

KEYENCE

LaserMarker プロバイダ

Version 1.0.0

ユーザーズ ガイド

May 13, 2019

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2019-05-13	初版.

【対応機器】

機種	バージョン	注意事項
MD-X1000/1500 シリーズ		
MD-F3200/5200 シリーズ		
MD-U1000 シリーズ		
ML-Z9600 シリーズ		

目次

1. はじめに	5
2. プロバイダの概要	5
2.1. 概要	5
2.2. メソッド・プロパティ	7
2.2.1. CaoWorkspace::AddController メソッド	7
2.2.1.1. Conn オプション	7
2.2.2. CaoController::get_VariableNames プロパティ	8
2.2.3. CaoController::AddVariable メソッド	12
2.2.4. CaoVariable::get_Value プロパティ	14
2.2.5. CaoVariable::put_Value プロパティ	14
2.2.6. CaoVariable::Execute メソッド	14
2.2.6.1. CaoController : Execute("GetReady")コマンド	17
2.2.6.2. CaoController : Execute("StartMarking")コマンド	17
2.2.6.3. CaoController : Execute("GetStartMarking")コマンド	18
2.2.6.4. CaoController : Execute("Check2DCode2")コマンド	18
2.2.6.5. CaoController : Execute("GetProgramNo")コマンド	19
2.2.6.6. CaoController : Execute("SetProgramNo")コマンド	20
2.2.6.7. CaoController : Execute("GetCounterNo")コマンド	20
2.2.6.8. CaoController : Execute("SetCounterNo")コマンド	21
2.2.6.9. CaoController : Execute("GetMarkedCharacter")コマンド	21
2.2.6.10. CaoController : Execute("GetError")コマンド	22
2.2.6.11. CaoController : Execute("ClearError")コマンド	22
2.2.6.12. CaoController : Execute("GetAllPosition")コマンド	23
2.2.6.13. CaoController : Execute("SetAllPosition")コマンド	24
2.2.6.14. CaoController : Execute("GetTime")コマンド	24
2.2.6.15. CaoController : Execute("SetTime")コマンド	25
2.2.6.16. CaoController : Execute("GetPowerOffset")コマンド	25
2.2.6.17. CaoController : Execute("SetPowerOffset")コマンド	25
2.2.6.18. CaoController : Execute("GetCharacterString")コマンド	26
2.2.6.19. CaoController : Execute("SetCharacterString")コマンド	26
2.2.6.20. CaoController : Execute("GetOperatingTime")コマンド	27
2.2.6.21. CaoController : Execute("GetLaserOperatingTime")コマンド	27
2.2.6.22. CaoController : Execute("GetScannerOperatingTime")コマンド	27

2.2.6.23. CaoController : Execute("GetShutterOperatingCount")コマンド	28
2.2.6.24. CaoController : Execute("GetContactorOperatingCount")コマンド	28
2.2.6.25. CaoController : Execute("GetMarkingUnitTemperature")コマンド	29
2.2.6.26. CaoController : Execute("GetControllerTemperature")コマンド	29
2.2.6.27. CaoController : Execute("GetLaserPowerCalibrationResult")コマンド	29
2.2.6.28. CaoController : Execute("GetCumulativeMarkingCount")コマンド	30
2.2.6.29. CaoController : Execute("SetCumulativeMarkingCount")コマンド	31
2.2.6.30. CaoController : Execute("GetDesiccantExpirationDate")コマンド	31
2.2.6.31. CaoController : Execute("GetSafetyShutterAOperatingCount")コマンド	31
2.2.6.32. CaoController : Execute("GetSafetyShutterBOperatingCount")コマンド	32
2.2.6.33. CaoController : Execute("GetMarkingLaserOscillatorTemperature")コマンド	32
2.2.6.34. CaoController : Execute("StartRetryMarking")コマンド	33
2.2.6.35. CaoController : Execute("GetRetryMarking")コマンド	36
2.2.6.36. CaoController : Execute("SendCommand")コマンド	37
2.3. エラーコード	38
3. 通信プロトコルコマンド対応表	39

