KEYENCE

LaserMarker provider

Version 1.0.0

User's Guide

May 13, 2019

Remark

Revision history

Version	Date	Content
1.0.0	2019-05-13	First edition

Compatible device

Model	Version	Note
MD-X1000/1500 series		
MD-F3200/5200 series		
MD-U1000 Series		
ML-Z9600 Series		

Table of contents

1. Introduction

This document is a user's guide for CAO providers that can be connected to laser markers (compatible models: MD-X1000/1500 Series, MD-F3200/5200 Series, MD-U1000 Series, and ML-Z9600 Series) manufactured by KEYENCE Corporation. The CAO provider (CaoProvKEYENCELaserMarker.dll) handled in this document is called the LaserMarker provider.The LaserMarker Provider is the "LaserMarker Communication Interface User Manual for MD-X1000/1500 Series MD-U1000 Series ML-Z9600 Series" (hereinafter referred to as the LaserMarker Communication Manual) supplied by KEYENCE Corporation. And we are developing it with reference to This document describes the features of LaserMarker providers and the methods implemented.

2. Provider overview

2.1. Overview

Fig. 2-1 shows the correspondence between classes of providers and laser markers. CaoProvController corresponds to the laser marker body, and CaoProvVariable corresponds to the settings of laser markers such as the number of programs to be printed and operating hours.

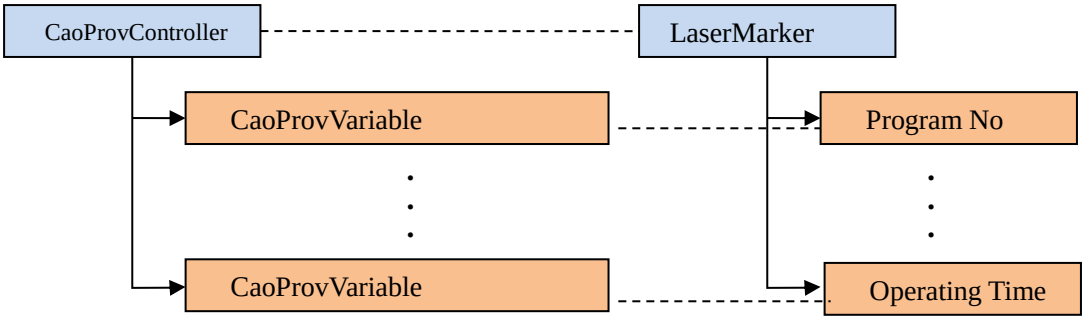

Fig. 2-1 Correspondence Diagram of Classes of Providers and Laser Markers

Fig. 2-2 shows the connection between a provider and a laser marker. Connect the laser markers you want to connect to using two types of connection methods: Ethernet and COM.

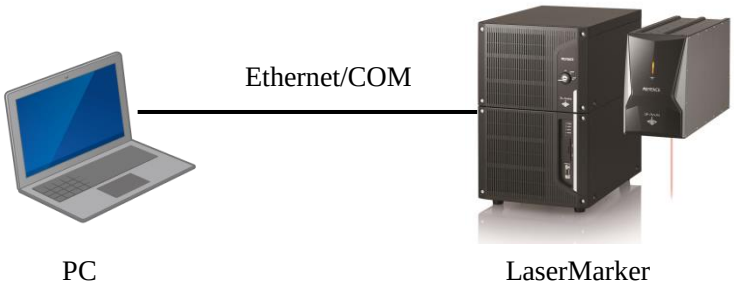

Fig. 2-2 Connection Diagram between Provider and Laser Marker

The file format of the LaserMarker provider is DLL (Dynamic Link Library), which is dynamically loaded from the CAO engine when it is used. In order to use the LaserMarker provider, it is necessary to install ORiN2SDK or to perform registry registration manually by referring to Table 2-1.

Table 2-1 LaserMarker providers' file formats

File name	CaoProvKEYENCELaserMarker.dll
ProgID	CaoProv.KEYENCE.LaserMarker
Registry registration[1]	Regsvr32 CaoProvKEYENCELaserMarker.dll
Unregistering the registry	Regsvr32 /u CaoProvKEYENCELaserMarker.dll

[1] You do not need to manually register/delete the ORiN SDK installations.

2.2. Method Properties

2.2.1. CaoWorkspace::AddController method

Connection parameters are set in the CaoWorkspace::AddController and communication connection is made. The AddController specifications are as follows:

Format

```
AddController(
        BSTR bstrCtrlName,      //[in]Controller name(Please enter any controller name)
        BSTR bstrProvName,      //[in]Provider name. Fixed value="CaoProv.KEYENCE.LaserMarker "
        BSTR bstrPcName,        //[in]Provider Execution Machine Name
        BSTR bstrOption         //[in]Option string
);
```

Table 2-2 Option string of CaoWorkspace::AddController

Setting items	Option String	Required	Description
Destination	Conn	✓	The description method differs between the case of Ethernet connection and the case of COM connection. Please see 2.2.1.1Conn option
Time-out	Timeout	-	Specify a timeout between 1 and 4294967295 ms. Defaults:3000 ms

2.2.1.1. Conn option

<Ethernet Connections Description Format>

"eth:< A >:[< B >[:< C >[:< D >]]]"[2]

Parameter	Meaning	Required	Valid value
A	Destination IP address	✓	192.168.0.20 etc.
B	Destination port No.	-	1~65535 Default value: 50002
C	Own IP adress	-	192.168.0.1 etc. Default value: Automatic specification
D	Own Port No.	-	1~65535 Default value: Automatic specification

[2] Parameters in braces ("[]") are optional.

Example 1) Connecting to IP address 192.168.0.20 and port number 50002

"Conn = eth:192.168.0.20", "Conn = eth:192.168.0.20:50002"

Example 2) Connection with IP address 192.168.0.20, port number 50002, timeout 1000 ms

"Conn = eth:192.168.0.20, Timeout = 1000"

"Conn = eth:192.168.0.20:50002, Timeout = 1000"

<COM Connection Description Format>

"com:< A >:[< B >[:< C >:< D >:< E >[:< F >]]]"[2]

Parameter	Meaning	Required	Valid value (default value: underscore)
A	COM port number	✓	1~256
B	Baud rate [bps]	-	4800/9600/19200/38400/57600/115200
C	Parity-check	-	N ... No parity E ... Even (Even) parity O ... Odd parity
D	Data bit [bit]	-	7/8
E	Stop bit [bit]	-	1/2
F	Flow control	-	0 ... No flow control 1 ... XON/XOFF flow control 2 ... Hardware controlling

Example 1) Connecting to the COM1 with defaults

"Conn = com:1" or "Conn = com:1:38400:N:8:1:0"

Example 2) Connecting to the COM1 with a timeout of 1000 ms by defaults

"Conn = com:1,Timeout=1000" or "Conn = com:1:38400:N:8:1:0,Timeout = 1000"

Usage examples (CaoScript)

```
Dim ctrl
Set ctrl = cao.AddController("LaserMarker", "CaoProv.KEYENCE.LaserMarker", "",
                                                    "Conn = eth:192.168.0.20:50002,Timeout=1000")
```

2.2.2. CaoController::get_VariableNames Properties

Gets lists of variables that are available to LaserMarker providers in Table 2-3.

