5 Pattern of the Robot Moving Inside the Sensor: Vertical Movement

Perform measurement in the Z direction by moving the tool up and down at the intersection of the sensor lines. Considering that the intersection point becomes the center of the double-circle, it is required to accurately teach the starting position to perform measurement accurately in the Z direction.

2.2.6. Setting the Parameters for Measuring and Moving

The parameters for measuring and moving should be set in the 1020_INIT_SET.JOB file.

giToolNrCorr=1 ----> Input of the number of the tool that needs to be calibrated

giToolNrOrig=25 ----> Input of the number of the tool that is installed for the first time

'-- moving pattern parameter

giStPoseDir=0 '0:Y+, 1:X+, 2:Y-, 3:X- ----> Direction of the start position for the measurement and movement

gdRadius=20 ----> Radius of the double circle pattern

gdMaxTrMov=20 'Max translational moving distance ----> Descending distance from the start position

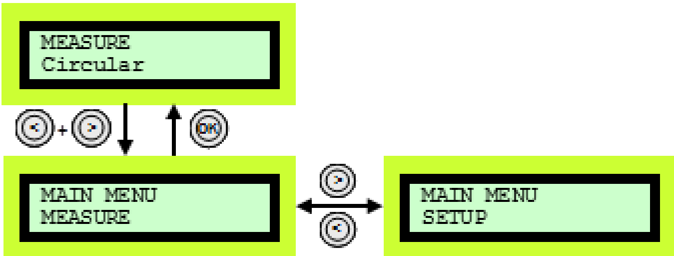
gdOscDist=30 'center - up, down distance ----> Upward and downward distance of the oscillate pattern

gdOrientLen=30 'Orientation Length ----> Descending distance of the second double-circle

gdMoveSpeed=100 'Moving speed ----> Speed of measuring and moving

2.2.7. Setting the Sensor Controller

Switching between MEASURE and SETUP mode of the sensor controller can be performed by the following operation.



Use the OK button to enter each menu. To move to the upper menu, press the arrow keys and check "<-- back -->" and then press OK.
If you press and hold the OK button when setting the parameters, the current values will be saved. If you want to exit without saving, press and hold the left and right arrow keys at the same time.

Before the measurement function is used, the following settings should be performed. For other items except for the ones described below, you may use the basic values. For details on the setting of the sensor, refer to the sensor manual.

(1) BASIC – Dimension: Selects among 2D, 3D, and 5D. The robot controller can only perform the work programs that match this setting.
(2) BASIC – Radius: Sets the radius of the double-circle pattern. This value should be equal to the radius in the robot program.
(3) BASIC – Z Length: Sets the upward and downward moving distance of the oscillate pattern. This value should be equal to the upward and downward distance in the robot program.
(4) BASIC – Orient Length: Sets the descending angle of the second double-circle pattern for measuring the tool deformation angle that is to be used in 5D. This value should be equal to the moving distance in the robot program
(5) BASIC – Options – Measure direct: Designates the direction of the tool moving forward in the measurement pattern. The basic direction is downward.
(6) BASIC – Options – Redundance. The margin of error between the calculated values inside the sensor in the process of measurement and movement. If this value is too small, an error will occur during measurement and movement, depending on the condition of the tool. The basic value is 0.1 mm. The value can be set by increasing it up to a value that will not cause an error during measurement and movement.
(7) USER – Max Tolerance: The maximum distance that can be calibrated through comparison with the reference tool. It should be set up to a maximum of 10.0 mm.
(8) USER – Max Ori Tol.: The maximum angle that can be calibrated through comparison with the reference tool. It should be set up to a maximum of 10.0 deg.
(9) GLOBAL – Bus - DevNet – Mac ID: Sets the node number of the DeviceNet slave. #19 is used basically. To use other numbers than #19, it is required to modify the G00.IDV file. Please contact our company for inquiries when required to perform this work.

※ Method of changing it directly: When you change the node number by 1, it is required to change to a value that corresponds to 128 in the case of a single signal, 16 in the case of byte, and 8 in the case of 8.

(10) GLOBAL – Bus - DevNet – Datarate: Sets the DeviceNet baudrate. Set it to be equal to the baudrate of the DeviceNet master of the controller.

(11) GLOBAL – Bus - DevNet – Byte Swap: Sets the byte swap of the word data to Yes.

HD HYUNDAI ROBOTICS

3

Guide for Functions

3.1. Vector Run

3.1.1. Overview of Vector Run

The sensor for tool calibration calculates the amounts of the X, Y, Z, RX, RY, and RZ offsets. At this time, the basic coordinate system of the sensor does not match the coordinate system of the robot. Therefore, it is required to perform an operation to make the direction of the sensor coordinate system match with the coordinate system of the robot. Such an operation is called a vector run.

A vector run is to be performed once after the sensor is installed. There is no need to perform it again later.