1. はじめに

本書は株式会社 KEYENCE 社製のレーザマーカ(MD-X1000/1500 シリーズ, MD-F3200/5200 シリーズ, MD-U1000 シリーズ, ML-Z9600 シリーズ)に対してデータの書き込み/読み込みおよびレーザマーカの操作を行う CAO プロバイダのユーザーズガイドです。

本書で扱う CAO プロバイダ(CaoProvKEYENCELaserMarker.dll)を LaserMarker プロバイダと呼びます。LaserMarker プロバイダは株式会社 KEYENCE から提供された「LaserMarker 通信インターフェースユーザーズマニュアル for MD-X1000/1500 シリーズ MD-F3200/5200 シリーズ MD-U1000 シリーズ ML-Z9600 シリーズ」(以降, LaserMarker 通信マニュアルと呼称します。)に沿って開発しています。本書は LaserMarker プロバイダの機能と, 実装されているメソッドについて説明します。

2. プロバイダの概要

2.1. 概要

図 2-1 にプロバイダの各クラスとレーザマーカの対応を示します。CaoProvController はレーザマーカ本体に, CaoProvVariable は, 印字するプログラム No や稼働時間などのレーザマーカの設定値に対応しています。

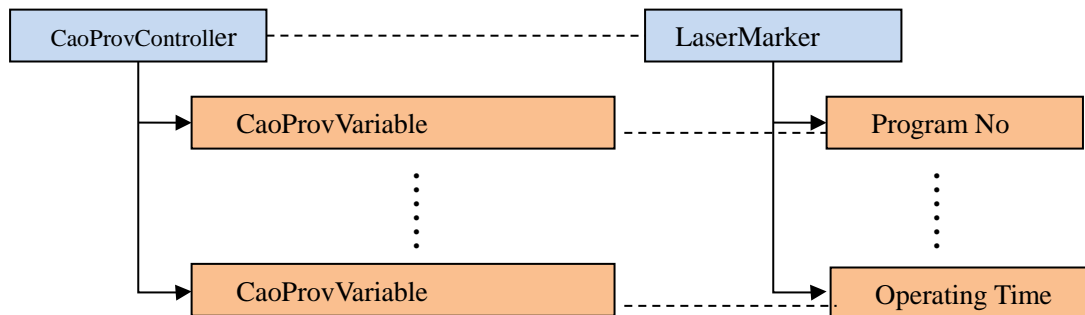


図 2-1 プロバイダの各クラスとレーザマーカの対応図

図 2-2 にプロバイダとレーザマーカとの接続を示します。接続対象とするレーザマーカには, Ethernet および COM の 2 種類の接続方法で接続してください。

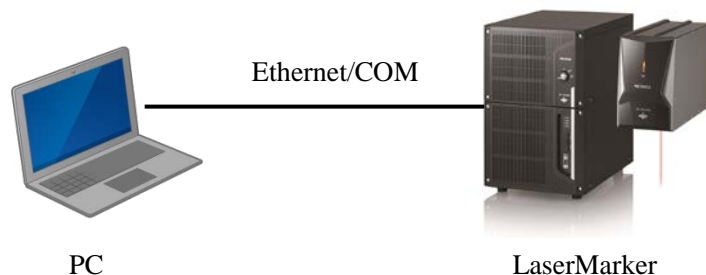


図 2-2 プロバイダとレーザマーカとの接続図

LaserMarker プロバイダのファイル形式は DLL(Dynamic Link Library)であり, CAO エンジンから使用時に動的にロードされます. LaserMarker プロバイダを使用するにあたっては ORiN2SDK をインストールするか, 表 2-1 を参照して手作業でレジストリ登録を行う必要があります.

表 2-1 LaserMarker プロバイダのファイル形式

ファイル名	CaoProvKEYENCELaserMarker.dll
ProgID	CaoProv.KEYENCE.LaserMarker
レジストリ登録 ¹	regsvr32 CaoProvKEYENCELaserMarker.dll
レジストリ登録の抹消	regsvr32 /u CaoProvKEYENCELaserMarker.dll

¹ ORiN SDK でインストールした場合は手動で登録/抹消する必要はありません.

