Points of Contact at DENSO Robotics in the Americas
If any problem arises with the robot unit or with this documentation, please contact us.

Applications Engineering
Robotics Planning

DENSO Products and Services Americas, Inc.
3900 Via Oro Avenue
Long Beach, CA 90810
Tel: 1-888-4-ROBOTX (888-476-2689)
E-Mail: robotics_applications@densodrive.com
Website: www.densorobots.com

Note
This document is meant is a quick guide and review on how to setup and get started using the Conveyor tracking function on the RC8 Controller. For detail information on each of these topics please consult our owner’s manual.

Link: http://densorobotics.com/content/user_manuals/19/005322.html
Table of Contents

SYSTEM OUTLINE ................................................................................................................................................. 4
Brief Description Conveyor Tracking ..................................................................................................................... 4
Hardware Installation .......................................................................................................................................... 6
  Encoder Specifications .................................................................................................................................................. 7
Setting up Encoder on RC8 ................................................................................................................................... 7
Inserting Conveyor Tracking Extension ................................................................................................................. 8
SENSOR CONVEYOR TRACKING ............................................................................................................................. 9
  Selecting the Conveyors and Devices (1 of 9) ............................................................................................................... 9
  Setting of the Encoder Joint Number (2 of 9) ............................................................................................................. 10
  I/O Setting (3 of 9) ...................................................................................................................................................... 11
  Hardware Connection Confirmation (4 of 9) .............................................................................................................. 13
  Calibration Procedure for the Conveyor (5 through 9 of 9) ....................................................................................... 14
Confirmation of the Calibration Result ................................................................................................................ 17
  Procedure of the Calibration Result ........................................................................................................................... 17
Parameter Setting .............................................................................................................................................. 18
Parameter List ................................................................................................................................................... 19
  Pass Motion of Position Designation Based on the Current Position (encoder value) .............................................. 19
PROGRAMMING EXAMPLES ............................................................................................................................... 19
  Main Program (SensorTracking.pcs) .................................................................................................................... 19
  Sensor Program (Sensor.pcs) .............................................................................................................................. 20
  Tracking Operation Program (Tracking.pcs) ......................................................................................................... 21
VISION CONVEYOR TRACKING ............................................................................................................................ 22
  Setup of the Vision Tracking System ................................................................................................................... 22
    Selecting the Conveyors and Devices .................................................................................................................. 22
    Setting of the Encoder Joint Number (2 of 15) .................................................................................................... 23
    I/O Setting (3 of 15) ............................................................................................................................................. 24
    Hardware Connection Confirmation (4 of 15) ..................................................................................................... 26
    Calibration Procedure for the Conveyor (5 to 8 of 15) ....................................................................................... 27
    Calibration Procedure for the Camera (9 to 15 of 15) ....................................................................................... 29
PROGRAMMING ................................................................................................................................................ 32
  Main program (VisionTracking.pcs) .................................................................................................................... 33
  Workpiece detection program (Vision.pcs) ............................................................................................................ 33
  Tracking operation program (Tracking.pcs) .......................................................................................................... 34
APPENDIX A. HIGH SPEED VISION CONVEYOR TRACKING USING COGNEX CAMERA .............................................. 36
  Programming ..................................................................................................................................................... 36
System Outline

**Brief Description Conveyor Tracking**
Conveyor tracking is a robot function to keep track of work pieces carried by the conveyor without stopping the conveyor and independent of speed. There are two types of conveyor tracking.

**Sensor Tracking**
The sensor tracking system is applied to pick up workpieces which are placed linearly (but with arbitrary interval) on the moving conveyor.

In this system, a signal is generated when a work piece traverses in front of a photoelectric sensor. Once this signal is generated, position data of the work piece is registered in the controller. Based on that, the controller calculates the expected work piece position on the conveyor and instructs the robot to follow the work piece. This process is performed whenever a work piece passes in front of the photoelectric sensor.

Wiring diagram is illustrated below.
Vision Tracking
The vision tracking system is applied to pick up workpieces which are placed on arbitrary position and attitude on the moving conveyor.

In this system, a signal is generated when an image recognition unit of the vision sensor detects a work piece. Once this signal is generated, the position and the angle of the work piece are registered in the controller. Based on that, the controller calculates the expected work piece position and instructs the robot to follow the work piece. This process is performed whenever the signal is generated.

A program which recognizes coordinates of workpieces operates as shown below.

Visual device starts recording once receiving an instruction from the controller. If workpiece exists, the visual device sends position data of workpiece to the controller. Once the visual device starts recording, it transmits trigger signal to the controller, then the controller immediately obtains position data from the encoder. The controller stores workpiece's position data transmitted from the visual device and position data obtained from encoder per workpiece.
Wiring diagram is illustrated below.

**Hardware Installation**
The following would need to be acquire in order to use the conveyor tracking functionality on the RC8 controller.

