

IP1000 Vision Process Package User Manual

Revision Date: 2026/06/08 (V1.0)

Table of Contents

1. Process Introduction	2
1.1. Parameter Settings	2
1.1.1. Vision Parameter Settings	2
1.1.2. Vision Range Settings	5
1.1.3. Vision Position Parameters	7
1.1.4. Position Debugging	8
1.1.5. Vision Calibration	9
1.2. Vision Instructions	11
1.2.1. VISION_RUN - Vision Start	11
1.2.2. VISION_TRG - Vision Trigger	12
1.2.3. VISION_POSNUM - Get Vision Position Count	12
1.2.4. VISION_POS - Get Vision Position	13
1.2.5. VISION_CLEAR - Clear Vision Position Data	15
1.2.6. VISION_END - Vision End	16
1.2.7. VISION_TRACE - Get Vision Trajectory Position	16
1.2.8. ONFLY_INIT - Flying Capture Start	17
1.2.9. ONFLY_END - Flying Capture End	20
1.2.10. ONFLY_GET_STATUS - Get Flying Capture Status	20
1.2.11. ONFLY_CALC_TOOL - Flying Capture Calculate Tool	21
1.3. Vision Process Examples	23
1.3.1. Example 1: Camera captures 10 times, then vision closes	23
1.3.2. Example 2: Camera sends continuous trajectory positions	23
1.3.3. Example 3: Additional data parameter description for tracking position	24
1.3.4. Example 4: After camera captures workpiece image, send data to robot for picking	27

1. Process Introduction

Identify and locate workpieces for more accurate robot picking. Features: precise positioning, easy deployment, simple to use, flexible parameter configuration, rich instruction set. Often combined with conveyor tracking for effective workpiece tracking and picking with simple programming.

1.1. Parameter Settings

Click Process > Vision Process to enter vision process parameter settings, as shown:



1.1.1. Vision Parameter Settings

Vision parameter settings screen shown below:



1. Camera Selection:

Process No.: Provides 1-99 process numbers. Each saves all vision and position parameters under it.

Type: Currently only custom type supported. Users can set parameters as needed.

2. User Coordinate System:

This system supports mapping vision positions to user coordinate system. Camera sends positions in vision coordinate system. Select a user coordinate system matched with the camera.



As shown above, if None is selected, camera sends positions in Cartesian coordinates by default.

Users can also select their own calibrated user coordinate system (calibrated in Settings > User Coordinate Calibration).

3. Network Parameters

Camera IP:

Enter camera IP when camera is server. Camera and controller IPs must be on same subnet.

When camera is client, enter connected controller IP. Camera as client means controller is server; camera initiates connection.

Camera: Client, Server

Port Count:

If vision server uses same port for send/receive, port count is 1. If different ports, port count is 2.

Port 1 is receive, Port 2 is send (port numbers cannot be the same)

Camera Data: Robot coordinates, Pixel coordinates

If robot coordinates selected, camera data is robot coordinates. If pixel coordinates selected, camera data is pixel coordinates in camera system.

4. Connection Parameters

Frame Header: Header before position data, must match camera config.

Separator: Between each data value. Must match camera config (cannot be empty).

End Marker: Marker after position data. Must match camera config.

Success Flag: Camera sends success flag after capturing and identifying workpiece.

Failure Flag: Camera sends failure flag when capture succeeds but recognition fails.

Timeout: Controller reports error (Vision Trigger Timeout) when exceeded.

Range (0,5000]s.
Single Target: Disable to identify multiple target positions - Single target data format example: @, x, y, Rz, \$ Format description: “@” is frame header; “,” is separator; “\$” is end marker; “x, y, Rz” is the actual position values - Multi-target data format example: @, N, x, y, Rz, x, y, Rz, \$ Format description: “@” is frame header; “N” is number of positions sent ; “,” is separator; “\$” is end marker; “x, y, Rz” is the actual position values.

Note: Enabling single target only identifies one position
Type: 2D, 2D+Height, 3D Example: (Header “ !” Separator “,” End “\$”) 2D: Data format: !, X, Y, Rz, \$ 2D+Height: Data format: !, X, Y, Rz, h, \$ 3D: Data format: !, X, Y, Z, A, B, C, \$ Additional data count: Stores non-position camera data (tool number, color , position validity etc.), User-defined additional data Value will defaults to first variable GS001, Additional data count store Example: Custom 1 = black. Camera sends additional data 1. After parsing to variable, camera captured black workpiece. Use conditional logic to set variable to 1, robot picks black workpiece. Example: (Header “ !” , Separator “,” , End “\$”) Single target format: !, X, Y, Rz, data..., \$ Multi-target format: !, N, X, Y, Rz, data..., X, Y, Rz, data..., \$