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

本プロバイダでは、CaoWorkspace::AddController で接続パラメータを設定して通信の接続を行います。以下に AddController の仕様を示します。

書式

```
AddController(
    BSTR bstrCtrlName,          // [in] コントローラ名(任意のコントローラ名を入力して下さい)
    BSTR bstrProvName,          // [in] プロバイダ名. 固定値 = "CaoProv.KEYENCE.LaserMarker"
    BSTR bstrPcName,            // [in] プロバイダの実行マシン名
    BSTR bstrOption              // [in] オプション文字列
);
```

表 2-2 CaoWorkspace::AddController のオプション文字列

設定項目	オプション文字列	必須	説明
接続先	Conn	○	Ethernet 接続の場合と COM 接続の場合で記述方法が異なります。記述方法については、2.2.1.1Conn オプションを参照してください。
タイムアウト	Timeout	-	タイムアウトを 1～4294967295 ms で指定してください。 デフォルト値:3000 ms

2.2.1.1. Conn オプション

<Ethernet 接続の記述書式>

"eth:< A >[:< B >[:< C >[:< D >]]]"²

パラメータ	意味	必須	有効値
A	接続先 IP アドレス	○	192.168.0.20 など
B	接続先ポート番号	-	1～65535 デフォルト値:50002
C	自 IP アドレス	-	192.168.0.1 など デフォルト値:自動指定
D	自ポート番号	-	1～65535 デフォルト値:自動指定

例 1) IP アドレス 192.168.0.20, ポート番号 50002 へ接続する場合

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

²角括弧("[]")内は省略可能なパラメータです。

例 2) IP アドレス 192.168.0.20, ポート番号 50002, タイムアウト 1000 ms で接続する場合

"Conn = eth:192.168.0.20, Timeout = 1000"

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

<COM 接続の記述書式>

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

パラメータ	意味	必須	有効値 (デフォルト値:アンダーバー)
A	COM ポート番号	○	1～256
B	ボーレート[bps]	-	4800/9600/19200/ <u>38400</u> /57600/115200
C	パリティチェック	-	<u>N</u> … パリティなし E … 偶数(Even)パリティ O … 奇数(Odd)パリティ
D	データビット	-	7/ <u>8</u>
E	ストップビット	-	<u>1</u> /2
F	フロー制御	-	<u>0</u> … フロー制御なし 1 … Xon/Xoff フロー制御 2 … ハードウェア制御

例 1)デフォルト値で COM1 へ接続する場合

"Conn = com:1" または"Conn = com:1:38400:N:8:1:0"

例 2) デフォルト値で COM1 へタイムアウト 1000 ms で接続する場合

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

使用例(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 プロパティ

LaserMarker プロバイダで使用可能な変数リストを取得します。表 2-3 のリストが取得できます。

戻り値の型	説明
VT_ARRAY VT_BSTR	CaoController の AddVariable の変数名に使用できる変数一覧が取得できます。

使用例(CaoScript)

Dim vntNamesList

vntNamesList = dat.ToVar(ctrl.VariableNames)

dbg.output vntNamesList(0) ⇒表示結果例: @MAKER_NAME

dbg.output vntNamesList(1) ⇒表示結果例: @VERSION

表 2-3 CaoController クラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@MAKER_NAME	VT_BSTR	メーカー名「KEYENCE」を返します.	○	-
@VERSION	VT_BSTR	プロバイダのバージョンを返します.	○	-
@READY	VT_UI1	レーザマーカが印字可能状態かを返します. 0 … 印字可能状態 1 … エラー発生中 2 … 印字中またはプログラム送信中	○	-
@PROGRAM_NO	VT_UI2	運転中のプログラム No を取得/変更できます. 値範囲:0~1999	○	○
@ERROR	VT_ARRAY VT_VARIANT	エラー発生状態とエラー番号を返します.	○	-
	0 VT_BOOL	エラー発生状態 false…エラーなし true…エラーあり		
	1 VT_BSTR	発生しているエラー番号. エラーが複数発生している場合, カンマ区切りで文字列で表示します. 例)"E001, E002, W100, T000" *エラー内容についてはレーザマーカ本体のユーザーズマニュアルを参照してください. エラー番号: E001~E400/W100~W200/T000~T100		
@ALL_POSITION	VT_ARRAY VT_R8	機器設定の設置位置補正を変更/取得します. X~Z 座標補正量の値範囲はご使用される機種に依存します. 「LaserMarker 通信マニュアル」の「付録-2 機種別入力値一覧」(p.49)を参照してください.	○	○
	0 VT_R8	X 回転角[度] 値範囲: -90.000~90.000		