1 – Encoder

<table>
<thead>
<tr>
<th>Description</th>
<th>Part No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder (RC8)</td>
<td>410010-7170</td>
</tr>
</tbody>
</table>

2 – Cable

<table>
<thead>
<tr>
<th>Description</th>
<th>Part No.</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder Hub</td>
<td>410010-7100</td>
<td>Must</td>
</tr>
<tr>
<td>Encoder Cable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1m</td>
<td>410141-5950</td>
<td></td>
</tr>
<tr>
<td>2m</td>
<td>410141-5960</td>
<td></td>
</tr>
<tr>
<td>3m</td>
<td>410141-5970</td>
<td></td>
</tr>
<tr>
<td>4m</td>
<td>410141-5980</td>
<td></td>
</tr>
<tr>
<td>6m</td>
<td>410141-5990</td>
<td></td>
</tr>
<tr>
<td>12m</td>
<td>410141-6000</td>
<td></td>
</tr>
<tr>
<td>Encoder Junction Cable</td>
<td></td>
<td>Optional</td>
</tr>
<tr>
<td>2m</td>
<td>410141-5900</td>
<td></td>
</tr>
<tr>
<td>4m</td>
<td>410141-5920</td>
<td></td>
</tr>
<tr>
<td>6m</td>
<td>410141-5930</td>
<td></td>
</tr>
<tr>
<td>12m</td>
<td>410141-5940</td>
<td></td>
</tr>
</tbody>
</table>
## Encoder Specifications

<table>
<thead>
<tr>
<th>Item</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part No.</td>
<td>410010-7520</td>
</tr>
<tr>
<td>Model</td>
<td>E6B2-CWZ1X (OMRON)</td>
</tr>
<tr>
<td>Type</td>
<td>Incremental</td>
</tr>
<tr>
<td>Supply Voltage</td>
<td>5VDC (+/- 5%)</td>
</tr>
<tr>
<td>Current consumption</td>
<td>160 mA max.</td>
</tr>
<tr>
<td>Output circuit configuration</td>
<td>Line driver output</td>
</tr>
<tr>
<td>Output phase</td>
<td>A, A, B, E, Z, Z</td>
</tr>
<tr>
<td>Maximum response frequency</td>
<td>100kHz</td>
</tr>
<tr>
<td>Slew speed</td>
<td>600 r/min</td>
</tr>
<tr>
<td>Resolution</td>
<td>1000 pulses/revolution (Note 1)</td>
</tr>
<tr>
<td>Starting torque</td>
<td>9980 μNm max.</td>
</tr>
<tr>
<td>Moment of inertia</td>
<td>1×10^-6 kgm² max.</td>
</tr>
<tr>
<td>Shaft loading</td>
<td>Radial : 30 N, Axial : 20 N</td>
</tr>
<tr>
<td>Dust &amp; splash proof</td>
<td>IP50</td>
</tr>
<tr>
<td>Ambient temperature</td>
<td>-10°C to +70°C</td>
</tr>
<tr>
<td>Ambient humidity</td>
<td>35% to 85% RH</td>
</tr>
</tbody>
</table>

**Note 1:** The number of pulses is multiplied by 4, so the count value is 4000 pulses/revolution.

**Note 2:** Since this equipment is neither dust-proof nor drip-proof, protect it with a cover or something if used in an adverse environment.

**Note 3:** Protect the encoder connectors by putting covers on them after installation.

---

### Setting up Encoder on RC8

Once the encoder is mounted on the conveyor and plugged in onto the encoder hub, you will need to change the motor settings.

**Operation path:** [F2 Arm] – [SHIFT] – [F6 MAINTENANCE] (MAINTAINER Level 5596060)

**STEP 1:** Press [F10 Joint Settings]

---

For Information about how to install the encoder see OMRON’s website

More information: http://densorobotics.com/content/user_manuals/19/005328.html
STEP 2: Press [F6 Motor Settings]

STEP 3: Press [F5 Change] and select Enable encoder only

Inserting Conveyor Tracking Extension


Confirm that the Conveyor Tracking option is enabled. If the extension is not enabled enter the license key that you purchased.

More information on our website: http://densorobotics.com/content/user_manuals/19/001795.html
Sensor Conveyor Tracking

This section describes how to setup the sensor tracking system. The joint setting is disabled at default setting. With the teach-pendant, set the joint setting to "Enabling the extended-joint". For more information about this settings, see “Enabling the extended-joint”.

Selecting the Conveyors and Devices (1 of 9)

STEP 1: Displaying the Tracking Setting Window. Choose [F10 Tracking] from the top window. To use [F10 Tracking], the license of the conveyor tracking is required.

"Tracking" window is displayed.

STEP 2: Displaying the Conveyor Setting Window. Press [Settings].

Note
To set the conveyor or to edit the items of the [F12 Detail], the log-in level needs to be the Maintainer or the higher.

Note
- Before pressing [Setting], set the "Enabled" on the left side of [Setting] to OFF state. You cannot press [Setting] when the "Enabled" is set to ON state.
- Setting the conveyor to OFF state after executing one of the following commands, the value set by each command is written into the parameter.
**STEP 3:** Setting the Device. Choose [Sensor] and press [Next]. [Sensor] is selected. Once pressing [Stop] - [OK], the calibration is canceled, and then the "Tracking [Master]"

**Setting of the Encoder Joint Number (2 of 9)**

**STEP 1:** Displaying the Joint Number Selection Window. Press [Joint No].

[Joint Number] setting window is displayed.

**Tip**
- Items to be displayed in the [Joint Setting] window are as follows.
  - Joint that is set to [Enable encoder only]
  - "Enabled" joint
  - Joint which is not selected by other conveyors
STEP 2: Selecting the Joint Number and Applying the Selection. Choose a joint number then press [OK].

I/O Setting (3 of 9)

STEP 1: Displaying the Number Selection Window. Press [I/O Number].

I/O setting window is displayed.

STEP 3: Confirmation of Setting Value. The joint number selected in STEP 2 is displayed. Press [Next].

Note

To change the joint number of the encoder which is assigned to the conveyor, set the [Conveyor joint number (Conveyor*)] to "0" from [Tracking[Master]] - [F12 Detail] - [Conveyor tracking], beforehand.
**STEP 2: Setting Number** Enter the I/O number to use, then press [OK].