3.1.2. Flow Chart of Vector Run

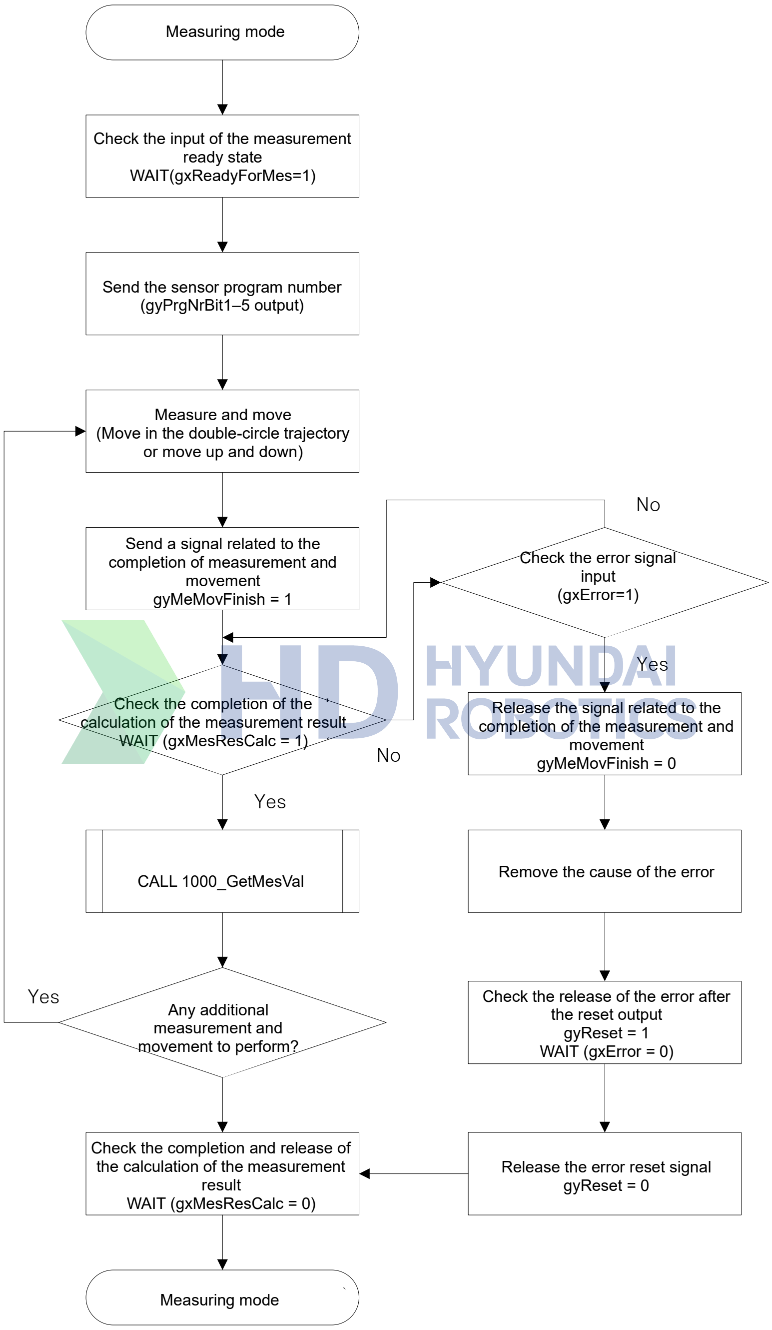


Figure 3.1 Flow Chart of the Vector Run Function

3.2. Reference Run

3.2.1. Overview of Reference Run

After the tool is properly installed, the measurement and movement should be performed to save the reference data. This reference data will be used as comparison data for tool calibration. The work of creating such reference data is called reference run. Individual reference data will be divided into 2D, 3D, and 5D data according to the setting of the measurement dimension. In the case of 2D, the double-circle will be measured once, in the case of 3D, the double-circle and oscillation will be measured once, and in the case of 5D, the first double-circle oscillation and second double-circle will be measured.