	1	VT_R8	Y 回転角[度] 値範囲: -90.000~90.000		
	2	VT_R8	Z 回転角[度] 値範囲: -180.000~180.000		
	3	VT_R8	X 座標補正量[mm] 値範囲: 機種依存		
	4	VT_R8	Y 座標補正量[mm] 値範囲: 機種依存		
	5	VT_R8	Z 座標補正量[mm] 値範囲: 機種依存		
@TIME	VT_DATE		内蔵時計の時刻を変更/取得します。 値範囲: 2000/1/1 00:00:00~2099/12/31/ 23:59:59 [書式] 年/月/日 時:分:秒 * 日と時の間に半角スペースを挿入してください。 例)"2000/1/1 00:00:00"	○	○
@POWER_OFFSET	VT_R4		機器設定のレーザパワーオフセット量[%]を変更/取得します。 値範囲: -100.0~100.0 * ブロックのレーザパワーにパワーオフセットを適用した値が 100%を超える場合, 100%で固定されます。 * 設定した値は全プログラムに対して適用します。 * レーザパワーが 0%のブロックには適用されません。	○	○
@OPERATING_TIME	VT_UI4		レーザマーカの累積稼働時間[h]を取得します。 値範囲: 0~99999	○	-
@LASER_OPERATING_TIME	VT_UI4		レーザ励起時間[h]を取得します。 値範囲: 0~99999 * MD-U1000 シリーズ, ML-Z9600 シリーズのみ使用可能です。	○	-
@SCANNER_OPERATING_TIME	VT_UI4		スキャナー稼働時間[h]を取得します。 値範囲: 0~99999 * MD-U1000 シリーズ, ML-Z9600 シリーズのみ使用可能です。	○	-
@SHUTTER_OPERATING_TIME	VT_UI4		シャッター動作回数[回]を取得します。 値範囲: 0~4294967295	○	-

		* MD-U1000 シリーズ, ML-Z9600 シリーズのみ使用可能です.		
@CONTACTOR_OPERATING_TIME	VT_UI4	コンタクター動作回数[回]を取得します. 値範囲:0~4294967295	○	-
@MARKING_UNIT_TEMPERATURE	VT_R4	ヘッド温度[°C]を取得します. 値範囲:-999.9~999.9 * MD-U1000 シリーズのみ使用可能です.	○	-
@CONTROLLER_TEMPERATURE	VT_R4	コントローラ温度[°C]を取得します. 値範囲:-999.9~999.9 * MD-U1000 シリーズのみ使用可能です.	○	-
@LASER_POWER_CALIBRATION_RESULT	VT_ARRAY VT_VARIANT	レーザパワーキャリブレーション実施の有無と結果[W]を取得します. * MD-U1000 シリーズのみ使用可能です.	○	-
	0 VT_BOOL	キャリブレーションの実施有無 false:なし true:あり		
	1 VT_R4	キャリブレーション結果[W] 値範囲:0~999.9		
@CUMULATIVE_MARKING_COUNT	VT_ARRAY VT_UI4	累積印字回数 1,2[回]を変更/取得します.	○	○
	0 VT_UI4	累積印字回数 1[回] 値範囲:0~4294967295		
	1 VT_UI4	累積印字回数 2[回] 値範囲:0~4294967295		
@DESICCANT_EXPIRATION_DATE	VT_DATE	ヘッド乾燥剤交換期限を取得します. 値範囲:2000/1/1 ~2099/12/31 * MD-U1000 シリーズでのみ使用可能です.	○	-
@SAFETY_SHUTTER_A_OPERATION_COUNT	VT_UI4	安全シャッターA の開閉回数[回]を取得します. 値範囲:0~4294967295 * ML-Z9600 シリーズでのみ使用可能です.	○	-
@SAFETY_SHUTTER_B_OPERATION_COUNT	VT_UI4	安全シャッターB の開閉回数[回]を取得します. 値範囲:0~4294967295 * ML-Z9600 シリーズでのみ使用可能です.	○	-
@MARKING_LASER_OSCILLATOR_TEMPERATURE	VT_R4	レーザ発振管温度[°C]を取得します. 値範囲:-999.9~999.9 * ML-Z9600 シリーズでのみ使用可能です.	○	-

