

# SICK AG

## PLOC2D Provider

Version 1.0.0

### User's guide

July 16, 2019

NOTES:



## Contents

1. Introduction.....	4
2. Overview of the provider .....	5
2.1. Installation .....	5
2.2. Method and property .....	6
2.2.1. CaoWorkspace::AddController method .....	6
2.2.2. CaoController:: GetVariableNames method .....	6
2.2.3. CaoController::AddVariable method .....	7
2.3. Variable list .....	7
2.3.1. Controller class.....	7
3. Command reference.....	9
3.1. Controller class.....	9
3.1.1. CaoController::Execute("LocateTargets", " jobNbrs") command.....	9
3.1.2. CaoController::Execute("SelectTarget", "job,index") command .....	9
3.1.3. CaoController::Execute("Next") command.....	10
3.1.4. CaoController::Execute("Prev") command.....	10
3.1.5. CaoController::Execute("Align") command .....	10
3.1.6. CaoController::Execute("Restart") command .....	11
4. Sample program.....	12
5. Error Code.....	14

## 1. Introduction

PLOC2D provider is an ORiN2 CAO provider that allows the access to variables provided by PLOC2D series based on InspectorP Device for locating object on a 2D surface. With simple API command, its possible to share located the Object-Coordinates with the robot.

This document describes the overview of the PLOC2D provider and the CAO interface (function specifications) implemented in the provide

## 2. Overview of the provider

### 2.1. Installation

PLOC2D provider module comprises the following DLLs. You do not need to perform this step if the provider has been installed with the ORiN2 SDK installer. If you install the provider manually, refer to the Table 2-1.

**Table 2-1 PLOC2D provider**

File name	CaoProvPLOC2D.dll
ProgID	CaoProv.SickAG.PLOC2D
Registry registration	regsvr32 CaoProvPLOC2D.dll
Registry un-registration	regsvr32 /u CaoProvPLOC2D.dll

## 2.2. Method and property

This section contains how to load a SICK device provider.

### 2.2.1. CaoWorkspace::AddController method

**Syntax** AddController ( <bstrCtrlName:BSTR>, <bstrProvName:BSTR>, <bstrPcName:BSTR>, [**<bstrOption:BSTR>**] )

Parameter	Data type	Explanation
bstrCtrlName	BSTR	[in] Controller name
BstrProvName	BSTR	[in] Provider name. Fixed value = "CaoProv.SickAG.PLOC2D"
bstrPcName	BSTR	[in] Computer name that the provider runs.(unused)
bstrOption	BSTR	[in] Specify the following options by using ", (comma)" for separation. <ul style="list-style-type: none"> <li>• Device={Sick Device} (e.g. Device=Inspector)</li> <li>• IP={Host/IP Address } (e.g. IP=192.168.0.1)</li> <li>• Port={port number} (e.g. Port=1234)</li> <li>• Timeout={Time ms} (e.g. Timeout=2000)</li> </ul>

### 2.2.2. CaoController:: GetVariableNames method

Obtain a variable list that is defined in the connected device (see AddController, Device option)

**Syntax** GetVariableNames ( [**<bstrOption:BSTR>**] )

Parameter	Data type	Explanation
bstrCtrlName	BSTR	[in] Option string(unused) All variable names depending on connected device type

### 2.2.3. CaoController::AddVariable method

Add CaoVariable objects used for the access to variables. The variable-Set differs depending on connected device type.

**Syntax** AddVariable ( <bstrName:BSTR>, [<bstrOption:BSTR>] )

Parameter	Data type	Explanation
bstrName	BSTR	[in] Specify any variable names.
bstrOption	BSTR	[in] Specify the following options by using ", (comma)" for separation. (unused)

## 2.3. Variable list

This document describes the PLOC2D Provider for using the PLOC2D Device (see AddController Device Option)

### 2.3.1. Controller class

Show a list of implemented controller class variables.