3.2.2. Flow Chart of Reference Run 2D

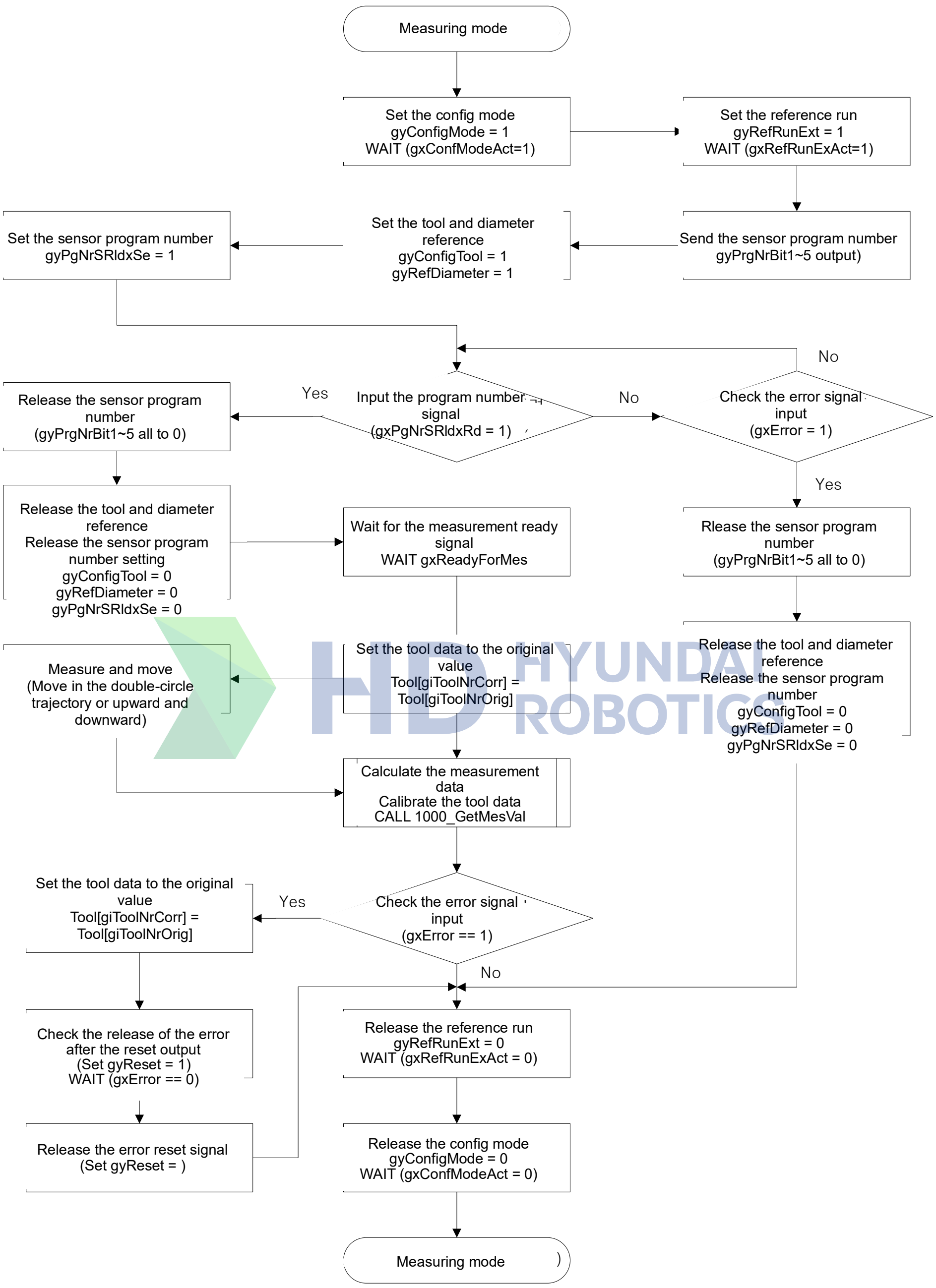


Figure 3.2 Flow Chart of Reference Run 2D Function

3.3. Measurement

3.3.1. Overview of Measurement

The tool mounted on the robot may be deformed slowly because it may collide with the surrounding facilities or it is replaced or it performs repetitive works continuously. If such a situation occurs, the production process will be disrupted, so a periodic tool measurement is required to prevent those problems from occurring. In the measurement operation, measurement and movement will be performed inside the tool calibration sensor, the amount of tool deformation will be calculated, and then calibration will be performed or an error will be detected.

3.3.2. Flow Chart of Measurement 2D

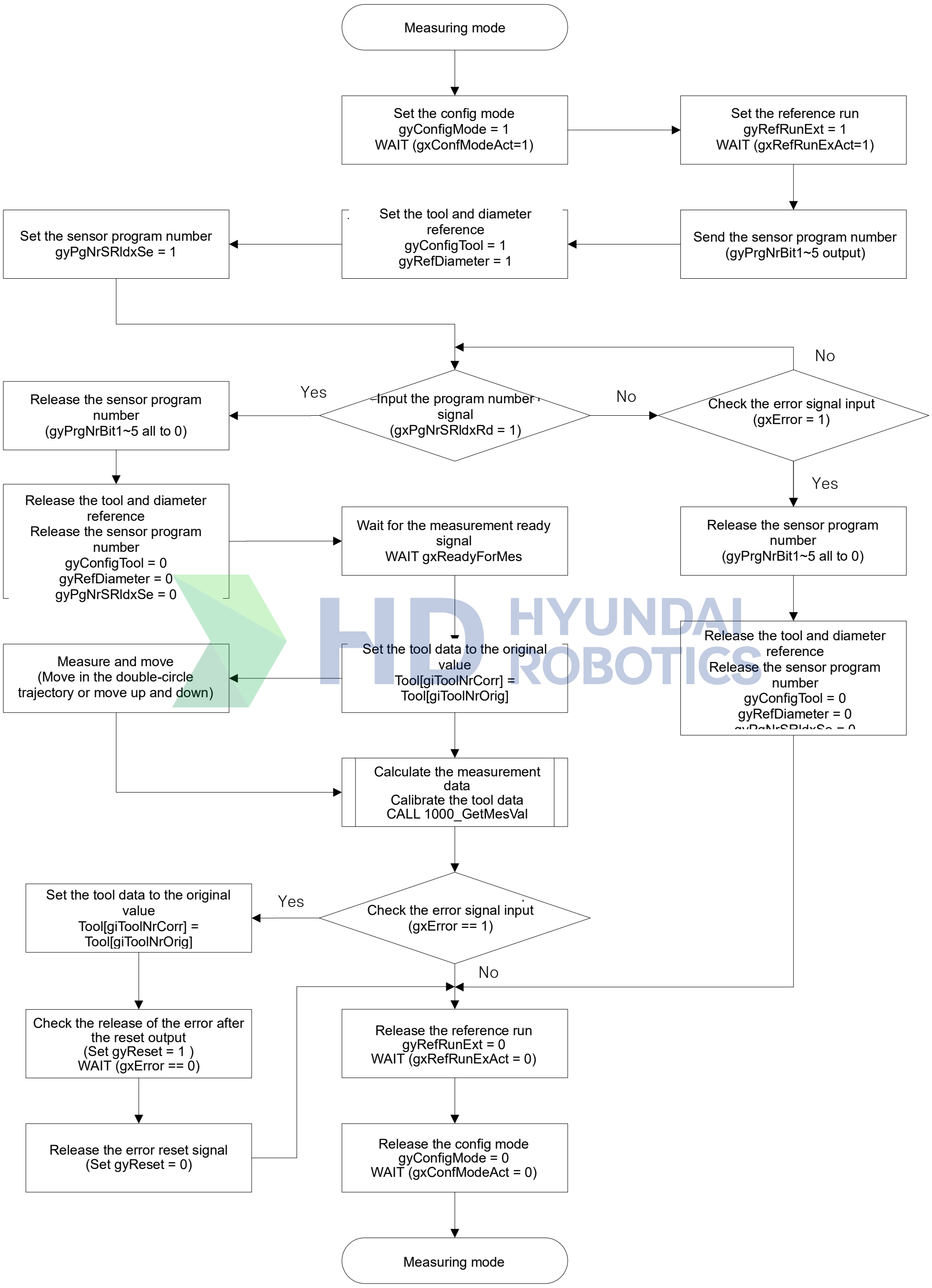


Figure 3.3 Flow Chart of 2D Measurement Function

HD HYUNDAI ROBOTICS

4

Example of Use

4.1. Contents of the Work

In this article, a simple example of the work describes how the calibration is performed when the function is actually used.

The individual steps of the work are as follows.

(1) Teach simple movements along the edges around the vise.

(2) Set the tool calibration data while the tool is in normal condition.

(3) Cause the tool to be deformed arbitrarily.