Return Values Type	Description
VT_ARRAY | VT_BSTR	You can get a list of variables that can be used for AddVariable variable names in the CaoController.

Usage examples (CaoScript)

```
Dim vntNamesList
VntNamesList = dat.ToVar(ctrl.VariableNames)
Dbg.output vntNamesList(0)          'Display example:@MAKER_NAME
Dbg.output vntNamesList(1)          'Example result:@VERSION
```

Table 2-3 CaoController Classes System Variables List

Variable name	Data type		Description	Attribute	
				Get	Put
@MAKER_NAME	VT_BSTR		Returns the manufacturer name "KEYENCE".	✓	-
@VERSION	VT_BSTR		Returns the version of the provider.	✓	-
@READY	VT_UI1		Returns whether the laser marker is ready for printing. 0 ... Print-ready status 1 ... Error occurring 2 ... Printing or sending program	✓	-
@PROGRAM_NO	VT_UI2		You can acquire or change the program number during operation. Value range: 0 to 1999	✓	✓
@ERROR	VT_ARRAY | VT_VARIANT		Returns the error status and error number.	✓	-
	0	VT_BOOL	Error condition false... No error true... Error present		

	1	VT_BSTR	The error number that has occurred. If there are multiple errors, display them as a comma-separated string. Examples) "E001,E002,W100,T000" * Refer to the user's manual for the laser marker body for details of the error. Error number: E001~E400/W100~W200/T000~T100		
@ALL_POSITION	VT_ARRAY | VT_R8		Changes/acquires the installation position correction of the device settings. The value range of X to Z coordinate correction depends on the model used. Refer to LaserMarker Communication Manual.	✓	✓
	0	VT_R8	X Rotation Angle [°] Value range:-90.000 to 90.000		
	1	VT_R8	Y Rotation Angle [°] Value range:-90.000 to 90.000		
	2	VT_R8	θ angle [°] Value range:-180.000 to 180.000		
	3	VT_R8	X coordinate correction [mm] Value range: Model dependent		
	4	VT_R8	Y coordinate correction [mm] Value range: Model dependent		
	5	VT_R8	Z coordinate correction [mm] Value range: Model dependent		
@TIME	VT_DATE		Changes/acquires the time of the built-in clock. Value range: 2000/1/100:00:00 to 2099/12/31/23:59:59 Format year/month/date/time:minutes:seconds * Insert a half-pitch space between the date and time. Example) "2000/1/100:00:00"	✓	✓
@POWER_OFFSET	VT_R4		Change/acquire the laser power offset amount [%] in the Instrument Settings. Value range:-100.0 to 100.0 * If the power offset applied to the laser power of the block exceeds 100%, it is fixed at 100%. * The set value applies to all programs. * It does not apply to 0% laser power blocks.	✓	✓

@OPERATING_TIME	VT_UI4		Get the cumulative operating time [h] for the laser marker. Value range: 0 to 99999	✓	-
@LASER_OPERATING_TIME	VT_UI4		Acquire laser activation time [h]. Value range: 0 to 99999 *Only MD-U1000 series and ML-Z9600 series are available.	✓	-
@SCANNER_OPERATING_TIME	VT_UI4		Acquires the scanner operating time [h]. Value range: 0 to 99999 *Only MD-U1000 series and ML-Z9600 series are available.	✓	-
@SHUTTER_OPERATING_TIME	VT_UI4		Get the number of shutters. Value range: 0 to 4294967295 *Only MD-U1000 series and ML-Z9600 series are available.	✓	-
@CONTACTOR_OPERATING_TIME	VT_UI4		Gets the number of contactor operations. Value range: 0 to 4294967295	✓	-
@MARKING_UNIT_TEMPERATURE	VT_R4		Get the head temperature [°C]. Value range:-999.9 to 999.9 *Only MD-U1000 series is available.	✓	-
@CONTROLLER_TEMPERATURE	VT_R4		Get the controller temperature [°C]. Value range:-999.9 to 999.9 *Only MD-U1000 series is available.	✓	-
@LASER_POWER_CALIBRATION_RESULT	VT_ARRAY | VT_VARIANT		Acquires whether laser power calibration is performed and the result [W]. *Only MD-U1000 series is available.	✓	-
	0	VT_BOOL	Whether to perform calibration False: None True: Yes		
	1	VT_R4	Calibration result [W] Value range: 0 to 999.9		
@CUMULATIVE_MARKING_COUNT	VT_ARRAY | VT_UI4		Changes/acquires the cumulative print count of 1,2 [times].	✓	✓
	0	VT_UI4	Cumulative print count 1 [times] Value range: 0 to 4294967295		
	1	VT_UI4	Cumulative print count 2 [times]		

		Value range: 0 to 4294967295		
@DESICCANT_EXPIRATION_DATE	VT_DATE	Get the Head Drying Agent Replacement Deadline. Value range: 2000/1/1 to 2099/12/31 *Available only in MD-U1000 series.	✓	-
@SAFETY_SHUTTER_A_OPERATION_COUNT	VT_UI4	Acquires the number of opening and closing times of Safety Shutter A. Value range: 0 to 4294967295 *Available only in ML-Z9600 series.	✓	-
@SAFETY_SHUTTER_B_OPERATION_COUNT	VT_UI4	Acquires the number of opening and closing times of Safety Shutter B. Value range: 0 to 4294967295 *Available only in ML-Z9600 series.	✓	-
@MARKING_LASER_OSCILLATOR_TEMPERATURE	VT_R4	Get the laser oscillator tube temperature [°C]. Value range:-999.9 to 999.9 *Available only in ML-Z9600 series.	✓	-

2.2.3. CaoController::AddVariable method

Generates CaoVariable objects from a CaoController. For option strings, enter blanks when using system variables, and enter character strings corresponding to each variable name in Table 2-4 when using user variables.