![Image of a settings interface]

**STEP 3: Setting the Edge Type.** Choose desired [Edge type] then press [Next]. Edge type is selected.

![Image of a settings interface]

For details about each edge type, see the table below.

<table>
<thead>
<tr>
<th>Edge Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rising edge</td>
<td>To detect a workpiece at the rising edge.</td>
</tr>
<tr>
<td>Falling edge</td>
<td>To detect a workpiece at the falling edge.</td>
</tr>
<tr>
<td>Rising and Falling</td>
<td>To detect a workpiece at the rising edge and falling edge.</td>
</tr>
</tbody>
</table>

**Note**

Available I/O numbers are listed below.

- **User input**
  - No. 8 to 15
- **Hand input**
  - No. 48 to 55
**STEP 1:** Checking Encoder Connection. Press [Check connected] button to enable the conveyor.

Operate the conveyor then check whether the value of [Current encoder value] changes properly.

**STEP 2:** Checking Trigger Detection.
Place a work piece on the conveyor to pass through the sensor, and then check whether the value of the [Data buffer] changes properly. Press [Next].

If [Data buffer] does not change, press [F1 Start trigger detection].
This section describes the calibration procedure for the conveyor, sensor and the robot. The following shows a calibration procedure.

1. Instructional video will show you how to calibrate the conveyor.
2. Sensor position ⇒ Pass through the object to the sensor.
3. Teaching of the upstream limit of the tracking ⇒ Stop the workpiece at the upstream limit of the tracking range, then execute robot teaching.
4. Teaching of the downstream limit of the tracking ⇒ Stop the workpiece at the downstream limit of the tracking range, then execute robot teaching.

**Note**
The "data buffer" in this context represents a "tracking buffer". Use WORK"0" when getting position.

**Tip**
The encoder value at the trigger signal input timing is stored in the tracking buffer.
STEP 3: Executing Teaching of Upstream Limit of the Tracking Range

Stop the workpiece at the upper end limit of the tracking range, then move the robot to the teaching position. Execute robot teaching and press [Get position of the upstream limit] to get the robot position. Checks the obtained value, and press [Next] if everything is satisfactory.

Tip
1. For about the tracking range, refer to "Parameter List".
2. When retrying to get the robot position, a system message is displayed to confirm whether to get the position.
3. The posture at the time of getting the upstream limit is set to the default posture of the tracking motion.
4. The height of gripping at the tracking motion is determined by a line which connects getting positions of the upstream limit and the downstream limit.

The obtained coordinate values can be changed. To do so, press the edit button on the bottom of the screen. Ten keys are displayed. By the ten keys, change each coordinates values.

Note
The obtained coordinates and encoder values on both of the upstream limit and the downstream limit are used to determine the conveyor’s lead length and the direction. If you change the coordinate values by the method shown above, keep in mind that the change may affect the lead length and the direction.
**STEP 4:** Executing Teaching of Downstream Limit of the Tracking Range

1. Stop the workpiece at the lower end limit of the tracking range.
2. Move the robot to the teaching position.
4. Checks the obtained value, and press [Next] if everything is satisfactory.

**STEP 5:** Confirmation of the Calibration Result

Check the calculated parameters, and press [Finish] if everything is satisfactory.

A system message is displayed to confirm whether to set the calculated data to the controller. Press [OK] if everything is satisfactory.

Calculated data is set to the controller. The setting change is not confirmed if [Cancel] is pressed.

The setting change is confirmed and returns to the "Tracking" window.
Confirmation of the Calibration Result
This section describes how to check the calibration result; obtain the current position of a work-piece by TrackCurPos command, move the robot manually by variables, check that the robot arm moves to the teaching position above the work-piece at the calibration.

Procedure of the Calibration Result
The sample program below confirms that the robot arm moves above the workpiece; execute the program to provide workpieces from the upstream of the sensor, stop the workpieces between the upstream limit and the downstream limit of the tracking, turn on IO[128] to stop the program, move to P[13] by variables

**STEP 1:** Execute the program below.

'TITLE "Get and evaluate the current position by conveyor tracking commands"

Sub Main
  Takearm Keep = 0
  Trackinitialize 1,0
  Wait IO[48] = OFF  'Move the conveyor to pass
                   'the workpiece through the sensor.
  Wait IO[48]      'IO[48]:Number of the sensor IO
  TrackSetSensor 1,1,2
  P[10] = TrackTargetPos(1)
  'Move the conveyor to stop the workpiece between
  'the upstream limit and the downstream limit.
  Wait IO[128] = OFF 'Get the current position of
  Wait IO[128]      'the workpiece at the timing of
                   'turning on the dummy IO[128]
  P[13] = TrackCurPos(1) 'The current position of
                      'the workpiece is stored in P[13].
End sub

**STEP 2:** Turn on the dummy IO[128].
**STEP 3:** Move to P[13] by variables. Check that the robot arm moves above the work-piece.
Parameter Setting
This section describes the parameter setting procedure.

**STEP 1:** Displaying Detail Setting Window. Press [F12 Detail] on the "Tracking" window.

Conveyor Tracking window is displayed.