(4) Check for an error with the workpiece because of the deformation of the tool.

(5) Operate the tool calibration function.

(6) Check, via the teaching position, whether the correct amount of calibration is applied to the deformed tool.

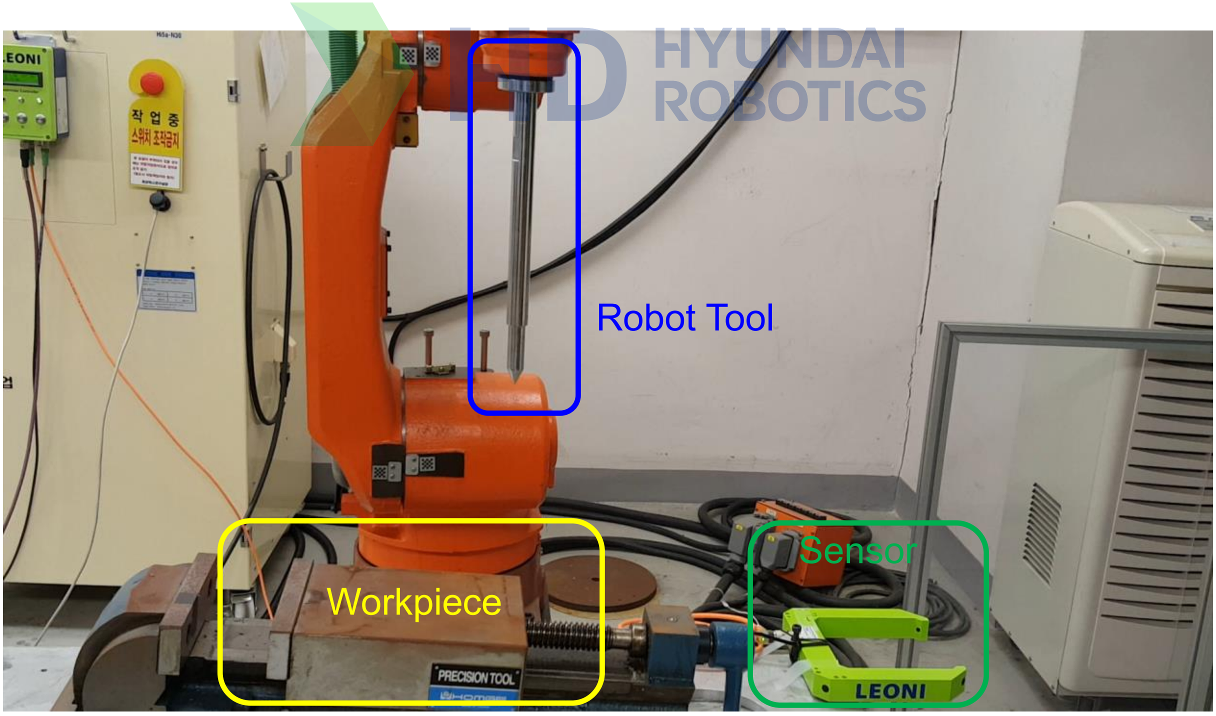


Figure 4.1 Configuration of the System for the Example Work

4.2. Registering the Direction of the Coordinate System of the Tool Data to the Sensor

First, register the tool coordinate system of the tool data by using the sensor.

```
'-- Vector-run extern without signal reflection
        '----------------------------------------------
        '----------------------------------------------
        '-- Initial parameter setting
        CALL 1020_INIT_SET

        '-- Calculate Circle Pose
        CALL 1021_CALC_C1

        '-- Activate the configuration mode -
        gyConfigMode=1
        WAIT gxConfModeAct

        '-- Activate Vector run extern --
        gyVectRunExt=1
        WAIT gxVecRunExAct

        '-- Set Program Number 1 --
        gyPrgNrBit1=1
        gyPrgNrBit2=0
        gyPrgNrBit3=0
        gyPrgNrBit4=0
        gyPrgNrBit5=0

        '-- Set to Program No. set --
        gyPgNrSRIdxSe=1

        '-- Wait for the confirmation of the program number --
        WAIT gxPgNrSRIdxRd OR gxError

        '-- Vector-run only if input gxPgNrSRIdxRd is set --
        IF gxPgNrSRIdxRd=1 THEN

        '-- Reset Program Number 1 --
        gyPrgNrBit1=0
```

```
     gyPrgNrBit2=0
     gyPrgNrBit3=0
     gyPrgNrBit4=0
     gyPrgNrBit5=0

     gyPgNrSRIdxSe=0

     '-- Wait until ready --
     WAIT gxReadyForMes

     '-- Reset tool data --
     Tool[giToolNrCorr]=Tool[giToolNrOrig]

     '-- Double-circle 1 with Tool_corrected --
S1   MOVE L,gpCirPose1,S=50%,A=0,T=giToolNrCorr                  '-- Move to the starting point
S2   MOVE C,gpCirPose2,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 1st quarter circle movement
S3   MOVE C,gpCirPose3,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 2nd quarter circle movement
S4   MOVE C,gpCirPose4,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 3rd quarter circle movement
S5   MOVE C,gpCirPose1,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 4th quarter circle movement
S6   MOVE C,gpCirPose2,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 1st quarter circle movement
S7   MOVE C,gpCirPose3,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 2nd quarter circle movement
S8   MOVE C,gpCirPose4,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 3rd quarter circle movement
S9   MOVE C,gpCirPose1,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 4th quarter circle movement

     '-- Execute "Get measured values and Correction"
     CALL 1000_GetMesVal

     '-- Double-circle 2 with tool_corrected
S10  MOVE L,gpCirPose1,S=50%,A=0,T=giToolNrCorr                  '-- Move to the starting point
S11  MOVE C,gpCirPose2,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 1st quarter circle movement
S12  MOVE C,gpCirPose3,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 2nd quarter circle movement
S13  MOVE C,gpCirPose4,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 3rd quarter circle movement
S14  MOVE C,gpCirPose1,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 4th quarter circle movement
S15  MOVE C,gpCirPose2,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 1st quarter circle movement
S16  MOVE C,gpCirPose3,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 2nd quarter circle movement
S17  MOVE C,gpCirPose4,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 3rd quarter circle movement
S18  MOVE C,gpCirPose1,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr   '-- 4th quarter circle movement

     '-- Execute "Get measured values and Correction"
     CALL 1000_GetMesVal
```

```
      '-- Double-circle 3 with tool_corrected --
S19  MOVE L,gpCirPose1,S=50%,A=0,T=giToolNrCorr                    '-- Move to the starting point
S20  MOVE C,gpCirPose2,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr  '-- 1st quarter circle movement
S21  MOVE C,gpCirPose3,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr  '-- 2nd quarter circle movement
S22  MOVE C,gpCirPose4,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr  '-- 3rd quarter circle movement
S23  MOVE C,gpCirPose1,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr  '-- 4th quarter circle movement
S24  MOVE C,gpCirPose2,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr  '-- 1st quarter circle movement
S25  MOVE C,gpCirPose3,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr  '-- 2nd quarter circle movement
S26  MOVE C,gpCirPose4,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr  '-- 3rd quarter circle movement
S27  MOVE C,gpCirPose1,S=gdMoveSpeed mm/s,A=3,T=giToolNrCorr  '-- 4th quarter circle movement

      '-- Execute "Get measured values and Correction"
      CALL 1000_GetMesVal

      '-- Acknowledge error during measurement --
      '-- Identify and rectify error cause --
      IF gxError=1 THEN
      '-- Reset tool data --
      Tool[giToolNrCorr]=Tool[giToolNrOrig]
      gyReset=1
      WAIT gxError=0
      gyReset=0
      ENDIF

      ELSE '-- Error when setting program number --
      '-- Reset Program Number 1 --
      gyPrgNrBit1=0
      gyPrgNrBit2=0
      gyPrgNrBit3=0
      gyPrgNrBit4=0
      gyPrgNrBit5=0

      gyInitTMeAct=0
      ENDIF

      '--Turn off Vector-run extern --
      gyVectRunExt=0
      WAIT gxVecRunExAct=0
```

```
'-- Turn off configuration mode
gyConfigMode=0
WAIT gxConfModeAct=0
END
```