5. Trigger Mode

I0: Send trigger signal to camera, set I0 signal port here
Ethernet: Default is Ethernet send, When Camera receives “TRG” (or Custom) after, should reply with coordinates to controller.
Trigger Condition – Single: When set to single, vision trigger instruction fires only once.
Trigger Condition – Continuous: When set to continuous, vision trigger instruction fires continuously.
Trigger Period: Time interval for continuous triggering

6. Receiving Coordinate System

Received position info is camera-sent position with specific tool and user
--

coordinate system.
Tool: Enable to include tool number in camera positions (for multi-tool operations).
User: Enable to include user coordinate system in camera positions (for multi-station).
Note: Before enabling User, hand-eye calibration user coord system cannot be None (when off, set to None) , User and Tool enable can be toggled simultaneously

7. Radian/Degree

A/B/C axis unit type in vision position parameters. Radian unit is rad, degree unit is degrees.
Note: Degree/radian switching in vision parameters affects data parsing. Angle/radian settings in operation parameters only affect teach pendant display.

1.1.2.Vision Range Settings

To prevent camera position parameters from exceeding robot reachable range, the maximum robot range is defined. If camera data exceeds range, a warning is displayed and data is automatically filtered. Calibration can be done by manual teaching or by entering parameters directly.



参数	值	注释
MX		X轴最大值(mm)(不填写则表示无限制)
mX		X轴最小值(mm)(不填写则表示无限制)
MY		Y轴最大值(mm)(不填写则表示无限制)
mY		Y轴最小值(mm)(不填写则表示无限制)
MZ		Z轴最大值(mm)(不填写则表示无限制)
mZ		Z轴最小值(mm)(不填写则表示无限制)

修改 清除

返回 范围标定

Process No. : Provides 1-99 process numbers. Each saves its own vision range parameters.

Range Calibration: Calibrate max and min values of XYZ axes in Cartesian coordinates.

坐标轴	最大值	最小值	标定值
X			
Y			
Z			

标定MX	标定MY	标定MZ
标定mX	标定mY	标定mZ
标定完成		

返回

Calibrate MX: Calibrate X-axis maximum;

Calibrate mX: Calibrate X-axis minimum;

Calibrate MY: Calibrate Y-axis maximum;

Calibrate mY: Calibrate Y-axis minimum;

Calibrate MZ: Calibrate Z-axis maximum;

Calibrate mZ: Calibrate Z-axis minimum;

Calibration Complete: Records calibrated X, Y, Z max and min range values.

1.1.3.Vision Position Parameters

工艺/视觉工艺/视觉位置参数设置

工艺号1

机器人抓取时的姿态
(坐标系都是直角坐标系)

高度可手动填写

偏移补偿

X方向偏移0.000mm

Y方向偏移0.000mm

Z方向偏移0.000mm

角度偏移0.000

接受位置类型点

角度方向正方向

比例系数1.000

示例格式: x,y,Rz,\$

接收数据:

基准点

值

X值0.00

Y值0.00

Z值0.00

A值0.00

B值0.00

C值0.00

相机坐标

值

X0.00

Y0.00

高度0.00

角度0.00

标定抓取姿态

运行到基准点

清除标定

试拍照

运行到该点

返回

修改

Process No. : Provides 1-99 process numbers. Each saves its own vision position parameter settings.

1. Offset Compensation: If robot picking position consistently has a fixed directional offset from actual position, enter compensation amount here for automatic correction.
2. Scale Factor: If camera sends position values scaled by a specific ratio, enter scale factor here. For example, if camera sends (300,200,100) but actual position is (3,2,1), enter 0.01 here.

Compensation: Cartesian coordinate deviation between picking position and actual position.
Formula: Scale Factor = Actual Position Value / Camera Sent Position Value.
3. Angle Direction: Camera sent position angle same or opposite to robot rotation angle.
4. Receiving Position Type: Point / Trajectory.

Point: Camera captures and sends single position to controller.
Trajectory: Camera identifies trajectory and sends a series of positions, executed via external point commands.

When receiving position type is set to Trajectory, the program file is as follows:

0

开始

1

点到点 P0001 速度10% 平滑0 加速度10 减速度10 0

2

视觉开始 工艺号:1

3

视觉触发 工艺号:1

4

获取视觉位置个数 工艺号:1 GI001

5

如果 (GI001 > 0)

6

获取视觉轨迹位置 工艺号:1 外部点运动队列 GD001

7

外部点 关节 速度10% 平滑0 加速度20 减速度20 0

8

结束如果

9

视觉结束 工艺号:1

10

结束

5. Robot Picking Posture

Calibrate Picking Posture: Mark the robot end-effector posture when picking. After calibration, every

pick uses this posture (XYZ values here do not affect picking position).

Move to Reference Point: Move to calibrated picking posture position.

Clear Calibration: Clear calibrated picking posture data.

6. Camera Coordinates

Camera Coordinates: If camera cannot send picking height, enter Z picking height in right table.

If camera can send picking height, this setting is ignored. After setup, press DEADMAN to power on and click [Test Capture]. Camera data displays in Camera Coordinates and Received Data.