Format

```
AddVariable (
        BSTR bstrVarName,       //[in] Variable name
        BSTR bstrOption         //[in] Option string
);
```

Usage examples (CaoScript)

```
Dim varCumulativeMarkingCount
Set varCumulativeMarkingCount = ctrl.AddVariable("@CUMULATIVE_MARKING_COUNT","")
Dim varString
Set varString = ctrl.AddVariable("String_001", " PRG=1999,BLK=255")
```

Table 2-4 CaoController Classes User Variables List

Variable name	Data type		Description	Attribute	
				Get	Put
Couter_*	VT_ARRAY | VT_UI4		By specifying the program number and the counter number in the bstrOption at the time of AddVariable, the current value of the counter and the number of repetitive prints can be changed/geted. Specify the bstrOption in the format "PRG=(program No), CTR=(counter No)". Program No: 0 to 1999 Counter No.: 0 to 9/A to J Example) "PRG=0,CTR=0", "PRG=1999,CTR=J" Enter a unique string in the variable name "Counter_*" so that you can avoid doing anything else. Example) "Counter_001", "Counter_test"	✓	✓
	0	VT_UI4	Current value of the counter Value range: 0 to 4294967295		
	1	VT_UI4	Number of repetitive prints Value range: 0 to 4294967295		
Marked_*	VT_BSTR		By specifying the program No. and the block No. in the bstrOption at the time of AddVariable, the final printed character strings printed by laser markers can be geted. Specify the bstrOption in the format "PRG=(program No), BLK=(block No)". Program No. value range: 0 to 1999 Block No. value range: 0 to 255 Example) "PRG=0,BLK=0", "PRG=1999,BLK=255" Enter a unique character string in the "~" of the variable name so that it does not depend on other characters. Example) "Marked_001", "Marked_test"	✓	-
String_*	VT_BSTR		You can change/acquire text strings or logos/photo files	✓	✓

		by specifying the program number and the block number in the bstrOption for AddVariable. Specify the bstrOption in the format "PRG=(program No), BLK=(block No)". Program No. value range: 0 to 1999 Block No. value range: 0 to 255 Example) "PRG=0,BLK=0", "PRG=1999,BLK=255" Enter a unique character string in the "~" of the variable name so that it does not depend on other characters. Example) "String_001", "String_test"		

2.2.4. CaoVariable::get_Value Properties

Gets the status of variables created by the AddVariable method. The type of return code depends on the variables named in AddVariable. For the return value of each variable, refer to the system variable in Table 2-3 or the user variable in Table 2-4.

Usage examples (CaoScript)

```
Dim vntCumulativeMarkingCount
vntCumulativeMarkingCount = DAT.ToVar (varCumulativeMarkingCount.Value)
Dbg. output vntCumulativeMarkingCount(0)        'Display example: 100
Dbg. output vntCumulativeMarkingCount(1)        'Display example: 100
Dbg.output varString                            'Display example: test
```

2.2.5. CaoVariable::put_Value Properties

Sets the values of variables created by the AddVariable method. The "put_Value" property can be used for the variables in Table 2-3 and Table 2-4, where "✓" is added to the "put".

Usage examples (CaoScript)

```
varCumulativeMarkingCount.Value = Array(0,0)
varString = "test2"
```

2.2.6. CaoController::Execute method

Executes provider-specific commands that belong to CaoController classes. The arguments of the Execute method specify the command as a BSTR type and the parameters as a VARIANT type. The parameters are optional.

Format

```
Execute (
        BSTR bstrCmd            //[in] command name
        [,VARIANT vntParam]     //[in] parameter
);
```

Argument	Description
bstrCmd	Specify from the command list of the Execute method in Table 2-5.
vntParam	Specify the parameters to use in the command name. For command names that begin with "Set", parameter settings are mandatory.

Table 2-5 List of CaoController::Execute method commands

Command name	Description	Compatible model			p.
		MD-X[3] MD-F[4]	MD-U[5]	ML-Z[6]	
GetReady	Get whether the laser marker is ready for printing.	✓	✓	✓	17
StartMarking	Printing starts.	✓	✓	✓	18
GetStartMarking	Acquires the print result of asynchronous processing.	✓	✓	✓	18
Check2DCode2	Scan the two-dimensional code with the built-in camera of the laser marker. (MD-X 1000 L /1500 L series is not supported.)	✓MD-X - MD-F	✓	-	19
GetProgramNo	Gets the running program number.	✓	✓	✓	20
SetProgramNo	Changes the running program number.	✓	✓	✓	21
GetCounterNo	Gets the current value of the specified counter number and the number of repetitive prints.	✓	✓	✓	21
SetCounterNo	Gets the current value of the specified counter	✓	✓	✓	22

[3] MD-X 1000/1500 series
[4] MD-F 3200/5200 series
[5] MD-U1000 series
[6] ML-Z9600 series

	number and the number of repetitive prints.				
GetMarkedCharacter	Gets the final print string for the specified block number.	✓	✓	✓	22
GetError	Get the errors that occur in laser markers.	✓	✓	✓	23
ClearError	Clears errors that occur in laser markers.	✓	✓	✓	23
GetAllPosition	Acquires the device installation position correction value.	✓	✓	✓	24
SetAllPosition	Changes the device installation position correction value.	✓	✓	✓	25
GetTime	Gets the time of the device.	✓	✓	✓	25
SetTime	Changes the time of the device.	✓	✓	✓	26
GetPowerOffset	Get laser power offset.	✓	✓	✓	26
SetPowerOffset	Change the laser power offset.	✓	✓	✓	26
GetCharacterString	Gets the character string and logo file of the specified block number.	✓	✓	✓	27
SetCharacterString	Changes the character string and logo file of the specified block number.	✓	✓	✓	27
GetOperatingTime	Get the cumulative operating time for the laser marker.	✓	✓	✓	28
GetLaserOperatingTime	Acquire laser activation time.	-	✓	✓	28
GetScannerOperatingTime	Acquires the operating time of the scanner.	-	✓	✓	28
GetShutterOperatingCount	Gets the number of shutter operations.	-	✓	✓	29
GetContactorOperatingCount	Gets the number of contactor operations.	-	✓	-	29
GetMarkingUnitTemperature	Get the head temperature.	-	✓	-	29
GetControllerTemperature	Acquire the controller temperature.	-	✓	-	30
GetLaserPowerCalibrationResult	Acquires laser power calibration results.	-	✓	-	30
GetCumulativeMarkingCount	Acquires the cumulated number of prints, 1 and 2.	✓	✓	✓	31
SetCumulativeM	Changes the cumulative number of prints 1 and 2.	✓	✓	✓	32

arkingCount					
GetDesiccantExpirationDate	Get the Head Drying Agent Replacement Deadline.	-	✓	-	32
GetSafetyShutterAOperatingCount	Acquires the number of opening and closing times of Safety Shutter A.	-	-	✓	32
GetSafetyShutterBOperatingCount	Acquires the number of opening and closing times of Safety Shutter B.	-	-	✓	33
GetMarkingLaserOscillatorTemperature	Acquire laser tube temperature.	-	-	✓	33
StartRetryMarking	Printing is repeated until the print quality exceeds the set value. (MD-X 1000 L /1500 L series is not supported.)	✓MD-X - MD-F	✓	-	34
GetRetryMarking	Gets the result of a StartRetryMarking. (MD-X 1000 L /1500 L series is not supported.)	✓MD-X - MD-F	✓	-	37
SendCommand	Sends any command as a string. * Refer to "LaserMarker Communication Manual" for supported models.	*	*	*	38

2.2.6.1. CaoController::Execute("GetReady ")

Returns whether the laser marker is ready for printing.