2.2.3. CaoController::AddVariable メソッド

CaoController から CaoVariable オブジェクトを生成します。オプション文字列に関しては、システム変数を使用する場合は空白を、ユーザー変数を使用する場合は表 2-4 の各変数名に対応した文字列を入力してください。

書式

AddVariable (

BSTR bstrVarName, // [in] 変数名(システム変数名 or ユーザー変数名を指定)

BSTR bstrOption // [in] オプション文字列

);

使用例(CaoScript)

```
Dim varCumulativeMarkingCount
```

```
Set varCumulativeMarkingCount = ctrl.AddVariable("@CUMULATIVE_MARKING_COUNT", "")
```

```
Dim varString
```

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

表 2-4 CaoController クラス ユーザー変数一覧

変数名	データ型		説明	属性	
				get	put
Counter_*	VT_ARRAY VT_UI4		AddVariable 時の bstrOption にプログラム No とカウンターNo を指定することで、カウンターの現在値と繰り返し印字回数を変更/取得できます。 bstrOption は「PRG=(プログラム No), CTR=(カウンターNo)」の書式で指定してください。 プログラム No: 0～1999 カウンターNo: 0～9/A～J 例) "PRG=0,CTR=0", "PRG=1999,CTR=J" 変数名の「*」には他とかぶらないようにユニークな文字列を入れてください。 例) "Counter_001", "Counter_test"	○	○
	0	VT_UI4	カウンター現在値 値範囲: 0～4294967295		
	1	VT_UI4	繰り返し印字回数		

		値範囲:0~4294967295		
Marked_*	VT_BSTR	<p>AddVariable 時の bstrOption にプログラム No とブロック No を指定することで、レーザマーカが印字した最終印字文字列を取得できます。</p> <p>bstrOption は「PRG=(プログラム No), BLK=(ブロック No)」の書式で指定してください。 プログラム No 値範囲:0~1999 ブロック No 値範囲:0~255 例)"PRG=0,BLK=0", "PRG=1999,BLK=255"</p> <p>変数名の「*」には他とかぶらないようにユニークな文字列を入れてください。 例)"Marked_001", "Marked_test"</p>	○	-
String_*	VT_BSTR	<p>AddVariable 時の bstrOption にプログラム No とブロック No を指定することで、文字列またはロゴ・フォトファイルを変更/取得できます。</p> <p>bstrOption は「PRG=(プログラム No), BLK=(ブロック No)」の書式で指定してください。 プログラム No 値範囲:0~1999 ブロック No 値範囲:0~255 例)"PRG=0,BLK=0", "PRG=1999,BLK=255"</p> <p>変数名の「*」には他とかぶらないようにユニークな文字列を入れてください。 例)"String_001", "String_test"</p>	○	○

2.2.4. CaoVariable::get_Value プロパティ

AddVariable メソッドで作成した変数の状態を取得します。戻り値の型は AddVariable で指定した変数名によって異なります。各変数の戻り値は表 2-3 のシステム変数または表 2-4 のユーザー変数を参照ください。

使用例(CaoScript)

```
Dim vntCumulativeMarkingCount
vntCumulativeMarkingCount = DAT.ToVar (varCumulativeMarkingCount.Value)
dbg.output vntCumulativeMarkingCount(0)      '⇒表示結果例:100
dbg.output vntCumulativeMarkingCount(1)      '⇒表示結果例:100
dbg.output varString                          '⇒表示結果例:test
```