**STEP 2:** Selecting Setting Item
Select an item to set, then press [F5 Edit].

**STEP 3:** Changing Setting Value
Input parameters with the ten keys, then press [OK].

Note
After parameters are changed in the [Detail] window, the controller needs to be restarted.
Parameter List

The parameters described in the table below are not configured automatically in the Detail window.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracking Start Range</td>
<td>Refer to the corresponding item of &quot;Terminology and Definitions&quot;. Enter a negative value to offset to the upstream direction.</td>
</tr>
<tr>
<td>Minimum Interval between workpieces (sensor)</td>
<td>This is a threshold value which prevents duplicate data of the work that already exists in the tracking buffer from being registered by TrackSetSensor command.</td>
</tr>
<tr>
<td>Low pass filter TC for conveyor pulse</td>
<td>This controls a sudden fluctuation of value by filtering the input value form the encoder. Setting a value too high lowers the responsiveness to the fluctuation of input value.</td>
</tr>
<tr>
<td>speed proportional gain</td>
<td>Use this parameter when catch up accuracy of robot deteriorates when the speed of conveyor is increased. When the robot cannot catch up the workpiece, set this value higher than 1. When the robot overpasses the workpiece, set this value lower than 1.</td>
</tr>
<tr>
<td>Stationary deviation compensation</td>
<td>Use this parameter when the robot catches up the workpiece always with a certain amount of shift. Specify the value with a positive value [mm] when the robot cannot catch up the workpiece. Specify the value with a negative value [mm] when the robot overpasses the workpiece.</td>
</tr>
</tbody>
</table>

Pass Motion of Position Designation Based on the Current Position (encoder value)
If "@a[Pass Start Displacement]" is designated at the "Pass motion of position designation" instead of "@[Pass Start Displacement]", the next line is executed when the distance between the current position (encoder value) and the target position becomes shorter than the designated value. Pass start displacement is specified by Integer-type data.

Move L, @a[50] P[10]

**Note**
As an error may be generated in a path motion of the position designation, the designated value should only be taken as a reference.

Programming Examples

**Main Program (SensorTracking.pcs)**

`'!TITLE "Sensor tracking main program"

Sub Main
  TrackInitialize 1, 0 ' Clear the conveyor 1 buffer, and then start detection a trigger from sensor
  Run Sensor                     ' Execute workpiece detection program
  Run Tracking                   ' Execute tracking operation program
End Sub`
Sensor Program (Sensor.pcs)

' !TITLE "Workpiece detection program by using sensor"
' [System header file]
' Header file that defines values for internal processing type of Variant type data.
' This file has been incorporated (registered) in RC8 before shipping.

#include "Variant.h"

Sub Main
    Dim index As Integer  ' Work number detected by sensor
    Dim workNum As Integer ' Number of workpieces to be registered in buffer at one
                           ' time sensor detection
    Dim arrayUser As Variant ' Userdata array to be set to a workpiece detected by
                            ' sensor
    Dim userData As Integer ' Userdata to be set to a workpiece detected by sensor

    ' Repeat until IO[128] turns ON
    Do
        ' Wait until the sensor detects workpieces
        ' and obtain the number of workpieces registered
        Call getWorkNumber(workNum)

        If workNum > 0 Then
            ' Declare integer type data array (userdata)
            arrayUser = CreateArray(workNum, VT_I4)

            ' Enter userdata value you want to set to userData
            userData = -1

            For index = 0 To workNum - 1
                ' Store the workpiece's userdata in array
                arrayUser(index) = userData
            Next index

            ' Add workpiece's data in the conveyor 1 buffer.
            TrackSetSensor 1, workNum, arrayUser
        End If
    Loop Until IO[128] = On
End Sub

' [User description]
' Sub-procedure that waits workpiece detection of sensor
' and then returns the number of workpiece registered.
' [out] workNum: Number of workpieces to be registered in buffer at one time sensor
' detection

Sub getWorkNumber(ByRef workNum As Integer)
    ' Wait a workpiece detection signal from sensor (detecting rising edge)
    Wait IO[48] = OFF
    Wait IO[48]

    ' Enter the number of workpieces registered at one sensor detection in workNum------
    workNum = 1
End Sub
Tracking Operation Program (Tracking.pcs)
'
TITLE "Tracking operation"

Sub Main
  TakeArm Keep = 0
  Move P, P[50]  ' Move to the home position

  ' Repeat tracking until IO128 turns ON.
  Do
    ' Move to the start position
    Move P, P[51]

    ' Take out a workdata from buffer and then set as a tracking-target work
    P[10] = TrackTargetPos( 1 )

    TrackStart 1  ' Start the tracking mode

    ' With the conveyor catch-up motion, move a robot arm to the approach ‘position' that is apart from the tracking target work by specified distance.
    TrackApproach P, P[10], 50

    ' With the conveyor catch-up motion, move a robot arm to the tracking-‘target work
    TrackMove L, P[10]

    Call Picking  ' Execute workpiece picking motion

    ' With the conveyor catch-up motion, move a robot arm from the current position to -Z direction of the tool coordinate
    TrackDepart L, 50

    TrackStop  ' End of tracking mode

    Approach P, P[52], 50  ' Move to above of the ejection position
    Move L, P[52]  ' Move to the ejection position
    Call Placing  ' Execute workpiece ejection operation

  Loop Until IO[128] = On
  End Sub

  ' Sub-procedure that executes workpiece picking motion
Sub Picking
  ' ---------Write a program for workpiece picking motion---------
End Sub

  ' Sub-procedure that executes workpiece ejection motion
Sub Placing
  ' ---------Write a program for workpiece ejection motion---------
End Sub
Vision Conveyor Tracking

Note
This function cannot be use with the cooperative control function, the force control function, nor exclusive control.

Setup of the Vision Tracking System
This section describes how to setup the vision tracking system. The joint setting is disabled at default setting. With the teach-pendant, set the joint setting to "Enabling the extended-joint". For about the way of settings, see "Enabling the extended-joint".