4.3. Registering 2D Reference Data

Then, register the 2D-direction reference data that is to be used in this example of use. It is possible to register 3D and 5D reference data depending on the application.

```
'-- Reference run extern 2D without signal reflection --
        '---------------------------------------------------
        '---------------------------------------------------
        '-- Initial parameter setting
        CALL 1020_INIT_SET

        '-- Calculate Circle Pose
        CALL 1021_CALC_C1

        '-- Activate configuration mode -
        gyConfigMode=1
        WAIT gxConfModeAct

        '-- Activate reference run extern --
        gyRefRunExt=1
        WAIT gxRefRunExAct=1

        '-- Set Program Number 1 --
        gyPrgNrBit1=1
        gyPrgNrBit2=0
        gyPrgNrBit3=0
        gyPrgNrBit4=0
        gyPrgNrBit5=0

        '-- Set referencing for tool --
        gyConfigTool=1

        '-- Set referencing for diameter --
        gyRefDiameter=1

        '-- Set to Program No. set --
        gyPgNrSRIdxSe=1

        '-- Wait for the confirmation of the program number --
        WAIT gxPgNrSRIdxRd OR gxError
```

```
'-- Reference run only when input gxPgNrSRIdxRd is set --
IF (gxPgNrSRIdxRd=1) THEN
'-- Reset Program Number 1 --
gyPrgNrBit1=0
gyPrgNrBit2=0
gyPrgNrBit3=0
gyPrgNrBit4=0
gyPrgNrBit5=0
'-- Reset referencing for tool --
gyConfigTool=0
'-- Reset referencing for diameter --
gyRefDiameter=0
'-- Reset to program number set --
gyPgNrSRIdxSe=0
'-- Wait until ready --
WAIT gxReadyForMes
'-- Reset tool data --
Tool[giToolNrCorr]=Tool[giToolNrOrig]
'-- Double-circle with tool_corrected
S1  MOVE L,gpCirPose1,S=50%,A=0,T=giToolNrCorr        '-- Move to the starting point
S2  MOVE C,gpCirPose2,S=50mm/s,A=3,T=giToolNrCorr    '-- 1st quarter circle movement --
S3  MOVE C,gpCirPose3,S=50mm/s,A=3,T=giToolNrCorr    '-- 2nd quarter circle movement --
S4  MOVE C,gpCirPose4,S=50mm/s,A=3,T=giToolNrCorr    '-- 3rd quarter circle movement --
S5  MOVE C,gpCirPose1,S=50mm/s,A=3,T=giToolNrCorr    '-- 4th quarter circle movement --
S6  MOVE C,gpCirPose2,S=50mm/s,A=3,T=giToolNrCorr    '-- 1st quarter circle movement --
S7  MOVE C,gpCirPose3,S=50mm/s,A=3,T=giToolNrCorr    '-- 2nd quarter circle movement --
S8  MOVE C,gpCirPose4,S=50mm/s,A=3,T=giToolNrCorr    '-- 3rd quarter circle movement --
S9  MOVE C,gpCirPose1,S=50mm/s,A=0,T=giToolNrCorr    '-- 4th quarter circle movement --

'-- Execute "Get measured values and Correction"
CALL 1000_GetMesVal

'-- Acknowledge error during measurement --
'-- Identify and rectify error cause --
IF gxError=1 THEN
'-- Reset tool data --
Tool[giToolNrCorr]=Tool[giToolNrOrig]
gyReset=1
WAIT gxError=0
gyReset=0
```

```
ENDIF
ELSE '-- Error when setting program number --

'-- Reset Program Number 1 --
gyPrgNrBit1=0
gyPrgNrBit2=0
gyPrgNrBit3=0
gyPrgNrBit4=0
gyPrgNrBit5=0
'-- Reset referencing for tool --
gyConfigTool=0
'-- Reset referencing for diameter -
gyRefDiameter=0
'-- Reset to program no. set --
gyPgNrSRIdxSe=0
ENDIF
'-- Turn off reference run extern --
gyRefRunExt=0
WAIT gxRefRunExAct=0
'-- Turn off configuration mode --
gyConfigMode=0
WAIT gxConfModeAct=0
END
```