2.2.5. CaoVariable::put_Value プロパティ

AddVariable メソッドで作成した変数に値を設定します。表 2-3, 表 2-4 の変数で put に「○」がついている変数が put_Value プロパティを使用可能です。

使用例(CaoScript)

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

2.2.6. CaoVariable::Execute メソッド

CaoController クラスに属するプロバイダ固有のコマンドを実行します。Execute メソッドの引数は、コマンドを BSTR 型、パラメータを VARIANT 型で指定します。パラメータは省略可能です。

書式

```
Execute (
    BSTR bstrCmd          // [in] コマンド名
    [,VARIANT vntParam]   // [in] パラメータ
);
```

引数	説明
bstrCmd	表 2-5 の Execute メソッドのコマンド一覧より指定ください。
vntParam	コマンド名で使用するパラメータを指定します。「Set」から始まるコマンド名の場合、パラメータの設定が必須です。

表 2-5 CaoController::Execute メソッドのコマンド一覧

コマンド名	説明	対応機種			p.
		MD-X ³ MD-F ⁴	MD-U ⁵	ML-Z ⁶	
GetReady	レーザマーカが印字可能状態かを取得します。	○	○	○	17
StartMarking	印字を開始します。	○	○	○	17
GetStartMarking	非同期処理の印字結果を取得します。	○	○	○	18
Check2DCode2	レーザマーカの内蔵カメラで二次元コードを読み取ります。 (MD-X 1000L / 1500L シリーズには対応していません。)	△MD-X ×MD-F	○	×	18
GetProgramNo	運転中のプログラム No を取得します。	○	○	○	19
SetProgramNo	運転中のプログラム No を変更します。	○	○	○	20
GetCounterNo	指定するカウンターNo の現在値と繰り返し印字回数を取得します。	○	○	○	20
SetCounterNo	指定するカウンターNo の現在値と繰り返し印字回数を取得します。	○	○	○	21
GetMarkedCharacter	指定するブロック No の最終印字文字列を取得します。	○	○	○	21
GetError	レーザマーカで発生しているエラーを取得します。	○	○	○	22
ClearError	レーザマーカで発生しているエラーをクリアします。	○	○	○	22
GetAllPosition	機器の設置位置補正値を取得します。	○	○	○	23
SetAllPosition	機器の設置位置補正値を変更します。	○	○	○	24
GetTime	機器の時刻を取得します。	○	○	○	24
SetTime	機器の時刻を変更します。	○	○	○	25
GetPowerOffset	レーザパワーオフセットを取得します。	○	○	○	25
SetPowerOffset	レーザパワーオフセットを変更します。	○	○	○	25
GetCharacterString	指定したブロック No の文字列およびログファイルを取得します。	○	○	○	26

³ MD-X 1000/1500 シリーズ⁴ MD-F 3200/5200 シリーズ⁵ MD-U1000 シリーズ⁶ ML-Z9600 シリーズ

SetCharacterString	指定したブロック No の文字列およびロゴファイルを変更します。	○	○	○	26
GetOperatingTime	レーザマーカの累積稼働時間を取得します。	○	○	○	27
GetLaserOperatingTime	レーザ励起時間を取得します。	×	○	○	27
GetScannerOperatingTime	スキャナー稼働時間を取得します。	×	○	○	27
GetShutterOperatingCount	シャッター動作回数を取得します。	×	○	○	28
GetContactorOperatingCount	コンタクター動作回数を取得します。	×	○	×	28
GetMarkingUnitTemperature	ヘッド温度を取得します。	×	○	×	29
GetControllerTemperature	コントローラー温度を取得します。	×	○	×	29
GetLaserPowerCalibrationResult	レーザパワーキャリブレーション結果を取得します。	×	○	×	29
GetCumulativeMarkingCount	累積印字回数 1,2 を取得します。	○	○	○	30
SetCumulativeMarkingCount	累積印字回数 1,2 を変更します。	○	○	○	31
GetDesiccantExpirationDate	ヘッド乾燥剤交換期限を取得します。	×	○	×	31
GetSafetyShutterAOperatingCount	安全シャッターAの開閉回数を取得します。	×	×	○	31
GetSafetyShutterBOperatingCount	安全シャッターBの開閉回数を取得します。	×	×	○	32
GetMarkingLaserOscillatorTemperature	レーザ発振管温度を取得します。	×	×	○	32
StartRetryMarking	印字品質が設定値以上になるまで印字を繰り返します。(MD-X 1000L / 1500L シリーズには対応していません。)	△MD-X ×MD-F	○	×	33
GetRetryMarking	StartRetryMarking の結果を取得します。(MD-X 1000L / 1500L シリーズには対応していません。)	△MD-X ×MD-F	○	×	36
SendCommand	任意のコマンドを文字列で送信します。* 対	*	*	*	37