Argument type	Description
Without	-

Return Values Type	Description
VT_UI1	0: Print-ready status 1: Error occurring 2: Printing or sending program

Usage examples (CaoScript)

```
Dim result
Result = ctrl.Execute("GetReady")
Dbg.output result          'Display example: 0
```

2.2.6.2. CaoController::Execute("StartMarking ")

Printing starts. The argument type parameter specifies synchronous/asynchronous processing.

Argument type		Description
VT_ARRAY | VT_VARIANT		-
0	VT_BOOL	False: Synchronization. The next process does not start until printing is finished. True: Asynchronous process. Move on to the next process before printing is finished.
1	VT_UI4	Set the timeout [ms] from the start to the end of printing. If the print time is longer than the set value, a timeout error occurs. * Unlike other commands, this command does not use the set values of the Timeout options set during AddController. Value range: 0 to 4294967295

Return Values Type	Description
Without	-

Usage examples (CaoScript)

```
Result = ctrl.Execute("StartMarking",Array(true,3000))          Asynchronous processing
```

2.2.6.3. CaoController::Execute("GetStartMarking ")

Acquires the result of asynchronous printing. Use asynchronously with StartMarking commands. If the printing is not finished, the printer waits until the printing is finished. If you have not previously used StartMarking commands for asynchronous operations, an error is returned.

Argument type	Description
Without	-

Return Values Type	Description
VT_I4	0: Success Non-zero: Error code

Usage examples (CaoScript)

```
Result = ctrl.Execute("StartMarking",Array(true,3000))
Result = ctrl.Execute("GetStartMarking")
Dbg.output Hex(result)                    ' Display example: 80000900 (timeout error)....
```

2.2.6.4. CaoController::Execute("Check2DCode2 ")

Scan the two-dimensional code with the built-in camera of the laser marker. The X, Y, and Z coordinates vary depending on the model. For details, see Appendix-2, "List of Entries by Model" on page 49 of the LaserMarker Communication Guide. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type		Description
VT_ARRAY | VT_VARIANT		When the image capturing position is 0, specify the image capturing position and scanning area. When the image capturing position is 2, specify the image capturing position, scanning area, and block number.
0	VT_UI1	Imaging position 0: Current scanner position 1: Arbitrary position 2: Block coordinate
1	VT_UI1	Read area [%] Value range: 0 to 100 When the reading area is set to 0[%], 100% is set when the image capturing position is 0/1, and automatic area is set when the image capturing position is 2.
2	VT_UI1	Block number Value range: 0 to 255 * Specify 0 when the image capturing position is 1, and omit it when the image capturing position is 0.
3	VT_R8	X coordinate [mm] of the imaging position. * If the imaging position is 0/2, omit this. * The value range varies depending on the model.
4	VT_R8	Y coordinate [mm] of the shooting position. * If the imaging position is 0/2, omit this. * The value range varies depending on the model.
5	VT_R8	Z coordinate [mm] of the shooting position. * If the imaging position is 0/2, omit this.

		* The value range varies depending on the model.

Return Values Type		Description
VT_ARRAY | VT_BSTR		Returns the print quality and character string of the read two-dimensional code.
0	VT_BSTR	Overall grades of AIM-DPM Value range: A to D/F
1	VT_BSTR	The read string.

Usage examples (CaoScript)

```
'Imaging position :2,Reading area :0, Block No:100
Result = ctrl.Execute("Check2DCode2", Array(2,0,100))
Dim varResult
VarResult = DAT.ToVar(result)
Dbg.output varResult(0)          ' Display example: A
Dbg.output varResult(1)          ' Display example:test
```

2.2.6.5. CaoController::Execute("GetProgramNo ")

Gets the running program number.

Argument type	Description
Without	-

Return Values Type	Description
VT_UI2	Program No. Value range: 0 to 1999

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetProgramNo")
Dbg.output result          ' Display example: 1999
```

2.2.6.6. CaoController::Execute("SetProgramNo ")

Changes the running program number.

Argument type	Description
VT_UI2	Program No. Value range: 0 to 1999

Return Values Type	Description
Without	-

Usage examples (CaoScript)

```
Result = ctrl.Execute("SetProgramNo",1999)          'Set program No:1999
```

2.2.6.7. CaoController::Execute("GetCounterNo ")

Gets the current value of the specified counter number and the number of repetitive prints.

Argument type		Description
VT_ARRAY| VT_VARIANT		-
0	VT_UI2	Program No. Value range: 0 to 1999
1	VT_BSTR	Specify the counter No. string type. Value range: 0 to 9/A to J

Return Values Type		Description
VT_ARRAY| VT_UI4		-
0	VT_UI4	Current value of the counter Value range: 0 to 4294967295
1	VT_UI4	Number of repetitive prints Value range: 0 to 4294967295

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetCounterNo",Array(0,"A"))
VarResult = DAT.ToVar(result)
Dbg.output varResult (0)          ' Display example: 1000...
```

```
Dbg.output varResult (1)          ' Display example: 1000...
```

2.2.6.8. CaoController::Execute("SetCounterNo ")

Changes the current value of the specified counter No. and the number of repetitive prints.

Argument type		Description
VT_ARRAY| VT_VARIANT		-
0	VT_UI2	Program No. Value range: 0 to 1999
1	VT_BSTR	Specify the counter No. string type. Value range: 0 to 9/A to J
2	VT_UI4	Current value of the counter after the change Value range: 0 to 4294967295
3	VT_UI4	Number of repetitive prints after change Value range: 0 to 4294967295

Return Values Type	Description
Without	-

Usage examples (CaoScript)

```
Result = ctrl.Execute("SetCounterNo",Array(0,"A",1000,1000))
```

2.2.6.9. CaoController::Execute("GetMarkedCharacter ")

Gets the last print string of the specified block number.