**Table 2-2 Controller class system variable list**

Variable identifier	Data type	Explanation	Attribute	
			get	put
@X	VT_R8	X position of the current target	<input type="radio"/>	<input type="checkbox"/>
@Y	VT_R8	Y position of the current target	<input type="radio"/>	<input type="checkbox"/>
@Z	VT_R8	Z position of the current target (no used, in simple case)	<input type="radio"/>	<input type="checkbox"/>
@RX	VT_R8	X- Rotation of the current target (no used, in simple case)	<input type="radio"/>	<input type="checkbox"/>
@RY	VT_R8	Y- Rotation of the current target (no used, in simple case)	<input type="radio"/>	<input type="checkbox"/>
@RZ	VT_R8	Z- Rotation of the current target	<input type="radio"/>	<input type="checkbox"/>
@MATCH	VT_R8	Index of the current target of related coordinates	<input type="radio"/>	<input type="checkbox"/>
@MATCHES	VT_R8	Count of targets after last locating request	<input type="radio"/>	<input type="checkbox"/>

@SCALE	VT_R8	Target scale in relation of reference shape	<input type="radio"/>	
@SCORE	VT_R8	Quality of target detection in percent	<input type="radio"/>	
@TIME	VT_R8	Execution duration for locating targets	<input type="radio"/>	
@LASTERRORCAM	V_R8	Cam error codes (see Cam user's manual)	<input type="radio"/>	
@MAKER	BSTR	Information about the provider author( <b>default device variable</b> )	<input type="radio"/>	
@VERSION	BSTR	Version of the provider ( <b>default device variable</b> )	<input type="radio"/>	
@LASTERRORDLL	BSTR	A error description as flowing text ( <b>default device variable</b> )	<input type="radio"/>	

## 3. Command reference

Describe explanation for implemented Execute commands.

### 3.1. Controller class

**Table 3-1 CaoController::Execute command list**

Command	Description	Page
LocateTargets	Locate all targets with given job ids	
SelectTarget	Select target (coordinates) after locating the targets	
Next	Forward-iteration through targets after locating	
Prev	Backward- iteration through targets after locating	
Align	Align the camera	
Restart	Restart camera	

#### 3.1.1. CaoController::Execute("LocateTargets", "jobNbrs") command

**Syntax** <Command name> ( <bstrSyntax>, [<bstrOption>] )

< bstrSyntax > : [in] Syntax (VT\_BSTR)

<bstrOption> : [in] BSTR option

jobNbrs	Job ids , separated with ,
---------	----------------------------

Return value : [out] Return 0 if success

#### Example

---

```
' Example program code
Call Ctrl.Execute("LocateTargets","1") 'For one job (1)
Call Ctrl.Execute("LocateTargets","1,2") 'For more job (1+2)
```

---

#### 3.1.2. CaoController::Execute("SelectTarget", "job,index") command

**Syntax** <Command name> ( <bstrSyntax>, [<bstrOption>] )

< bstrSyntax > : [in] Syntax (VT\_BSTR)

<bstrOption> : [in] BSTR option (VT\_BOOL)

job	Job id
Index	Index of target , for given job

Return value : [out] Return 0 if success

#### Example

---

```
' Example program code
Call Ctrl.Execute("SelectTarget","7,1") 'Get coordinates for target with index 1 for job 7
Call Ctrl.Execute("SelectTarget","6,2") 'Get coordinates for target with index 2 for job 6
```

---

### 3.1.3. CaoController::Execute("Next") command

**Syntax** <Command name> ( <bstrSyntax>, [<bstrOption>] )

< bstrSyntax > : [in] Syntax (VT\_BSTR)  
 <bstrOption> : [in] BSTR option (unused)

Return value : [out] Returns 0 if success

#### Example

---

```
' Example program code
Call Ctrl.Execute("Next") 'After calling, the coordinate variable contains date for next target
```

---

### 3.1.4. CaoController::Execute("Prev") command

**Syntax** <Command name> ( <bstrSyntax>, [<bstrOption>] )

< bstrSyntax > : [in] Syntax (VT\_BSTR)  
 <bstrOption> : [in] BSTR option (unused)

Return value : [out] Returns 0 if success

#### Example

---

```
' Example program code
Call Ctrl.Execute("Next") 'After calling, the coordinate variable contains date for prev target
```