Move to Point: After capture, press DEADMAN to power on, click [Move to Point] to move robot to capture position for verification.

Test Capture: With servo powered on, click Test Capture to open network connection and receive data. Warning if format does not match example.

Example Format: Based on connection parameters set in Vision Parameter Settings screen.

For example 2D+Height: Frame header is @, separator is ,, end marker is \$, format is @, x, y, Rz, h, \$.

Received Data: Position sent by camera.

For example: Camera sends x=100, y=-50, Rz=1, h=1000. Received data displays as @, 100, -50, 1, 1000, \$.

1.1.4. Position Debugging

Used with conveyor for debugging. After camera capture, position is sent and stored in both [Original Position] and [Offset Position]. Workpiece moves with conveyor. Click Calculate Offset to compute offset position overwriting [Offset Position]. Click Move Here and robot goes to calculated offset position.

Enter via Process > Vision Process > Position Debugging. Used for conveyor debugging when using vision + conveyor tracking.

工艺/视觉工艺/位置调试									
工艺号 1					传送带工艺号 1				
原始点位	UX	UY	UZ	角度	偏移后点位	UX	UY	UZ	角度
原1					偏1				
原2					偏2				
原3					偏3				
原4					偏4				
原5					偏5				
原6					偏6				
原7					偏7				
原8					偏8				
原9					偏9				
原10					偏10				

Process No.: Vision process number.

Conveyor Process No.: Conveyor process number to debug.

Capture: Press DEADMAN to power on, click [Capture] to test. Camera position data displays in [Original Position] and [Offset Position].

Move Here: After capture, press DEADMAN to power on, select position and click [Move Here]. Robot moves to camera sent position.

Calculate Offset: After capture, run conveyor to move workpiece. Click Calculate Offset to update [Offset Position] with offset workpiece position.

Clear: Clear all positions.

1.1.5.Vision Calibration

Vision calibration establishes transformation from camera coordinates to robot coordinates, determining target position in robot coordinate system for accurate robot movement to target.

Based on whether camera is mounted on robot arm, calibration is mainly Eye-in-Hand and Eye-to-Hand. Detailed instructions follow.

1. Eye-in-Hand

Used when camera is mounted on tool. Camera lens should be roughly parallel to calibration template plane, and relative height between camera and template must remain constant during calibration. Mark robot positions including at least 3 same-posture points and 3 different-posture points. Run calculation - robot moves to each position and captures pixel data. After all data collected, camera-to-robot coordinate transformation is calculated.

工艺/视觉工艺/视觉标定

工艺号: 1

眼在手内 眼在手外

序号	像素U	像素V	机器人X	机器人Y	机器人Rz
1	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00

标定点个数: 6

注：至少标定3个平移点和3个旋转点，标定点个数范围（6-30）。

Process No.: Vision process number.

Calibration Points: Number of points to teach, range (6-30).

Points: Minimum 6, maximum 30. At least 3 different positions and 3 different postures required. (e.g., move X/Y axis to mark 3 points, then rotate C axis to mark 3 more).

Mark Point: Record current robot position data.

Move to Point: Press DEADMAN to power on, select index and click [Move Here]. Robot moves to marked position.

Clear Point: Clear position data for selected index. Pixel data not cleared – recaptured after running calculation.

Run Calculation: Robot automatically moves to each taught position, triggering capture and recording pixel data at each point. After collection, calibration calculation runs automatically and displays results.

2. Eye-to-Hand

Camera fixed on bracket (shooting downward) or mounted below target (shooting upward). Camera lens should be roughly parallel to calibration template, and relative height between camera and template must remain constant during calibration.

Note: Calibration template is fixed to robot tool end-effector.

Method Overview: 3 steps: hand-eye calibration, tool calibration, and high-precision tool calibration.

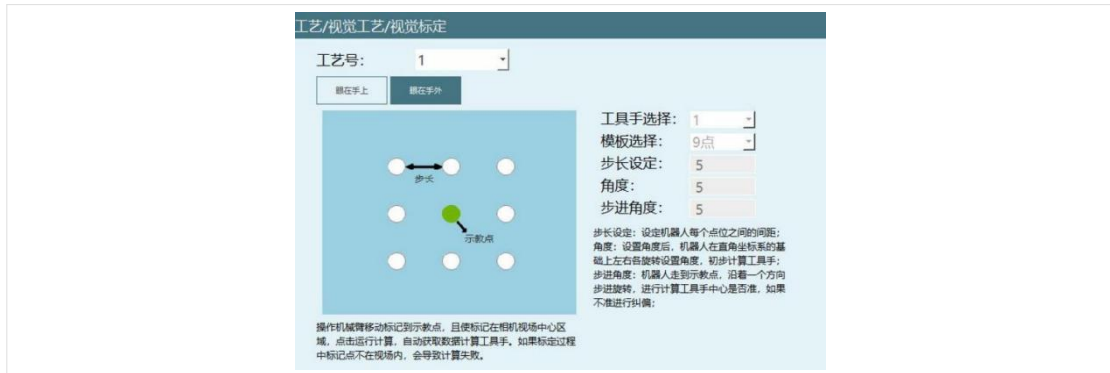
Note: This method can only calibrate tool offset on view plane, suitable for SCARA robots.