4.4. Teaching the Work Program

Teach the possible work programs along the edges of a rectangular structure, as shown below.

```
=== 0901.JOB ===
'Program for checking
S1    MOVE P,S=50%,A=3,T=1
S2    MOVE L,S=30%,A=0,T=1
        DELAY 0.1
S3    MOVE L,S=40mm/s,A=3,T=1         DELAY 0.1
S4    MOVE L,S=40mm/s,A=3,T=1         DELAY 0.1
S5    MOVE P,S=50%,A=3,T=1
        END
================
```

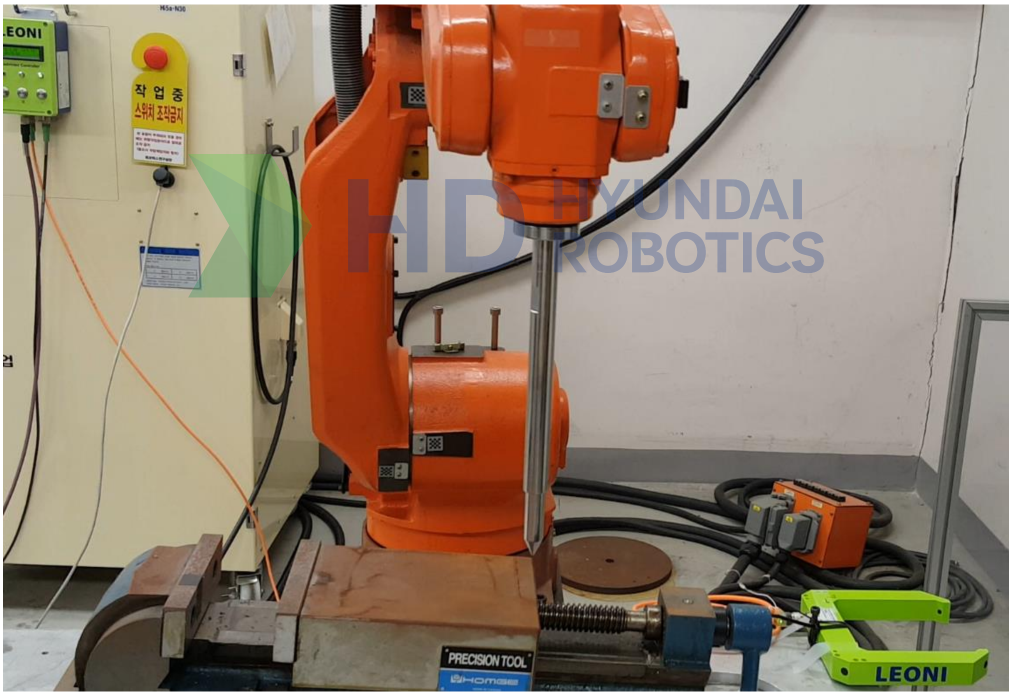


Figure 4.2 Workpiece Teaching Position Prior to the Deformation of the Tool

4.5. Forced Deformation of the Tool and Checking for an Error with the Work Position

For the purpose of the test, force the robot tool to be deformed.

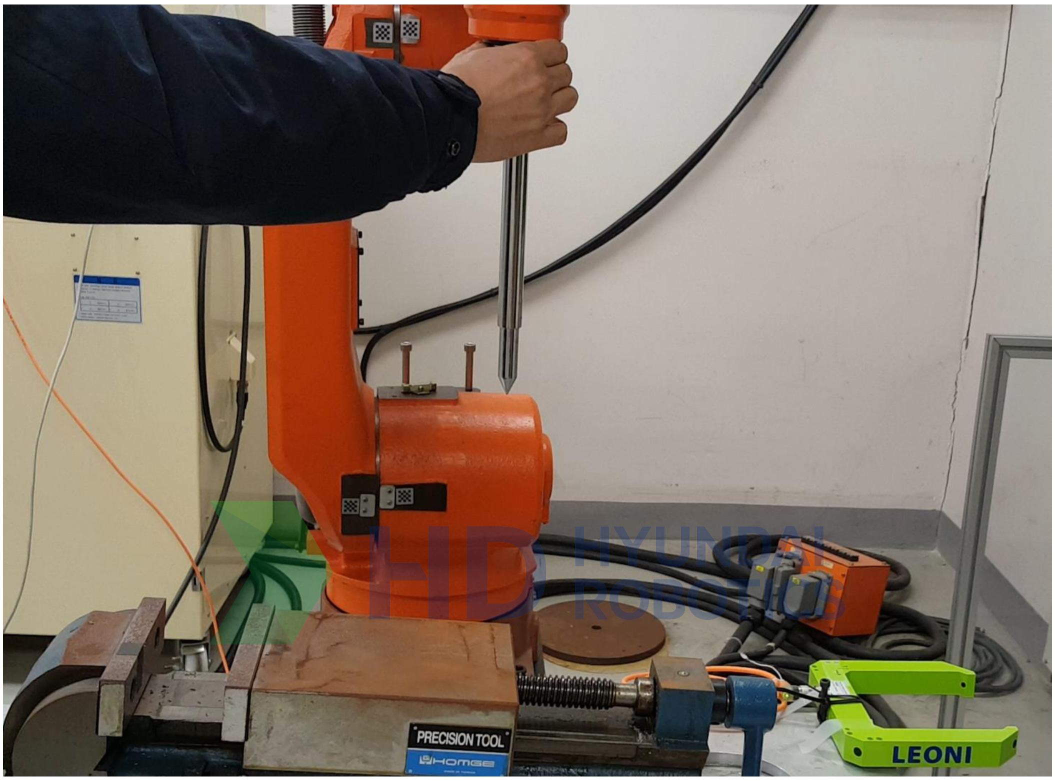


Figure 4.3 Forced Deformation of the Tool

Check that an error occurs to the workpiece because of the deformation of the tool.