Tip
This section describes the setup procedure for a system which includes one robot and one conveyor.

Selecting the Conveyors and Devices
Operation path: [SHIFT] – [Tracking]

STEP 1: Press [F10 Tracking] Displaying the Tracking Setting Window

STEP 1.5: Tracking widow is displayed.

Note
In order to edit the items of the [F12 Detail], you must login as a Maintainer (5596060).
**STEP 2:** Press [Settings]

![Image of Settings menu](image1.png)

**Note**

Make sure that the conveyor that you will edit has the Enabled option set to OFF.

**STEP 3:** Choose [Vision] and press [Next]

![Image of Vision selection](image2.png)

Setting of the Encoder Joint Number (2 of 15)

**STEP 1:** Press [Joint No.] – The Joint Number windows is displayed.

![Image of Joint Number setting](image3.png)
**STEP 2:** Choose the joint number where the encoder is plugged in on the encoder hub. Press [OK]

**STEP 3:** The joint number selected in STEP2 is displayed. Press [Next]

**I/O Setting (3 of 15)**

**STEP 1:** Press [I/O Number] – The I/O setting window is displayed.
**STEP 2:** Enter the I/O number to use, then press [OK].

**STEP 3:** Choose desired [Edge type] then press [Next]. Edge type is selected.

**Note**

Available I/O numbers are listed below.

<table>
<thead>
<tr>
<th>Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>User input</td>
<td>No. 8 to 15</td>
</tr>
<tr>
<td>Hand input</td>
<td>No. 48 to 55</td>
</tr>
</tbody>
</table>

**Tip**

The I/O would be used to indicate the vision system has detected a part. Make sure your vision system is able to supply 24v DC outputs to the RC8 controller.

**For details about each edge type, see the table below.**

<table>
<thead>
<tr>
<th>Edge Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rising Edge</td>
<td>To detect a workpiece at the rising edge</td>
</tr>
<tr>
<td>Falling Edge</td>
<td>To detect a workpiece at the falling edge</td>
</tr>
<tr>
<td>Rising and Falling</td>
<td>To detect a workpiece at the rising edge and falling edge.</td>
</tr>
</tbody>
</table>
Hardware Connection Confirmation (4 of 15)

**STEP 1:** Press [Check connected] button to enable the conveyor. Turn on the conveyor and check that the value of [Current encoder value] changes properly.

**STEP 2:** Take an image of the workpiece to trigger the I/O and confirm that the value of [Data buffer] changes properly. The “data buffer” in this context represents a “tracking buffer.”

If [Data buffer] does not change, press [F1 Start trigger detection].
The following shows a conveyor calibration procedures.

**STEP 1: Confirmation of the calibration procedure.**
Instructional video of the calibration procedure is played. Once the video ends, press [Next].

**Tip**
[Skip] button is used when the conveyor calibration is not required because the setup has been done before.

**STEP 2: Execute Teaching of Upstream Limit of the Tracking Range.**
1. Stop the workpiece at the upper end limit of the tracking range then move the robot to the teaching position.
2. Execute robot teaching and press [Get position of the upstream limit] to get the robot position.
3. Checks the obtained value, and press [Next] if everything is satisfactory.

The obtained coordinate values can be changed. To do so, press the edit button on the bottom of the screen. Ten keys are displayed. By the ten keys, change each coordinates values.

**Note**
The obtained coordinates and encoder values on both of the upstream limit and the downstream limit are used to determine the conveyor feed rate and the direction. If you change the coordinate values by the method shown above, keep in mind that the change may affect the conveyor feed rate and the direction.
**STEP 3:** Execute Teaching of Downstream Limit of the Tracking Range.

1. Stop the workpiece at the lower end limit of the tracking range then move the robot to the teaching position.
2. Execute robot teaching and press [Get position of the downstream limit] to get the robot position.
3. Checks the obtained value, and press [Next] if everything is satisfactory.

**Tip**

The midpoint between the upstream limit position and downstream limit position is the conveyor reference position.

**STEP 4:** Check the calculated parameters, and press [Next] if everything is satisfactory.
Calibration Procedure for the Camera (9 to 15 of 15)
Calibration measures the exact positional relationship between the conveyor, vision sensor, and robot. The calibration result is stored as the parameter shown below and used when performing the conveyor tracking operation.

**Note**
Each conveyor needs to be calibrated respectively

**STEP 1: Confirmation of the calibration procedure**
Instructional video of the calibration procedure is played. Once the video ends, press [Next].

**STEP 2: Place 3 reference markers on the conveyor and inside the field of view of the vision system. Take a picture of the 3 markers.**

Here is a sample of a reference marker
**STEP 3:** Enter the vision coordinates of the reference markers taken by the camera. Once selecting visual coordinates for each reference point, the number pad is displayed. Enter the value then press [OK] to confirm.

Note

The camera units have to be in pixel coordinates. Do not calibrate your camera system previous to this step. The robot controller will create the camera-conveyor-robot calibration.

**STEP 4:** Without moving the reference markers, turn on the conveyor and get the markers inside the range of the robot conveyor tracking (between upstream and downstream).

Move the robot to the place where the end effectors can grip the reference point. Press [Get Position] at the TCP point.

Check the obtained position, and then press [Next] if everything is satisfactory. To teach reference point 2 and 3, perform STEP 1 and STEP 4 again.
The obtained coordinate values can be changed. To do so, press the edit button on the bottom of the screen. The number pad is displayed. By the ten keys, change each coordinates values.