Argument type		Description
VT_ARRAY| VT_UI2		-
0	VT_UI2	Program No. Value range: 0 to 1999
1	VT_UI2	Block No. Value range: 0 to 255

Return Values Type	Description
VT_BSTR	Returns the last printed string.

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetMarkedCharacter",Array(1999,255))
Dbg.output result                    'Display example: "test",...
```

2.2.6.10. CaoController::Execute("GetError ")

Gets the last print string of the specified block number.

Argument type	Description
Without	-

Return Values Type		Description
VT_ARRAY| VT_VARIANT		-
0	VT_BOOL	Indicates the error status. False: No errors True: Errored
1	VT_BSTR	Comma-separated list of errors that are occurring. Error Number:T000~T100 / W100~W200 / E001~E400 / S000~S100 Examples) T000,T001,S000,S100 * Error status 0: If no error occurs, empty characters are stored.

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetError")
Dim varResult
VarResult = DAT.ToVar(result)
Dbg.output varResult(0)              'Display example: "True"
Dbg.output varResult(1)              'Display example: "T000,T001,S000,S100"
```

2.2.6.11. CaoController::Execute("ClearError ")

Clears the currently pending errors.

Argument type	Description
Without	-

.

Return Values Type	Description
Without	-

Usage examples (CaoScript)

```
Result = ctrl.Execute("ClearError")
```

2.2.6.12. CaoController::Execute("GetAllPosition ")

Acquires the correction of the set position of the equipment.

Argument type	Description
Without	-

Return Values Type		Description
VT_ARRAY| VT_R8		The value range of the X, Y, and Z coordinate correction amounts varies depending on the equipment. For more information, refer to the LaserMarker Communication Manual.
0	VT_R8	X Rotation angle [°] Value range:-90.000 to 90.000
1	VT_R8	Y Rotation angle [°] Value range:-90.000 to 90.000
2	VT_R8	*Angle* Value range:-180.000 to 180.000
3	VT_R8	X coordinate correction [mm] Value range: Model dependent
4	VT_R8	Y coordinate correction [mm] Value range: Model dependent
5	VT_R8	Z coordinate correction [mm] Value range: Model dependent

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetAllPosition")
Dim varResult
VarResult = DAT.ToVar(result)
Dbg.output varResult(0)                'Display example: "90"
Dbg.output varResult(1)                'Display example: "90"
Dbg.output varResult(2)                'Display example: "180"
Dbg.output varResult(3)                'Display example: "62.5"
Dbg.output varResult(4)                'Display example: "62.5"
Dbg.output varResult(5)                'Display example: "62.5"
```

2.2.6.13. CaoController::Execute("SetAllPosition ")

Changes the position correction of the device. This command applies to all programs.

Argument type		Description
VT_ARRAY| VT_R8		The range of the X, Y, and Z coordinate correction varies depending on the model. For more information, refer to the LaserMarker Communication Manual.
0	VT_R8	X Rotation angle [°] Value range:-90.000 to 90.000
1	VT_R8	Y Rotation angle [°] Value range:-90.000 to 90.000
2	VT_R8	*Angle* Value range:-180.000 to 180.000
3	VT_R8	X coordinate correction [mm] Value range: Model dependent
4	VT_R8	Y coordinate correction [mm] Value range: Model dependent
5	VT_R8	Z coordinate correction [mm] Value range: Model dependent

Return Values Type	Description
Without	-

Usage examples (CaoScript)

```
Result = ctrl.Execute("SetAllPosition",Array(-55.555,-44.44,-33.3,0,50.55,60))
```

2.2.6.14. CaoController::Execute("GetTime ")

Gets the time of the built-in clock of the equipment.

Argument type	Description
Without	-

Return Values Type	Description
VT_DATE	Gets the current time of the device. Value range: 2000/01/0100:00:00 to 2099/12/3123:59:59

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetTime")
Dbg.output result                    'Display example : "2019/01/0123:59:59"
```

2.2.6.15. CaoController::Execute("SetTime ")

Changes the time of the built-in clock of the equipment.

Argument type	Description
VT_DATE	Specify the time to set. Value range: 2000/01/0100:00:00 to 2099/12/3123:59:59

Return Values Type	Description
Without	-

Usage examples (CaoScript)

```
Result = ctrl.Execute("SetTime","2000/01/01 00:00:00")
```

2.2.6.16. CaoController::Execute("GetPowerOffset ")

Get laser power offset.

Argument type	Description
Without	-

Return Values Type	Description
VT_R4	Power offset [%] Value range:-100.0 to 100.0

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetPowerOffset")
Dbg.output result                    'Display example: "-99.9"
```

2.2.6.17. CaoController::Execute("SetPowerOffset ")

Change the laser power offset.

Argument type	Description
VT_R4	Power offset [%] Value range:-100.0 to 100.0

Return Values Type	Description
Without	-

Usage examples (CaoScript)

```
Result = ctrl.Execute("SetPowerOffset",-99.9)
```

2.2.6.18. CaoController::Execute("GetCharacterString ")

Gets the specified program number, block number character string, two-dimensional code, logo file, etc.

Argument type		Description
VT_ARRAY | VT_UI2		-
0	VT_UI2	Program No. Value range: 0 to 1999
1	VT_UI2	Block No. Value range: 0 to 255

Return Values Type	Description
VT_BSTR	Read character string/logo file name

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetCharacterString",Array(100,100))
Dbg.output result                    'Display example: "test"
```

2.2.6.19. CaoController::Execute("SetCharacterString ")

Changes the specified program number, block number character string, two-dimensional code, logo file, etc.

Argument type		Description
VT_ARRAY | VT_VARIANT		-
0	VT_UI2	Program No. Value range: 0 to 1999
1	VT_UI2	Block No. Value range: 0 to 255
2	VT_BSTR	Character string/logo file name to be set

Return Values Type	Description
Without	-

Usage examples (CaoScript)

```
Result = ctrl.Execute("SetCharacterString",Array(100,100,"test2"))
```

2.2.6.20. CaoController::Execute("GetOperatingTime ")

Get the cumulative operating time [h] for the laser marker.