Figure 4.4 An Error Has Occurred

4.6. Measuring the Amount of Deformation by Using the Sensor, and Auto-Calibrating the Tool

```
'-- Measurement 2D without signal reflection
'-- Initial parameter setting
      CALL 1020_INIT_SET

      '-- Calculate Circle Pose
      CALL 1021_CALC_C1

      '-- Reset pending error
      IF gxError=1 THEN
      gyReset=1
      WAIT gxError=0
      gyReset=0
      ENDIF

      '-- Set Program Number 1 --
      gyPrgNrBit1=1
      gyPrgNrBit2=0
      gyPrgNrBit3=0
      gyPrgNrBit4=0
      gyPrgNrBit5=0

      '-- WAIT ready --
      WAIT gxReadyForMes

      '-- Reset tool data --
      Tool[giToolNrCorr]=Tool[giToolNrOrig]
      '-- Double-circle with Tool[giToolNrCorr]
S1   MOVE L,gpCirPose1,S=50%,A=0,T=giToolNrCorr         '-- Move to the starting point      --
S2   MOVE C,gpCirPose2,S=50mm/s,A=3,T=giToolNrCorr   '-- 1st quarter circle movement --
S3   MOVE C,gpCirPose3,S=50mm/s,A=3,T=giToolNrCorr   '-- 2nd quarter circle movement --
S4   MOVE C,gpCirPose4,S=50mm/s,A=3,T=giToolNrCorr   '-- 3rd quarter circle movement --
S5   MOVE C,gpCirPose1,S=50mm/s,A=3,T=giToolNrCorr   '-- 4th quarter circle movement --
S6   MOVE C,gpCirPose2,S=50mm/s,A=3,T=giToolNrCorr   '-- 1st quarter circle movement --
S7   MOVE C,gpCirPose3,S=50mm/s,A=3,T=giToolNrCorr   '-- 2nd quarter circle movement --
S8   MOVE C,gpCirPose4,S=50mm/s,A=3,T=giToolNrCorr   '-- 3rd quarter circle movement --
S9   MOVE C,gpCirPose1,S=50mm/s,A=3,T=giToolNrCorr   '-- 4th quarter circle movement --
```

```
'-- Execute "Get measured values and Correction"
'-- Get measurement result and Correction --
'------------------------------------------
V100%=19 'sensor node number
'-- Set measurement movement to finished --
gyMeMovFinish=1

'-- Wait until the measurement result is calculated or error --
WAIT gxMesResCalc=1 OR gxError=1

'-- Read measured values only if gxMesResCalc == 1 --
IF gxMesResCalc=1 THEN
gdCorrX =(gxXOffsetBit - 30000)   / 100.0 '-- X-Offset --
gdCorrY =(gxYOffsetBit - 30000)   / 100.0 '-- Y-Offset --
gdCorrZ =(gxZOffsetBit - 30000)   / 100.0 '-- Z-Offset --
gdCorrRX=(gxXOriOffset - 30000) / 100.0 '-- Rotation around X --
gdCorrRY=(gxYOriOffset - 30000) / 100.0 '-- Rotation around Y --
gdCorrRZ=(gxZOriOffset - 30000) / 100.0 '-- Rotation around Y --
ENDIF

'-- If error gxError set correction to zero
IF gxError=1 THEN
gdCorrX =0
gdCorrY =0
gdCorrZ =0
gdCorrRX=0
gdCorrRY=0
gdCorrRZ=0
ENDIF

'-- Reset Measurement movement to finished
gyMeMovFinish=0

'-- WAIT until the calculated measurement result is reset
WAIT gxMesResCalc=0
'-- Apply correction to tool data
LP1=P*
LR1=(0,0,0,0,0,0)T
LR1.X=gdCorrX
```

```
        LR1.Y=gdCorrY
        LR1.Z=gdCorrZ
        LR1.RX=gdCorrRX
        LR1.RY=gdCorrRY
        LR1.RZ=gdCorrRZ

S1      MOVE P,P*-LR1,S=10%,A=0,T=giToolNrCorr
        LP2=P*
        ATDC T=giToolNrCorr,OrgP=LP1,NewP=LP2
        DELAY 0.1
        END
        '-- Acknowledge error during measurement --
        '-- Identify and rectify error cause --
        IF gxError=1 THEN
        '-- Reset tool data --
        Tool[giToolNrCorr]=Tool[giToolNrOrig]
        gyReset=1
        WAIT gxError=0
        gyReset=0
        ENDIF

        '-- Reset Program Number 1 --
        gyPrgNrBit1=0
        gyPrgNrBit2=0
        gyPrgNrBit3=0
        gyPrgNrBit4=0
        gyPrgNrBit5=0
        END
```