---

### 3.1.5. CaoController::Execute("Align") command

**Syntax** <Command name> ( <bstrSyntax>, [<bstrOption>] )

< bstrSyntax > : [in] Syntax (VT\_BSTR)  
<bstrOption> : [in] BSTR option (unused)  
  
Return value : [out] Returns 0 if success

**Example**

---

' Example program code  
Call Ctrl.Execute("Align") 'After calling, the coordinate variable contains the new alignment

---

**3.1.6. CaoController::Execute("Restart") command**

**Syntax** <Command name> ( <bstrSyntax>, [<bstrOption>] )

< bstrSyntax > : [in] Syntax (VT\_BSTR)  
<bstrOption> : [in] BSTR option (unused)  
  
Return value : [out] Returns 0 if success

**Example**

---

' Example program code  
Call Ctrl.Execute("Restart") 'After calling, the camera is restarting

---

## 4. Sample program

### List 4-1 SampleCode.pcs

```
'!TITLE "Robot program"

Sub Main
    TakeArm Keep = 0

    Dim ctrl as object ' cam provider handle

    Dim machtes as integer 'count of matches
    Dim it as integer ' iterator
    Dim pos as position 'position of selected target

    'Init Cam
    ctrl = Cao.AddController("Sick1", "CaoProv.SickAG.PLOC2D", "", "@EventDisable=
false,Device=Inspector,IP=192.168.0.1,Port=1223,Timeout=2000")

    'Init needed variables
    Call ctrl.AddVariable("@X", "")
    Call ctrl.AddVariable("@Y", "")
    Call ctrl.AddVariable("@RZ", "")
    Call ctrl.AddVariable("@MATCHES", "")

    ' locate all targets for job 7 an get match count
    Call ctrl.Execute("LocateTargets", "7") ' Job 7
    machtes = ctrl.Variables("@MATCHES").Value ' get match count

    ' iterate all matches
    For it = 1 to machtes

        'get target coordinate by selecting
        Call ctrl.Execute("SelectTarget", "7," & it)

        ' map data to pos
```

```
pos = P(ctrl.Variables("@X").Value, ctrl.Variables("@Y").Value, 0, 0, 0,  
ctrl.Variables("@RZ").Value)
```

```
' do anything with postion (e.g. move robot)  
'Move L, @@ pos
```

```
next
```

```
'relese cam provider  
call ctrl.Variables.Remove(ctrl.Variables("@X").Index)  
call ctrl.Variables.Remove(ctrl.Variables("@Y").Index)  
call ctrl.Variables.Remove(ctrl.Variables("@RZ").Index)  
call ctrl.Variables.Remove(ctrl.Variables("@MATCHES").Index)  
cao.Controllers.Remove ctrl.Index
```

```
End Sub
```



## 5. Error Code

Error Code	Description
ExecuteFunction('Unknown error while execute')	Execute Command name not found
SelectTarget('parse match')	Could not parse match to number in option string
SelectTarget('parse job id')	Could not parse job-id to number in option string
SelectTarget('index out of range')	Target number must be >1 or < matches-count
SelectTarget('request data failed')	Could not request Camera for select target
LocateTargets('request data failed')	Could not request Camera for "LocateTargets"
Align('bad response')	Response from camera invalid
Align('request data failed')	Could not request Camera for "Align"
SystemRestart('bad response')	Response from camera invalid
SystemRestart('request data failed')	Could not request Camera for "SystemRestart"
Socket('send data failed')	Cannot send data over socket (no connection?)
Socket('receive data failed')	Cannot receive data from socket (no connection?)
Socket('connection refused')	Connection refuse on socket
SetVariables('Result_BadRequest')	Error Parsing variable in response string
LocateTargets('Invalid Request parameter count')	Request Parameter invalid (see Command spec)
SelectTarget('Invalid Request parameter count')	Request Parameter invalid (see Command spec)
Next('no more matches')	All Targets iterate max matches count reached
Prev('no more matches')	All Targets iterate min matches count reached
UnknownDevice('device or, 'Device' option not set')	Device invalid ( <b>only Inspector available in this version</b> )