**Note**

The obtained coordinates and encoder values on both of the upstream limit and the downstream limit are used to determine the conveyor feed rate and the direction. If you change the coordinate values by the method shown above, keep in mind that the change may affect the conveyor feed rate and the direction.

**STEP 5:** Check the calculated parameters, and press [Finish] if everything is satisfactory.

**STEP 6:** System message is displayed. Press [OK] if everything is satisfactory. The calculated data will be stored in the controller unless [Cancel] is pressed.
The setting change is confirmed and returns to the "Tracking" window. 

For the installation of one conveyor and one robot, this is the end of the setup.

To continue the setup for additional conveyors follow the instruction on the owner’s manual
http://densorobotics.com/content/user_manuals/19/005362.html

To continue the setup for additional robots follow the instructions on the owner’s manual
http://densorobotics.com/content/user_manuals/19/005363.html

**Programming**

The below shows programming examples of the vision tracking. This program is designed for workpieces whose length are less than the tracking range.

More info regarding programming can be found on the owner’s manual
http://densorobotics.com/content/user_manuals/19/005366.html

**Note**

- When you perform the teaching with 4-Axis robots, set the 4-th axis around 0 degrees.
- For the Control set of motion optimization (SpeedMode), select "0: None" or "1: PTP".
- Conveyor-related commands cannot be executed in the teach check mode. If you execute these commands in the teach check mode, an error "82201808 Change to automatic mode to execute the command" will be issued.
- For two or more robots, each slave obtains the workpiece data by using the same process as the master.

Vision tracking is achieved by the following three programs.

1. **Main program**
   Main program is an initialization program with regard to the conveyor tracking.

2. **Workpiece detection program**
   Workpiece detection program is a program that detects workpiece positions and saves the obtained data into conveyor tracking buffer. This is where the communication with the camera system takes place.

3. **Tracking operation program**
   The tracking operation program obtains the positional data of the workpiece from the tracking buffer area, and then executes tracking and picking up operation by use of obtained positional data.
Main program (VisionTracking.pcs)
A copy of this program is available on the owner’s manual
http://densorobotics.com/content/user_manuals/19/img/005366/VisionTracking.pcs

Clear the tracking buffer area of conveyor 1, and then start detecting a trigger from vision sensor signal.
TrackInitialize 1, 0

Execute workpiece detection program and tracking operation program concurrently.
Run Vision
Run Tracking

Tip
In order to guarantee that the vision and tracking programs start at the same time it might be good practice to terminate each one before restarting by using the KILL command.

Workpiece detection program (Vision.pcs)
A copy of this program is available on the owner’s manual
http://densorobotics.com/content/user_manuals/19/img/005366/Vision.pcs

Initialize the vision sensor. The contents of initialization are written by the user for respective vision sensors used.
Call visInitialize

Tip
For TCP/IP communication with your vision device
http://densorobotics.com/content/user_manuals/19/000608.html
The RC8 includes vision libraries to easily interface with most major vision system
http://densorobotics.com/content/user_manuals/19/005927.html

Execute workpiece detection processing with vision sensor. The contents of work detection processing are written by the user for respective vision sensors used.
Call visSearch

Tip
When sending trigger commands to the vision system from the RC8 controller, it is possible to change trigger speed based on conveyor speed by using the command TrackConveyorSpd.

Obtain the number of detected workpieces by vision sensor. The contents of processing are written by the user for respective vision sensors used.
Call getWorkNum(workNum)

Obtain the vision sensor coordinates and the userdata values assigned to workpieces as much as the number of detected workpieces. The contents of processing are written by the user for respective vision sensors used.

For index = 0 To workNum - 1
    Call getVisData(index,visionX,visionY,visionAngl,userData)
...
The following program shows the contents of sub-procedure "getVisData()". Vision sensor detection coordinates and user data are returned to the caller of the procedure through a call-by-reference argument of procedure, as a set. In the following example, a workpiece is considered as being detected when "x = 100[px]", "y = 100[px]", and θ = 90[deg].

Sub getVisData(ByVal index As Integer, ByRef visionX As Double, ByRef visionY As Double, ByRef visionAngl As Double, ByRef userData As Integer)
    visionX = 100
    visionY = 100
    visionAngl = 90
    userData = 1
End Sub

Arrange the data obtained by "getVisData()" as an array, and then add it to the tracking buffer.

For index = 0 To workNum - 1
    ... arrayVis(index) = V(visionX, visionY, visionAngl)
    arrayUser(index) = userData
Next index

TrackSetVision 1, workNum, arrayVis, arrayUser

Tracking operation program (Tracking.pcs)
A copy of this program is available on the owner’s manual
http://densorobotics.com/content/user_manuals/19/img/005366/Tracking.pcs

Move the robot arm to the home position before starting the conveyor tracking.

Move P, P[50]

The main loop for the tracking and pick-up motion that is executed whenever a workpiece is detected. In this program, turning ON of an internal I/O is specified as the loop continuation condition so that the loop can be stopped in the middle of the program.

Do
    P[10] = TrackTargetPos(1)
    ...
    TrackApproach P, P[10], 50
    ...
LOOP UNTIL IO[128] = ON

This is the processing within the loop. Pick up a workpiece from the conveyor tracking buffer and then set it to the tracking target work. If there are no workpieces detected, the program will wait until workpiece is detected and its data is stacked in the conveyor tracking buffer.

P[10] = TrackTargetPos(1)

Start the tracking mode. Please note that executing any commands designed for non-tracking mode, such as Move command, will interrupt the tracking motion. (Commands whose name do not start from "Track" will interrupt the tracking motion.) The robot does not start its motion at the timing of TrackStart command execution. The robot starts its motion once a tracking motion command, such as TrackApproach, is executed.