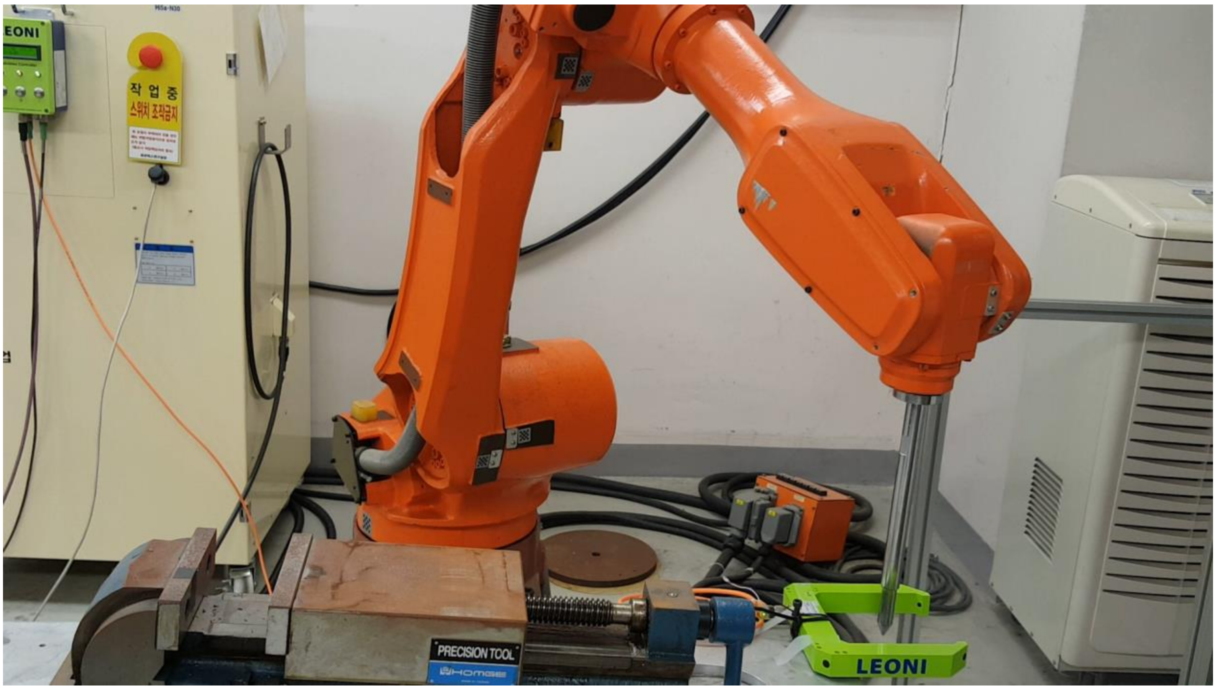


Figure 4.5 Measurement of the Amount of Tool Deformation, and Auto-Calibration

4.7. Checking the Work Program Again After Auto-Calibrating the Tool, and then Resuming the Work

After the tool calibration is completed, check the work program and resume the original work.

Figure 4.6 Checking the Position According to the Work Program

As shown above, when the existing work program is resumed, you can find that the error has disappeared. Then, you may proceed with the original work continuously.

4.8. Other Work Programs

The various work programs used in the CALL command in the example program are about the setting of the defaults of the parameters and the creation of the circle trajectory for measurement.

Program to set the defaults of the parameters:

```
'-- tool number to be corrected
giToolNrCorr=1
giToolNrOrig=25
'-- moving pattern parameter
giStPoseDir=0 '0:Y+, 1:X+, 2:Y-, 3:X-
gdRadius=20
gdMaxTrMov=20 'Max trnaslational moving distance
gdOscDist=30 'center - up,down distance
gdOrientLen=30 'Orientation Length
gdMoveSpeed=100 'Moving speed
END
```

Program to create a circle trajectory:


```
'-- copy current pose to the start pose --
gpStartP=P*
'-- calculate poses for circle
gpStartP.Z=gpStartP.Z - gdMaxTrMov
gpCirPose1=gpStartP
gpCirPose2=gpStartP
gpCirPose3=gpStartP
gpCirPose4=gpStartP
IF giStPoseDir=0 THEN
gpCirPose2.X=gpCirPose2.X - gdRadius
gpCirPose2.Y=gpCirPose2.Y - gdRadius
gpCirPose3.X=gpCirPose3.X - 0
gpCirPose3.Y=gpCirPose3.Y - gdRadius*2
gpCirPose4.X=gpCirPose4.X + gdRadius
gpCirPose4.Y=gpCirPose4.Y - gdRadius
ELSEIF giStPoseDir=1 THEN
gpCirPose2.X=gpCirPose2.X - gdRadius
gpCirPose2.Y=gpCirPose2.Y + gdRadius
gpCirPose3.X=gpCirPose3.X - gdRadius*2
gpCirPose3.Y=gpCirPose3.Y + 0
gpCirPose4.X=gpCirPose4.X - gdRadius
gpCirPose4.Y=gpCirPose4.Y - gdRadius
ELSEIF giStPoseDir=2 THEN
gpCirPose2.X=gpCirPose2.X + gdRadius
gpCirPose2.Y=gpCirPose2.Y + gdRadius
gpCirPose3.X=gpCirPose3.X + 0
gpCirPose3.Y=gpCirPose3.Y + gdRadius*2
```

```
gpCirPose4.X=gpCirPose4.X - gdRadius
gpCirPose4.Y=gpCirPose4.Y + gdRadius
ELSEIF giStPoseDir=3 THEN
gpCirPose2.X=gpCirPose2.X + gdRadius
gpCirPose2.Y=gpCirPose2.Y - gdRadius
gpCirPose3.X=gpCirPose3.X + gdRadius*2
gpCirPose3.Y=gpCirPose3.Y - 0
gpCirPose4.X=gpCirPose4.X + gdRadius
gpCirPose4.Y=gpCirPose4.Y + gdRadius
ENDIF
END
```

HD HYUNDAI ROBOTICS

● **Daegu Office (Head Office)**

50, Techno sunhwan-ro 3-gil, yuga, Dalseong-gun, Daegu, 43022, Korea

● **GRC**

477, Bundangsuseo-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea

● **대구 사무소**

(43022) 대구광역시 달성군 유가읍 테크노순환로 3 길 50

● **GRC**

(13553) 경기도 성남시 분당구 분당수서로 477

● **ARS: +82-1588-9997 (A/S center)**

● **Email: robotics@hyundai-robotics.com**