	応機種については、「LaserMarker 通信マニュアル」を参照ください。				
--	--	--	--	--	--

2.2.6.1. CaoController: :Execute("GetReady")コマンド

レーザマーカが印字可能状態かを返します。

引数の型	説明
なし	-

戻り値の型	説明
VT_UI1	0:印字可能状態 1:エラー発生中 2:印字中またはプログラム送信中

使用例(CaoScript)

```
Dim result
result = ctrl.Execute("GetReady")
dbg.output result      ⇒表示結果例:0
```

2.2.6.2. CaoController: :Execute("StartMarking")コマンド

印字を開始します。引数の型のパラメータで同期/非同期処理を指定します。

引数の型	説明
VT_ARRAY VT_VARIANT	-
0	VT_BOOL false:同期処理. 印字が終了するまで次の処理を開始しません. true:非同期処理. 印字が終了する前に次の処理に移ります.
1	VT_UI4 印字開始～終了までのタイムアウト[ms]を設定します. 設定値より印字時間が長い場合, タイムアウトのエラーが発生します. * 本コマンドは他のコマンドと異なり, AddController 時に設定した Timeout オプションの設定値を使用しません. 値範囲:0~4294967295

戻り値の型	説明
なし	-

使用例(CaoScript)

```
result = ctrl.Execute("StartMarking",Array(true,3000))
```

‘非同期処理

2.2.6.3. CaoController: :Execute(“GetStartMarking”)コマンド

非同期処理で印字した結果を取得します。非同期で StartMarking コマンドと組み合わせて使用します。印字が終了していない場合、印字が終了するまで待機します。事前に非同期処理で StartMarking コマンドを使用していない場合、エラーを返します。

引数の型	説明
なし	-

戻り値の型	説明
VT_I4	0: 成功 0 以外: エラーコード

使用例(CaoScript)

```
result = ctrl.Execute("StartMarking",Array(true,3000))
```

```
result = ctrl.Execute("GetStartMarking")
```

```
dbg.output Hex(result)
```

⇒表示結果例: 80000900(タイムアウトエラー)

2.2.6.4. CaoController: :Execute(“Check2DCode2”)コマンド

レーザマーカの内蔵カメラで二次元コードを読み取ります。X, Y, Z 座標は機種によって値の範囲が変わります。詳しくは、「LaserMarker 通信マニュアル」の「付録-2 機種別入力一覧」(p.49)を参照ください。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型		説明
VT_ARRAY VT_VARIANT		撮像位置が 0 の場合、撮像位置、読み取りエリアのみを指定してください。 撮像位置が 2 の場合、撮像位置、読み取りエリア、ブロック番号のみを指定してください。
0	VT_UI1	撮像位置 0: 現在のスキャナー位置 1: 任意位置 2: ブロック座標
1	VT_UI1	読み取りエリア[%] 値範囲: 0~100 読み取りエリア 0[%]を指定した場合、撮像位置が 0/1 のときは 100%, 撮像位置が 2 のときは自動エリア指定となります。

2	VT_UI1	ブロック番号 値範囲:0~255 * 撮像位置が 1 のときは 0 を指定, 撮像位置が 0 のときは省略してください.
3	VT_R8	撮影位置の X 座標[mm]. * 撮像位置が 0/2 のときは省略してください. * 値範囲は機種によって異なります.
4	VT_R8	撮影位置の Y 座標[mm]. * 撮像位置が 0/2 のときは省略してください. * 値範囲は機種によって異なります.
5	VT_R8	撮影位置の Z 座標[mm]. * 撮像位置が 0/2 のときは省略してください. * 値範囲は機種によって異なります.