Steps:

1. 9-point (16-point) hand-eye calibration: Move robot so tool template is at center of field of view. Click Run Calculation and system auto-calculates 9 (16) positions, moving to each and collecting 9 (16) sets of coordinates (pixel and robot without-tool coordinates). System then calculates camera-to-robot transformation.

2. Tool calibration: Move robot to 1st marked point from step 1. Using flange center as calibration plane center, rotate tool in both directions by set angle, collecting 2 sets of coordinates. System auto-calculates the tool.

3. High-precision tool calibration: Move robot to 1st point from step 1. Using tool end from step 2 as center, rotate tool around center on calibration plane by step angle (tool position unchanged, posture changes) 6 times, recording 6 sets of coordinates. Then move to 1st point again and rotate 6 times in opposite direction. After



Tool Selection: Select tool to calibrate.

Template Selection: 9-point or 16-point.

Note: 9-point calibration covers 3x3 grid. Verify robot can reach all four corners. If not, adjust camera range or reduce step spacing. Same for 16-point.

Step Size: Auto-calculates spacing between 9 (16) points based on current robot position.

Angle: After setting angle, robot rotates left and right by set angle in Cartesian coordinates for initial tool calculation.

Step Angle: In step 3 high-precision tool calibration, robot rotates around tool end in one direction by this angle per step.

1.2. Vision Instructions

1.2.1.VISION_RUN - Vision Start



Format: VISION_RUN [Command] ID=1 [Process No.].

Function: After executing Vision Start, controller connects to camera.

Parameters:

ID	Range [1,99]. Process No. in instruction must match vision process screen selection.
----	--

1.2.2.VISION_TRG - Vision Trigger



Format: VISION_TRG [Command] ID=1 [Process No.].

Function: After executing Vision Trigger, waits for vision server to return position data. Continues to next instruction after receiving data.

Parameters:

ID	Range [1,99]. Process No. in instruction must match vision process screen selection.
----	--

Note: Trigger mode is set in Vision Process > Vision Parameter Settings:

1. IO trigger: Running this instruction sends corresponding IO signal.
2. Ethernet: Running this instruction sends custom string to camera.

1.2.3.VISION_POSNUM - Get Vision Position Count



Format: VISION_POSNUM [Command] ID=1 [Process No.] GI001 [Global Numeric Variable]. Function: Records position count sent by camera and stores into selected variable.

Parameters:

ID	Range [1,99]. Process No. in instruction must match vision process screen selection.
Global Numeric Variable	Store camera position count into selected variable E.g.: Camera sends 3 positions, selected variable is GI001, after executing this instruction GI001=3 Note: each Get Position execution, position count decreases by 1

Example:

1. NOP
2. VISION_RUN ID = 1
- Vision Start
3. VISION_TRG ID = 1
- Vision Trigger
4. VISION_POSNUM ID = 1 GI001
- Get Vision Position Count
5. VISION_END ID = 1
- Vision End
6. END

Example: Line 2 establishes camera communication. Line 3 triggers camera to send position data. Line 4 stores position count into GI001. If camera sends 3 positions, GI001=3 after execution.

1.2.4.VISION_POS - Get Vision Position

VISION_POS

参数	值	注释
ID	1	工艺号(1-99)
全局位置变量	GP0001	1-9999整数
附加数据首个变量	GD001	将附加数据保存到变量

示例: VISION_POS ID = 1 GP0001 GD001

确定

取消

Format: VISION_POS [Command] ID=1 [Process No.] GP001 [Global Variable]
I/GI/D/GD/S/GS

[Additional Data Variable].

Function: Stores camera position data into variable.

Parameters:

ID	range【1 ,99】,Process No. in instruction must match vision process screen
Global Position Variable	Camera positions cached sequentially in selected global position variable E.g.: Global position variable is GP0001, Camera sends 2 positions, Get Vision Position instruction stores position data to variable GP0001 . First execution of this instruction GP0001 stores first positionInfo, RunInstruction GP0001 is Position Info
First Additional Data Variable	Variable type: int, float, string This instruction stores defined additional data sequentially into variables, Additional data count must be set in Vision Process > Vision Parameter Settings E.g.: SetAdditional data count= 2, first additional data variable is GD001 This instruction stores additional data into GD001, GD002 Data usage shown later in this chapter as Example Note (writing multiple different additional data and single data)

Example:

1. NOP
2. VISION_RUN ID = 1
Vision Start
3. VISION_TRG ID = 1
Vision Trigger
4. VISION_POS ID = 1 GP0001
Get Vision Position
5. VISION_END ID = 1
Vision End
6. END

Example: Line 2 establishes camera communication. Line 3 triggers camera to send position data. Line 4 stores position into GP0001. If camera sends 3 positions, first execution stores 1st position, second execution stores 2nd position in GP0001.