TrackStart 1

Wait until the tracking target work enters the tracking range.
Wait TrackInRange(1), 10000, timeoutFlg
With performing the catch up motion in order that the robot moves with the conveyor speed, move the robot arm to the approach position which is 50 mm above of the tracking target work.

\textbf{TrackApproach} \( P, P[10], 50 \)

With the catch up motion, move the robot arm to the tracking target work.

\textbf{TrackMove} \( P, P[10] \)

With the catch up motion, move the robot arm to the depart position which is 50 mm above of the tracking target work.

\textbf{TrackDepart} \( P, 50 \)

Terminate the tracking mode. Stop the catch up motion and reduce the robot arm speed until the robot fully stops. If the workpiece moves across the downstream limit of the tracking range before executing this command, an error will be issued.

\textbf{TrackStop}

Place the picked workpiece to the ejection position.

\textbf{Approach} \( P, P[52], 50 \)
APPENDIX A. High Speed Vision Conveyor Tracking using Cognex camera

In this application we use a VS068 to pick poker chips out of a conveyor. The poker chips are being detected a Cognex vision system.

Programming
Some of the parts and sections of this code have been omitted as it is not necessary to understand the main concept.

PublicVariables.h
'!TITLE "Public Variable Definition"

'----Arm Motion Constans----
#Define AUTO_SPD 100   ' Working Speed
#Define SAFE_SPD 30   ' Safe Motion Speed
#Define CONV_TOOL 2   ' Tool Offset Used for Picking Parts
#Define DEFAULT_CLR_HEIGHT 100 ' Default Clearing Height when Approaching or Departing

'----Conveyor Tracking Constants----
#Define CONVEYOR 3   ' Active Conveyor
#Define CONV_APP 50   ' Approach Height when picking parts
#Define CONV_DEP 50   ' Depart Height when picking parts
#Define OFFSET_CAL -1  ' Flag used if you intend to adjust the offset when picking from the conveyor
'----Vision System Constants----
#define CHIPSIZE  25  ' Diameter of Poker Chip's Inner Circle (mm)

'---ORiN2 Cognex Provider Constants----
Dim caoCtrl As Object   ' Cao Engine Object for ORiN Provider

MainProgram.pcs
'!TITLE "Main Program"
#include "PublicVariables.h"

Sub Main
'---Activate Conveyor Tracking----
TrackInitialize CONVEYOR, 0  ' Initialize Tracking

'---Start Vision & Motion Programs----
Kill Vision\Vision    ' Reset Vision Program
Kill Tracking\Tracking   ' Reset Tracking Program
Run Vision\Vision     ' Vision System
Run Tracking\Tracking   ' Tracking Operation Program
End Sub

Vision.pcs
'!TITLE "Workpiece detection by vision sensor"

'[System header file]
' Header file that defines values for internal processing type of Variant type data
' This file has been incorporated (registered) in RC8 before shipping.
#include "Variant.h"
#include "\VAR_TAB.h"
#include "\DIO_TAB.h"
#include "\PublicVariables.h"

Sub Main
Dim index As Integer   ' Work number detected by vision sensor
Dim workNum As Integer   ' Number of workpieces detected by vision sensor
Dim arrayVis As Variant   ' Array of the coordinate detected by vision sensor

Call visInitialize   ' Initialize the vision sensor

Do
Call visSearch
Wait IO[I_PART_FOUND] = ON  ' Execute the work detection by vision sensor
I[DATA_PROCESS_START_T] = Timer ' Start Timer

'---Obtain Number of Parts Detected----
workNum = getWorkNumber()  ' Obtain the number of workpieces detected by vision sensor

'---Extract
If workNum > 0 Then
arrayVis = CreateArray(workNum,VT_VARIANT)   ' Create Vector type data array (vision sensor detection coordinate)
For index = 0 To workNum - 1
arrayVis(index) = getVisData(index)  ' Store vision sensor detection coordinate in an array
Next index

'---Add Part Position to Vision Buffer----
TrackSetVision CONVEYOR, workNum, arrayVis ' Add work data in conveyor 1 buffer
End If

Wait IO[I_PART_FOUND] = OFF   ' Execute the work detection by vision sensor
I[DATA_PROCESS_END_T] = Timer

' Obtain Difference Between Processing time and Trigger period.
I[DATA_PROCESS_DELAY] = (I[TRIG_PERIOD]) - (I[DATA_PROCESS_END_T] - I[DATA_PROCESS_START_T])

' Delay in order to make Trigger pulse synchronous.
If I[DATA_PROCESS_DELAY] > 0 Then Delay I[DATA_PROCESS_DELAY]
Loop Until IO[128] = On
End Sub

'----Start Cognex Vision Provider----
Sub visInitialize
  Dim CtrlName As String = "Insight"        ' Instance Name
  Dim CtrlProv As String = "caoProv.Cognex.In-Sight" ' Provider Name
  Dim CtrlOpt As String = ""    ' Provider Options
  Dim CtrlConn As String = "conn=eth:192.168.0.123" ' Provider Connection Settings

  'Initialize Vision Provider (Driver)
  On Error GoTo ErrorHandler
  caoCtrl = cao.AddController(CtrlName, CtrlProv, CtrlOpt, CtrlConn) 'Start Provider
  PrintMsg (Now & ": " & "Vision Provider Initialized")
  Exit Sub

ErrorHandler:
  PrintMsg (Now & ": " & "Failed to Initialize Vision Provider")
  End Sub