Argument type	Description
Without	-

Return Values Type	Description
VT_UI4	Cumulative operating time [h] Value range: 0 to 99999

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetOperatingTime")
Dbg.output result                    ' Display example: 1000 [h]
```

2.2.6.21. CaoController::Execute("GetLaserOperatingTime ")

Acquires the excitation time [h] of the laser marker. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type	Description
Without	-

Return Values Type	Description
VT_UI4	Laser Exit Time [h] Value range: 0 to 99999

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetLaserOperatingTime")
Dbg.output result                    ' Display example: 1000 [h]
```

2.2.6.22. CaoController::Execute("GetScannerOperatingTime ")

Acquires the scanner operating time [h]. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type	Description
Without	-

Return Values Type	Description
VT_UI4	Scanner operating time [h] Value range: 0 to 99999

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetScannerOperatingTime")
Dbg.output result                    ' Display example: 1000 [h]
```

2.2.6.23. CaoController::Execute("GetShutterOperatingCount ")

Gets the number of shutter operations. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type	Description
Without	-

Return Values Type	Description
VT_UI4	Number of shutter operations Value range: 0 to 4294967295

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetShutterOperatingCount")
Dbg.output result                    ' Display example: 1000 [h]
```

2.2.6.24. CaoController::Execute("GetContactorOperatingCount ")

Gets the number of contactor operations. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type	Description
Without	-

Return Values Type	Description
VT_UI4	Number of contactor operations Value range: 0 to 4294967295

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetContactorOperatingCount")
Dbg.output result                    ' Display example: 1000 [times]
```

2.2.6.25. CaoController::Execute("GetMarkingUnitTemperature ")

Get the head temperature [°C]. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type	Description
Without	-

Return Values Type	Description
VT_R4	Head temperature Value range:-999.9 to 999.9

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetMarkingUnitTemperature")
Dbg.output result                    ' Display example: 100.0 [℃]
```

2.2.6.26. CaoController::Execute("GetControllerTemperature ")

Get the controller temperature [°C]. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type	Description
Without	-

Return Values Type	Description
VT_R4	Head temperature Value range:-999.9 to 999.9

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetControllerTemperature")
Dbg.output result                    ' Display example: 100.0 [℃]
```

2.2.6.27. CaoController::Execute("GetLaserPowerCalibrationResult ")

Acquire whether laser power calibration is performed and the result [W]. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type	Description
Without	-

Return Values Type		Description
VT_ARRAY | VT_VARIANT		-
0	VT_BOOL	Whether to perform calibration.

		False: None True: Yes
1	VT_R4	Calibration result [W] (0 is stored if the calibration is false).) Value range: 0 to 999.99

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetLaserPowerCalibrationResult")
Dim varResult
VarResult = DAT.ToVar(result)
Dbg.output result(0)                    ' Display example: True
Dbg.output result(1)                    ' Display example: 100.0 [W]
```

2.2.6.28. CaoController::Execute("GetCumulativeMarkingCount ")

Acquires the cumulated number of prints, 1 and 2.

Argument type	Description
Without	-

Return Values Type		Description
VT_ARRAY | VT_UI4		-
0	VT_UI4	Cumulative print count 1 [times] Value range: 0 to 4294967295
1	VT_UI4	Cumulative print count 2 [times] Value range: 0 to 4294967295

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetCumulativeMarkingCount")
Dim varResult
VarResult = DAT.ToVar(result)
Dbg.output varResult (0)                ' Display example: 100 [times]
Dbg.output varResult (1)                ' Display example: 200 [times]
```

2.2.6.29. CaoController::Execute("SetCumulativeMarkingCount ")

Changes the cumulative number of prints 1 and 2.

Argument type		Description
VT_ARRAY | VT_UI4		-
0	VT_UI4	Cumulative print count 1 [times] Value range: 0 to 4294967295
1	VT_UI4	Cumulative print count 2 [times] Value range: 0 to 4294967295

Return Values Type	Description
Without	-

Usage examples (CaoScript)

```
Result = ctrl.Execute("SetCumulativeMarkingCount",Array(500,400))
```

2.2.6.30. CaoController::Execute("GetDesiccantExpirationDate ")

Get the Head Drying Agent Replacement Deadline. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type	Description
Without	-

Return Values Type	Description
VT_DATE	Head Drying Agent Exchange Deadline Value range: 2000/01/01 to 2099/12/31

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetDesiccantExpirationDate")
Dbg.output result                    'Display example :"2022/05/0100:00:00"
```

2.2.6.31. CaoController::Execute("GetSafetyShutterAOperatingCount ")

Acquires the number of opening and closing times of Safety Shutter A. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type	Description

Without	-
Return Values Type	Description
VT_UI4	Number of opening and closing of safety shutter A Value range: 0 to 4294967295

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetSafetyShutterAOperatingCount")
Dbg.output result                    ' Display example: 1000 [times]
```

2.2.6.32. CaoController::Execute("GetSafetyShutterBOperatingCount ")

Acquires the number of opening and closing times of Safety Shutter B. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type	Description
Without	-

Return Values Type	Description
VT_UI4	Number of opening and closing of safety shutter B Value range: 0 to 4294967295

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetSafetyShutterBOperatingCount")
Dbg.output result                    ' Display example: 1000 [times]
```

2.2.6.33. CaoController::Execute("GetMarkingLaserOscillatorTemperature ")

Get the laser oscillator tube temperature [℃]. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

Argument type	Description
Without	-

Return Values Type	Description
VT_R4	Head temperature Value range:-999.9 to 999.9

Usage examples (CaoScript)

```
Result = ctrl.Execute("GetMarkingUnitTemperature")
Dbg.output result                    ' Display example: 100.0 [℃]
```

2.2.6.34. CaoController::Execute("StartRetryMarking ")

This command is a wrapped version of several commands for ease of use in actual production facilities. Fig. 2-3 shows the Operation of this command.In actual production sites, printing may not be successful at one time. This command can be printed repeatedly to meet the print quality standard(General grades of AIM-DPM) set in this case. Fig. 2-4 shows the flow chart. This command is executed asynchronously, so you can proceed to the next process during laser printing. Some models may not be able to use this command. Refer to Table 2-5 for supported models.

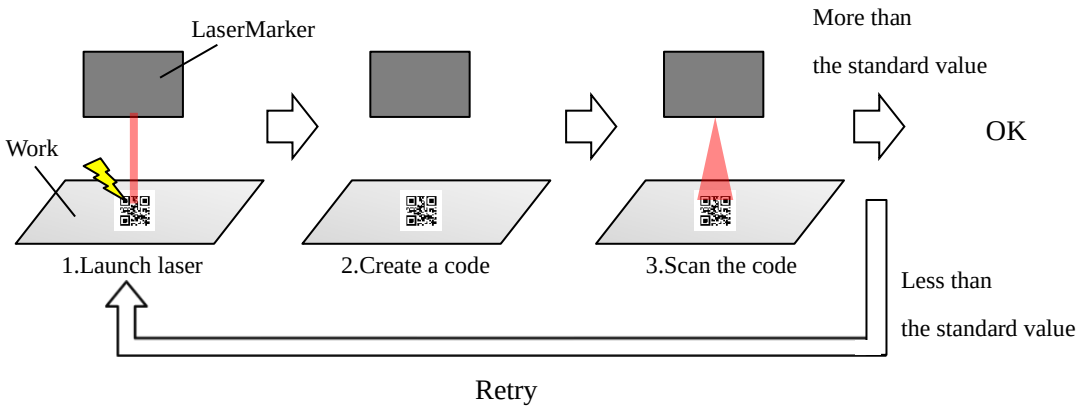

Fig. 2-3 Operation of the StartRetryMarking command

Argument type		Description
VT_ARRAY | VT_VARIANT		-
0	VT_UI2	Program No. used for the first printing Value range: 0 to 1999
1	VT_UI4	Timeout of the first print StartMarking commands[ms]. (Except for StartMarking commands, the time-out set by Option strings during AddController is used.) Value range: 0 to 4294967295
2	VT_UI2	Program No. to be used for the second and subsequent printing Value range: 0 to 1999
3	VT_UI4	Time-out of the second and subsequent StartMarking commands[ms]. (Except for