1.2.5.VISION_CLEAR - Clear Vision Position Data



Format: VISION_CLEAR [Command] ID=1 [Process No.].

Function: Clears position data sent by camera for current process number.

Parameters:

ID	Range [1,99]. Process No. in instruction must match vision process screen selection.
----	--

Example:

1. NOP
2. VISION_RUN ID = 1 Vision Start
3. VISION_TRG ID = 1 Vision Trigger
4. VISION_POSNUM ID = 1 GI001 Get Vision Position Count
5. VISION_POS ID = 1 GP0001 Get Vision Position
6. VISION_CLEAR ID = 1 Clear Vision Position Data
7. VISION_END ID = 1 Vision End
8. END

Example: Line 2 camera communication established. Line 3 triggers camera. Line 4 stores position count into GI001 (e.g. 3 positions, GI001=3, each Get Position reduces count by 1). Line 5 stores position data.

1.2.6.VISION_END - Vision End

工程预览/程序指令/指令插入/参数设定

VISION_END

参数	值	注释
ID	1	工艺号(1-99)

示例: VISION_END ID=1

确定 取消

Format: VISION_END [Command] ID=1 [Process No.].

Function: Ends vision process. Controller disconnects from camera.

Parameters:

ID	Range [1,99]. Process No. in instruction must match vision process screen selection.
----	--

1.2.7.VISION_TRACE - Get Vision Trajectory Position

VISION_TRACE

参数	值	注释
工艺号	1	工艺号
点位存放位置	变量	外部点队列或变量中
点位存放的首个变量	GP0001	更多 第一个点存放的位置
点位存放数	I001	更多 1条轨迹点位总个数
附加数据首个变量	GD001	更多 将附加数据保存到变量

示例: VISION_TRACE ID=1 0 GD001

确定 取消

Format: VISION_TRACE [Command] ID=1 [Process No.];

0/1 [“0” stores positions in external axis motion queue, “1” stores in variables] P/GP [First position variable] I/GI [Position count];

I/GI/D/GD/S/GS [First additional data variable].

Function: When camera sends multiple positions, choose different storage modes to execute trajectory.

Parameters:

Process No.	range【1 , 99】, Process No. in instruction must match vision process screen
Position Storage	Variable, External Point Motion Queue Variable (sent positions stored inVariable) External Point Queue (After camera sends positions , insert external point command after Get Trajectory Position instruction, so program can execute external point trajectory)
First Position Variable (default GP0001)	Cannot modify when storage is external point queue Variable type P,GP,P [],GP[] After camera sends positionsAfter Get Trajectory, positions stored in selected firstVariable, Position storageVariablebased on Select Variablestore E. g. : Camera sends 3 positions, select firstVariable is GP0005 , position data stored inVariable GP0005, GP0006, GP0007
Position Count (default I001)	When Position Storage is external point queue, this parameter cannot be modifiedPosition Count type can select I,GI, I [],GI[]Type Camera position count stored to selectedVariable E. g. : Camera sends 4, Select Position CountVariable is I005, execute Get Trajectory instruction when Variable I005=4
First Additional Data Variable	Variable type: int, float, string This instruction stores defined additional data sequentially into variables, Additional data count must be set in Vision Process > Vision Parameter Settings E. g. : SetAdditional data count= 2 , first additional data variable is GI001 This instruction stores additional data into GI001, GI002

1.2.8.ONFLY_INIT - Flying Capture Start

Flying Capture: When robot tool passes capture point, rapidly triggers camera. During capture, object or camera is in motion.

– Flying Capture Trigger Mode: Capture Point

Format: ONFLY_INIT (Command); ID (Flying Capture Process No.); VISIONID (Bound Vision Process No.); TYPE (Capture Point);

RADIUS (Capture Point Range Radius); DELAY (Record Coordinate Delay);

CAMERA_TRIGGER_DELAY (Camera Trigger Delay).

– Flying Capture Trigger Mode: IO Signal

Format: ONFLY_INIT (Command); ID (Flying Capture Process No.); VISIONID (Bound Vision Process No.); TYPE (IO);

VALUE (Port Value); DELAY (Record Coordinate Delay); CAMERA_TRIGGER_DELAY (Camera Trigger Delay).

Function: Start flying capture task.

– Flying Capture Trigger Mode: Single Axis

Format: ONFLY_INIT (Command); ID (Flying Capture Process No.); VISIONID (Bound Vision Process No.); TYPE (Single Axis);

TRIGGER_START_POS (Trigger Start Position); TRIGGER_COUNT (Trigger Count); ANGLE_SPACING (Trigger Angle Spacing);

DELAY (Record Coordinate Delay); CAMERA_TRIGGER_DELAY (Camera Trigger Delay). Note: Capture point mode only supports linear commands, no smoothing.