'----Part Search----
Sub visSearch
  On Error GoTo ErrorHandler

  ' Send Trigger Signal Until a Part Is Found by the Vision System
  ' The Trigger Frequency is defined by the part size and the conveyor speed.
  Do While 1
    caoCtrl.SetEvent 8           ' Send Trigger Signal.
    I[TRIG_PERIOD] = (((CHIPSIZE/TrackConveyorSpd(CONVEYOR))/2)*1000

    ' Wait for IO to turn ON or for Trigger period to Complete
    Wait IO[I_PART_FOUND] = ON, I[TRIG_PERIOD]

    ' If Part is Found Exit Loop and return to main loop
    If IO[I_PART_FOUND] = On Then Exit Do
    Loop
  Exit Sub

ErrorHandler:
  Select Case Hex(Err.OriginalNumber)
  Case "80020012"     ' Divide by Zero Error
    MsgBox "Increase Conveyor Speed to Continue"
    Wait TrackConveyorSpd(CONVEYOR) > 25 ' Wait Until Conveyor Speed is Up
    Resume
  Case Else
    Err.Raise Err.OriginalNumber   ' Output Error Message
    Resume
  End Select
  End Sub

'----Parts Found----
Function getWorkNumber() As Integer
  Dim cellCol As String = "E"        ' Parts Found Column (Cognex Insight SpreadSheet)
  Dim cellRow As Integer = 25       ' Parts Found Row (Cognex Insight SpreadSheet)
---Obtain Spreadsheet Cell Value---
On Error GoTo ErrorHandler
getWorkNumber = Val(caoCtrl.GetValue(cellCol, cellRow))
If getWorkNumber > 0 Then PrintMsg (Now & ": " & getWorkNumber & " Part(s) Found!")
Exit Function

ErrorHandler:
PrintMsg (Now & ": " & "Failed to Obtain " & cellCol & Str(cellRow) & " Value")
Err.Raise Err.OriginalNumber ' Output Error Message
End Function

---Get Part Position Data---
Function getVisData(ByVal index As Integer) As Vector
Dim cellRow As Integer = 49 ' First Row on Spreadsheet for coordinate data
Dim cellXCol As String = "C"
Dim cellYCol As String = "D"
Dim cellAngCol As String = "E"

Dim visionX As Double   ' X-coordinate of work image detected by vision sensor [pixel]
Dim visionY As Double   ' Y-coordinate of work image detected by vision sensor [pixel]
Dim visionAngl As Double   ' Attitude angle of work image detected by vision sensor [deg]

' Initialize Variables
cellRow = cellRow + index

On Error GoTo ErrorHandler
visionX = Val(caoCtrl.GetValue(cellXCol, cellRow))  ' X Value
visionY = Val(caoCtrl.GetValue(cellYCol, cellRow))  ' X Value
visionAngl = Val(caoCtrl.GetValue(cellAngCol, cellRow)) ' Ang Value
getVisData = V(visionX, visionY, visionAngl)  ' Set Return Variable
PrintMsg (Now & ": Index " & index & "-> X:" & Str(visionX) & " Y:" & Str(visionY) & " Ang:" & Str(visionAngl))
Exit Function

ErrorHandler:
PrintMsg (Now & ": Index " & index & " Failed to obtain coordinate data")
Err.Raise Err.OriginalNumber  ' Output Error Message
End Function

Tracking.pcs
'!TITLE "Tracking operation"

#include ".\VAR_TAB.h"
#include ".\EOAT\Vacuum.pcs"
#include ".\PublicVariables.h"

Sub Main
TakeArm Keep = 0
Call ResetVacuum ' Stop Vacuum

---Move to Save Position----
ChangeTool CONV_TOOL ' Change Tool
Speed SAFE_SPD ' Set Safe Speed
Depart L, @P DEFAULT_CLR_HEIGHT ' Move Linearly Upwards to avoid collision
Move P, @P [HOME_MAIN] ' Move to the home position
Move P, @P [HOME_CONV] ' Move to the start position near the conveyor
Speed AUTO_SPD ' Set Work Speed
HighPathAccuracy True ' Enable High Path Accuracy Option
'---Conveyor Tracking Loop----
Do ' TrackTargetPos will return a part from the Tracking Buffer (FIFO Type).
   ' If the buffer is empty the program will halt until a part is added to
   ' the buffer (by the Vision Program)
P[CONV_PICK] = TrackTargetPos(CONVEYOR)

'----Pick Part From Conveyor----
Call SetVacuum ' Start Vacuum

' Start Part Tracking. The robot will keep monitoring the part position based
' on the encoder mounted in the conveyor so it can update its path when moving
' down to pick.
TrackStart CONVEYOR ' Start the tracking mode
TrackApproach P, P[CONV_PICK], @P CONV_APP ' Approach Pick
TrackMove L, @a[1] DevH(P[CONV_PICK], P[PICK_OFFSET]) ' Pick Part (Add Offset if needed)
TrackDepart L, @P CONV_DEP ' Depart From Pick
TrackStop ' End of tracking mode

'----Place Part on Staging Area----
Approach P, P[PLACE_PART], @P CONV_APP ' Approach Place Point
Move L, P[PLACE_PART] ' Place Part
If OFFSET_CAL = 0 Then Wait IO[129] = ON
Call ResetVacuum ' Stop Vacuum
Depart L, @P CONV_DEP ' Depart Place Point
Loop Until IO[128] = On
HighPathAccuracy Off 'Disable High Path Accuracy Option
Motor Off 'Turn Motor Power Off
End Sub