		StartMarking commands, the time-out set by Option strings during AddController is used.) Value range: 0 to 4294967295
4	VT_UI1	Block No. for checking print quality Value range: 0 to 255
5	VT_BSTR	Specify the print quality in the range from A to D based on the standard of print quality. General grades of the AIM-DPM. When the printed character string does not satisfy the set print quality standard, printing is executed again. Value range: A to D
6	VT_UI1	When the printed character string does not meet the print quality criteria set in 4, set the maximum number of times to repeat. Value range: 1 to 255

Return Values Type	Description
Without	-

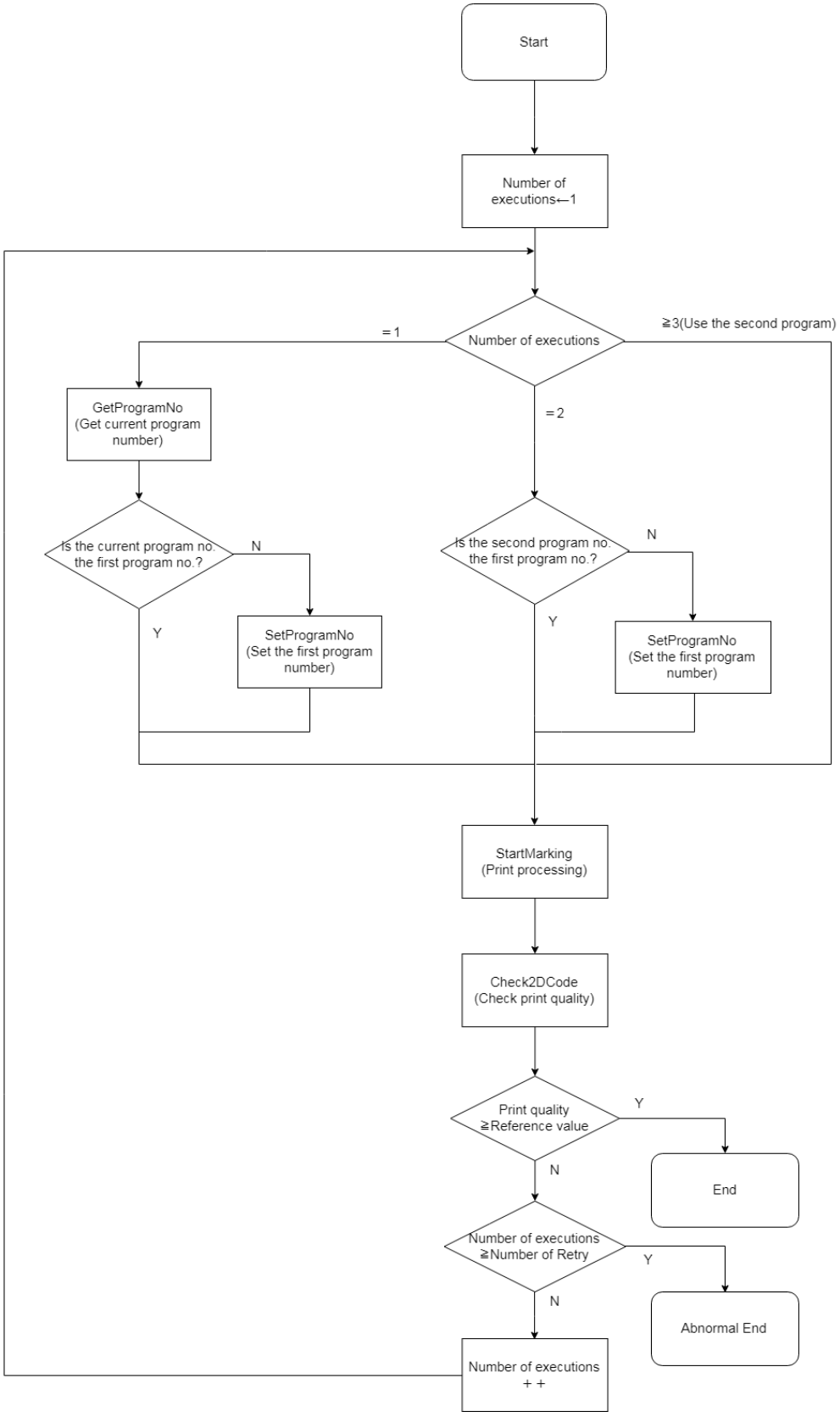

Fig. 2-4 Flow chart of StartRetryMarking

2.2.6.35. CaoController::Execute("GetRetryMarking ")

Used in conjunction with the StartRetryMarking command to get the results of the StartRetryMarking command. If the printing is not finished, the printer waits until the printing is finished. If you have not previously used StartRetryMarking commands for asynchronous operations, an error is returned.

Argument type	Description
Without	-

Return Values Type		Description
VT_ARRAY | VT_VARIANT		-
0	VT_BSTR	AIM-DPM overall grades read last (A~D/F) Values: A to D /F/(blank) * If the printing quality has not been determined once, a blank is stored.
1	VT_BSTR	Result of reading the printed character string Example) If you print a QR code with the character string "test", "test" is stored in the QR code.
2	VT_UI1	The number of times printing was performed. Value range: 1 to 255
3	VT_BSTR	Contains the name of the function in which the error occurred. If an error occurs halfway, you know which command failed. However, if no error occurs, a (blank) is stored. Example) If the GetProgramNo command fails, "GetProgramNo" is stored.
4	VT_I4	Indicates the error code. 0: Success Non-0: Error (For error details, please see Table 2-6 Unique Error Codes)

Usage examples (CaoScript)

```
'Program No. used for the first time: 100
 'First print timeout: 2000 ms
'Program No. used for the second time or later: 101
 'Second or later print timeout: 5000 ms
 'Specified block No:255
'Standard of print quality: A or more
'Maximum number of repetitions: 10 times
```

```
Result = ctrl.Execute("StartRetryMarking",Array(100,2000,101,5000,255,"A",10))
'Describes the operation to be performed in parallel with the printing here....
Result = ctrl.Execute("GetRetryMarking","")
Dim vntResult
VntResult = DAT.ToVar(result)
Dbg.output vntResult(0)          'Display example: "B"
Dbg.output vntResult(1)          'Display example: "test"
Dbg.output vntResult(2)          'Display example: "5"
Dbg.output vntResult(3)          'Display example: "Check2DCode2"
Dbg.output Hex(vntResult(4))     'Display example: "80110001" (Error code: S001)
```

2.2.6.36. CaoController::Execute("SendCommand ")

This is used when you want to use commands that are described in the "LaserMarker Communication Manual" and are not implemented by this provider.