Single axis mode does not support linear commands.

Parameters:

Process No.	VisionFlying CaptureProcess No. 1-99 , Process No.Variable. VariableonlysupportsVariable (INT,GINT)
VisionProcesses No.	Enter bound vision process number. Manual or variable. VariableonlysupportsVariable (INT,GINT) Instruction when VisionConnect, not will TriggerCapture; VisionClose when Flying Capture will Close
Flying Capture Trigger	1. Capture Point 2. IO , on (1) or (0)
Capture Point	Capture position. Requires manual teaching and manual assignment to variable.
Capture Pointrange	Capture Point , CaptureTriggerCapture
IO	IO EnterVariable (DIN, DIN[]) TriggerCapture, record is TriggerafterRobot Position
Value	: define in Value. range[0, 1]

	Variable: Define IO port value by assigning to variable.
Record Coordinate Delay	Delay from trigger capture to recording robot coordinates, unit ms

Example: Capture point trigger.

0	开始
> 1	直线 P0001 速度10毫米/秒 平滑0 加速度1 减速度1 0
2	视觉开始 工艺号:1
3	飞拍开始 工艺号:1 视觉工艺号1 拍照点 拍照点范围半径5毫米 记录坐标延时5毫秒 相机触发延时5毫秒
4	直线 P0002 速度10毫米/秒 平滑0 加速度1 减速度1 0
5	飞拍结束 工艺号1
6	视觉结束 工艺号:1
7	结束

Example: IO signal capture.

0	开始
1	直线 P0001 速度10毫米/秒 平滑0 加速度1 减速度1 0
2	视觉开始 工艺号:1
3	飞拍开始 工艺号:1 视觉工艺号1 IO信号 DIN2 端口值B001 记录坐标延时5毫秒 相机触发延时5毫秒
4	直线 P0002 速度10毫米/秒 平滑0 加速度1 减速度1 0
5	飞拍结束 工艺号1
6	视觉结束 工艺号:1
7	结束

Example: Single axis.

Note: Positive angle spacing is forward direction, negative is reverse.

Single axis mode, J4, trajectory from -44 to 44, start position 0, trigger count 3, angle spacing 5. After triggering at 0, next trigger at 5, then 10 – triggers 3 times.

Single axis mode, J4, trajectory from -44 to 44, start position 0, trigger count 3, angle spacing -5. After triggering at 0, next trigger at -5, then -10 – only triggers once since trajectory -44 to 44 is positive direction.

Program Logic: During flying capture trajectory P001 to P002, J4 axis triggers capture when passing 0 degrees.

文件名称: 手册4 页码: 1 /1 当前次数: 1/1

0	开始
> 1	点到点 P0001 速度10% 平滑0 加速度10 减速度10 0
2	视觉开始 工艺号:1
3	飞拍开始 工艺号:1 视觉工艺号1 单轴 J4 触发起始位置0度 触发次数1次 触发角度间隔2 记录坐标延时5毫秒 相机触发延时0毫秒
4	点到点 P0002 速度10% 平滑0 加速度10 减速度10 0
5	飞拍结束 工艺号1
! 6	视觉结束 工艺号:1
7	结束

ONFLY_INIT

参数	值	注释
工艺号	1 更多	1-99
视觉工艺号	1 更多	1-99
飞拍触发方式	单轴	
轴	J4 更多	
触发起始位置	0 更多	度
触发次数	1 更多	
触发角度间隔	2 更多	
记录坐标延时	5 更多	ms
相机触发延时	0 更多	ms

示例: 飞拍开始 工艺号2 视觉工艺号1 单轴 J1 触发起始位置1度 触发次数1次 触发角度间隔2 记录坐标延时1毫秒 相机触发延时10毫秒

1.2.9.ONFLY_END - Flying Capture End

Format: ONFLY_END [Command] ID=1 [Process No.].

Function: End flying capture task.

Can be used standalone. Executing Flying Capture End before Flying Capture Start has no effect. Parameters:

Process No.	Vision Flying Capture Process No., range [1,99] supportsProcess No.Variable (INT,GINT)
-------------	---

Note: Inserting Flying Capture End alone does not cause runtime error. It has no effect if Flying Capture Start was not executed first.

1.2.10. ONFLY_GET_STATUS - Get Flying Capture Status

Format: ONFLY_GET_STATUS [Command] ID=2 [Flying Capture Process No.] STATUS=B001 [Status].

Function: Get whether flying capture task is active, store status into selected variable.

Parameters:

Process No.	Vision Flying Capture Process No., range [1,99] supportsProcess No.Variable (INT,GINT)
Result	getFlying CaptureOpen Status, get to StatusstoreVariable (BOOL,GB00L) “0” Flying CaptureOpen , “ 1” Flying CaptureOpen E. g. : Select Variable is B001,getFlying CaptureOpenStatusInstruction, Flying CaptureOpen B001=1,Flying CaptureOpen B001=0

1.2.11. ONFLY_CALC_TOOL - Flying Capture Calculate Tool

Format: ONFLY_CALC_TOOL [Command] ID=2 [] GRABTOOL=1 [Grab Tool]
NEWTOL=2 [New Tool].

Function: When robot grips workpiece at eccentric position with rotation angle, flying capture calculates and switches to new tool so workpiece center is placed at release position without offset or rotation.

Parameters:

Process No.	Vision Flying Capture Process No., range [1,99] supportsProcess No.Variable (INT,GINT)
Grab Tool	Supports manual tool number or variable (INT,GINT), range [1,999] PickCapture Tool
New Tool	Supports manual tool number or variable (INT,GINT), range [1,999] get to New Tool

Example: Single capture point flying capture tool calculation program.

Program Logic: Switch to tool 1, go to pick position, grip workpiece and pass capture point, trigger camera to capture and return position while controller records current position, flying capture tool calculation reads data to calculate new tool 2, switch to tool 2, place workpiece center at release position via offset without offset or rotation.

- i. Robot grips workpiece center – calculated tool 2 should match tool 1;
- ii. Robot grips workpiece at eccentric position, no rotation – after calculating and switching to tool 2, workpiece center placed without offset;
- iii. Robot grips workpiece at eccentric position with rotation angle – after calculating and switching to tool

2, workpiece center placed at release position without offset or rotation.

Note: If camera has not returned position data when executing flying capture tool calculation, error: Tool calculation error will occur.

All points below must be set in Work File Variables screen by selecting tool and writing current position, otherwise executed without tool. P0004 point tool must be manually set to 0.

```
NOP
SWITCHTOOL (1) #Switch to calibrated tool
MOVJ P0001 VJ = 80 % PL = 0 ACC = 80 DEC = 80 0
#WorkpiecePick
VISION_RUN ID = 1
#VisionStart
ONFLY_INIT ID = 1 VISIONID = 1 TYPE = Point GP0001 RADIUS = 1 DELAY
= 1.9 CAMERA_TRIGGER_DELAY = 0
#Flying CaptureStart, TriggerCapture Point GP0001 1 record 1.9
Camera when 0
MOVL P0002 V = 1000 mm/s PL = 0 ACC = 100 DEC = 100 0
#Flying CaptureTrajectory
MOVL P0003 V = 1000 mm/s PL = 0 ACC = 100 DEC = 100 0
#Flying CaptureTrajectoryEnd
ONFLY_CALC_TOOL ID = 1 GRABTOOL = 1 NEWTOOL = 2
#Flying CaptureTool Tool 2
SWITCHTOOL (2)
#Tool 2
MOVL P0004 V = 1000 mm/s PL = 0 ACC = 100 DEC = 100 0
#Workpiece in Place to ReleasePosition
ONFLY_END ID = 1
#Flying CaptureEnd
VISION_END ID = 1
#VisionEnd
END
```

1.3. Vision Process Examples

1.3.1.Example 1: Camera captures 10 times, then vision closes



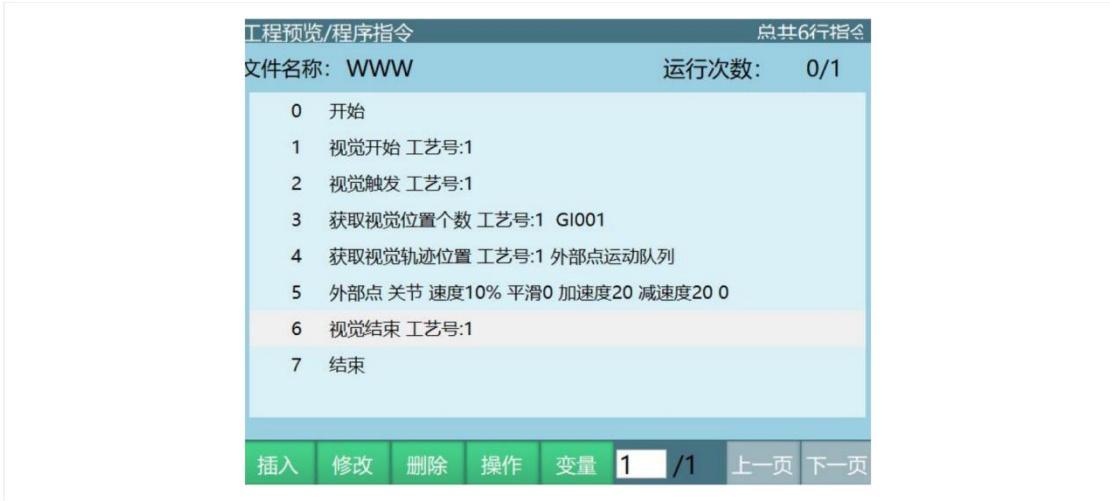
1.3.2.Example 2: Camera sends continuous trajectory positions

When camera sends continuous trajectory positions, use Get Trajectory Position instruction. The following example uses a network

debugging tool to send positions. Vision communication established and positions sent. Sent positions are for illustration only.