Argument type	Description
VT_BSTR	Refer to the communication command in the LaserMarker Communication Manual and enter the data command you want to send. Example) To acquire print start acceptance. " RX,TriggerLock "

Return Values Type	Description
VT_BSTR	Received data command Example) If the sent "RX,TriggerLock" command was successful "RX,OK,0" (0: ready to start printing, 1: disabled to start printing)

Usage examples (CaoScript)

```
Result = ctrl.Execute("SendCommand","RX,TriggerLock")
Dbg.output result          ' Display example: "RX,OK,0"....
```

2.3. Error-code

This provider defines a unique error code. The unique error codes are shown in Table 2-6. For ORiN2 common errors, refer to the Error Codes section of the ORiN2 Programming Guide.

Table 2-6 Unique Error Codes

Error Number	Description
0x8011xxxx	The error code of the laser marker starts with "S___". The last four digits of "xxxx" converted to decimal numbers correspond to "___". Check the user's manual for the cause of the error.[7]
0x8012 xxxx	The error code for the laser marker starts with "E___". The last four digits of "xxxx" converted to decimal numbers correspond to "___". Check the user's manual for the cause of the error.[7]
0x8013 xxxx	Error code of laser marker starting from "T___". The last four digits of "xxxx" converted to decimal numbers correspond to "___". Check the user's manual for the cause of the error. [7]
0x8014 xxxx	The error code of the laser marker starts with "W___". The last four digits of "xxxx" converted to decimal numbers correspond to "___". Check the user's manual for the cause of the error. [7]
0x80150000	Error code of laser marker starting from other than S, E, T, or W. Capture the response packet with WireShark etc. and investigate the error cause from the user's manual to be used.
0x80160000	This error code is printed when you use the GetStartMarking command before you use the StartMarking command, or when you use the GetStartMarking command before you use the StartRetryMarking command.
0x80170000	This error code is printed when the number of times of printing exceeds the set maximum number of times of repetition of StartRetryMarking commands.
0x80180000	StartMarking (asynchronous) and errors generated when using other commands while the StartRetryMarking is running. You can wait for the StartMarking (asynchronous) and StartRetryMarking to finish by setting the Timeout for AddController longer.
0x80190000	This error code is output when the ProviderCancel command is executed while the StartRetryMarking command is executed. To execute the StartRetryMarking command again, execute the ProviderClear command.
0x801A0000	This error code is output when an invalid packet is received. Capture the response

[7] If the lower 4 digits are "FFFF", an incorrect response packet may have been received. Capture the response packet with WireShark etc. and investigate the error cause from the user's manual to be used.

	packet with WireShark etc. and investigate the error cause from the user's manual to be used.

3. Communication protocol command correspondence table

Table 3-1 shows the correspondences between Execute methods, Variable variables, and communication commands in the LaserMarker communication manuals.

Table 3-1 Communication Command Correspondence Table

Execute method	Variable name	Get/Put	Communications command
GetReady	@READY	Get	[RX]Ready
StartMarking	-	-	[WX]StartMarking
Check2DCode2	-	-	[WX]Check2DCode2
GetProgramNo	@PROGRAM_NO	Get	[RX]ProgramNo
SetProgramNo		Put	[WX]ProgramNo
GetCounterNo	Counter_*	Get	[RX]Counter
SetCounterNo		Put	[WX] Counter
GetMarkedCharacter	Marked_*	Get	[RX]MarkedCharacter
GetError	@ERROR	Get	[RX]Error
ClearError			[WX]ErrorClear
GetAllPosition	@ALL_POSITION	Get	[RX]AllPosition
SetAllPosition		Put	[WX]AllPosition
GetTime	@TIME	Get	[RX]TimeSetting
SetTime		Put	[WX]TimeSetting
GetPowerOffset	@POWER_OFFSET	Get	[RX]PowerOffset
SetPowerOffset		Put	[WX]PowerOffset
GetCharacterString	String_*	Get	[RX]CharacterString
SetCharacterString		Put	[WX]CharacterString
GetOperatingTime	@OPERATING_TIME	Get	[RX]OperatingTime
GetLaserOperatingTime	@LASER_OPERATING_TIME	Get	[RX]LaserOperatingTime
GetScannerOperatingTime	@SCANNER_OPERATING_TIME	Get	[RX]ScannerOperatingTime
GetShutterOperatingCount	@SHUTTER_OPERATING_TIME	Get	[RX]ShutterOperatingCount

GetContactorOperatingCount	@CONTACTOR_OPERATING_TIME	Get	[RX]ContactorOperatingCount
GetMarkingUnitTemperature	@MARKING_UNIT_TEMPERATURE	Get	[RX]MarkingUnitTemperature
GetControllerTemperature	@CONTROLLER_TEMPERATURE	Get	[RX]ControllerTemperature
GetLaserPowerCalibrationResult	@LASER_POWER_CALIBRATION_RESULT	Get	[RX]LaserPowerCalibratingResult
GetCumulativeMarkingCount	@CUMULATIVE_MARKING_COUNT	Get	[RX]CumulativeMarkingCount
SetCumulativeMarkingCount		Put	[WX]CumulativeMarkingCount
GetDesiccantExpirationDate	@DESICCANT_EXPIRATION_DATE	Get	[RX]DesiccantExpirationDate
GetSafetyShutterAOperatingCount	@SAFETY_SHUTTER_A_OPERATION_COUNT	Get	[RX]SafetyShutterAOperatingCount
GetSafetyShutterBOperatingCount	@SAFETY_SHUTTER_B_OPERATION_COUNT	Get	[RX]SafetyShutterBOperatingCount
GetMarkingLaserOscillatorTemperature	@MARKING_LASER_OSCILLATOR_TEMPERATURE	Get	[RX]MarkingLaserOscillatorTemperature