1. Get trajectory position - store in external point



2. Store positions in variables

For example: First variable is P0001, network tool sends 5 positions.
Program shown below:



1.3.3.Example 3: Additional data parameter description for tracking position

1. Vision Process > Vision Parameter Settings: Set additional data count, max 10 supported.

工艺/视觉工艺/视觉参数设置

相机选择
工艺号: 1 类型: 自定义

用户坐标系
手眼标定用户坐标系: 无

网络参数
控制器IP: 192.168.0.233
相机: 客户端
端口数: 1 相机数据: 机器人坐标
端口1: 5050 端口2: 0

触发方式
Ethernet
TRG
单次 触发周期: 35 ms

连接参数
帧头: @ 成功发送标志符: OK
分隔符: , 失败发送标志符: NG
结束符: \$ 超时时间: 30 s
单目标: ☒ 类型: 2D
附加数据个数: 2

接收坐标系
工具 ☐ 用户 ☒
弧度/角度
弧度/角度转换: 角度制

2. Click Vision Position Parameters to view example format. "data" represents additional data.

工艺/视觉工艺/视觉位置参数设置

工艺号: 1

机器人抓取时的姿态
(坐标系都是直角坐标系)

高度可手动填写

基准点	值	相机坐标	值
X值	0.00	X	0.00
Y值	0.00	Y	0.00
Z值	0.00	高度	0.00
A值	3.14	角度	0.00
B值	0.00		
C值	0.00		

偏移补偿
X方向偏移: 0.000 mm
Y方向偏移: 0.000 mm
Z方向偏移: 0.000 mm
角度偏移: 0.000

接受位置类型: 点
角度方向: 正方向
比例系数: 1.000

示例格式: @,x,y,Rz,data ..., \$
接收数据:

返回 修改

标定抓取姿态
运行到基准点
清除标定
试拍照
运行到该点

3. External device: Network debugging tool sends positions according to example format in Vision Position Parameters.



4. Click Test Capture. After network tool sends positions, "1,2" in received data represents additional data. Additional data defaults to GS001, incrementing based on defined count.



5. If Get Conveyor Tracking Position instruction is inserted, execution stores additional data values into user-selected variables (int, float, string). Users can represent workpiece shape/color with values (e.g. GD001=1 for red, GD002=2 for green), then use conditional logic to track and pick different colored workpieces.

文件名称: 传送带1	页码: 1 /2	当前次数: 1/1
0	开始	
1	直线 P0001 速度10毫米/秒 平滑0 加速度1 减速度1 0	
2	传送带工件检测开始 工艺号 1	
3	循环 当(I001 == 0)时	
4	传送带跟踪开始 P0002 工艺号 1 速度10毫米/秒 加速度20	
5	获取传送带跟踪位置 工艺号 1 GP0001 GD001	
6	如果 (GD001 == 1)	
7	直线 P0005 速度10毫米/秒 平滑0 加速度1 减速度1 0	
8	直线 P0006 速度10毫米/秒 平滑0 加速度1 减速度1 0	
9	结束如果	
10	如果 (GD001 == 2)	

11	直线 P0007 速度10毫米/秒 平滑0 加速度1 减速度1 0	
12	直线 P0008 速度10毫米/秒 平滑0 加速度1 减速度1 0	
13	结束如果	
14	延时1秒	
15	传送带跟踪结束 工艺号 1	
16	结束循环	
17	传送带工件检测结束 工艺号 1	
18	结束	

1.3.4.Example 4: After camera captures workpiece image, send data to robot for picking

As shown:

文件名称: 视觉	页码: 1 /2	当前次数: 1/1
0	开始	
> 1	赋值 I001 = 0	
2	##安全点\$\$	
3	点到点 P0001 速度10% 平滑0 加速度10 减速度10 0	
4	视觉开始 工艺号:1	
5	循环 当(I001 == 0)时	
6	视觉触发 工艺号:1	
7	获取视觉位置个数 工艺号:1 GI001	
8	循环 当(GI001 > 0)时	
9	获取视觉位置 工艺号:1 GP0001 GD001	
10	##获取到的视觉位置点位\$\$	
11	点到点 GP0001 速度10% 平滑0 加速度10 减速度10 0	
12	##工件放置位置\$\$	
13	点到点 P0010 速度10% 平滑0 加速度10 减速度10 0	
14	结束循环	
15	结束循环	
16	清除视觉位置信息 工艺号:1	
17	视觉结束 工艺号:1	
18	结束	