戻り値の型		説明
VT_ARRAY VT_BSTR		読みとった二次元コードの印字品質と文字列を返します.
0	VT_BSTR	AIM-DPM の総合グレード 値範囲:A~D/F
1	VT_BSTR	読みとった文字列.

使用例(CaoScript)

‘撮像位置:2, 読み取りエリア:0, ブロック番号:100 の場合

```
result = ctrl.Execute("Check2DCode2", Array(2,0,100))
```

```
Dim varResult
```

```
varResult = DAT.ToVar(result)
```

```
dbg.output varResult(0)          ‘表示例:A
```

```
dbg.output varResult(1)          ‘表示例:test
```

2.2.6.5. CaoController: : Execute(“GetProgramNo”)コマンド

運転中のプログラム No を取得します.

引数の型	説明
なし	-

戻り値の型	説明
VT_UI2	プログラム No 値範囲:0~1999

使用例(CaoScript)

```
result = ctrl.Execute("GetProgramNo")
```

```
dbg.output result                   ⇒表示結果例:1999
```

2.2.6.6. CaoController: :Execute("SetProgramNo")コマンド

運転中のプログラム No を変更します。

引数の型	説明
VT_UI2	プログラム No 値範囲:0~1999

戻り値の型	説明
なし	-

使用例(CaoScript)

```
result = ctrl.Execute("SetProgramNo",1999)                   ⇒プログラム No:1999 を設定
```

2.2.6.7. CaoController: :Execute("GetCounterNo")コマンド

指定したカウンターNo の現在値および繰り返し印字回数を取得します。

引数の型	説明
VT_ARRAY VT_VARIANT	-
0 VT_UI2	プログラム No 値範囲:0~1999
1 VT_BSTR	カウンターNo. 文字列型で指定してください。 値範囲:0~9/A~J

戻り値の型	説明
VT_ARRAY VT_UI4	-
0 VT_UI4	カウンター現在値 値範囲:0~4294967295
1 VT_UI4	繰り返し印字回数 値範囲:0~4294967295

使用例(CaoScript)

```
result = ctrl.Execute("GetCounterNo",Array(0,"A"))
```

```
varResult = DAT.ToVar(result)
```

dbg.output varResult (0) ⇒表示結果例:1000

dbg.output varResult (1) ⇒表示結果例:1000

2.2.6.8. CaoController: :Execute("SetCounterNo")コマンド

指定したカウンターNo の現在値および繰り返し印字回数を変更します。

引数の型		説明
VT_ARRAY VT_VARIANT		-
0	VT_UI2	プログラム No 値範囲:0~1999
1	VT_BSTR	カウンターNo. 文字列型で指定してください. 値範囲:0~9/A~J
2	VT_UI4	変更後のカウンター現在値 値範囲:0~4294967295
3	VT_UI4	変更後の繰り返し印字回数 値範囲:0~4294967295

戻り値の型		説明
なし		-

使用例(CaoScript)

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

2.2.6.9. CaoController: :Execute("GetMarkedCharacter")コマンド

指定したブロック No の最終印字文字列を取得します。

引数の型		説明
VT_ARRAY VT_UI2		-
0	VT_UI2	プログラム No 値範囲:0~1999
1	VT_UI2	ブロック No. 値範囲:0~255

戻り値の型		説明
VT_BSTR		最終印字文字列を返します。

使用例(CaoScript)

```
result = ctrl.Execute("GetMarkedCharacter",Array(1999,255))
dbg.output result
```

⇒表示結果例:「test」

2.2.6.10. CaoController: :Execute("GetError")コマンド

指定したブロック No の最終印字文字列を取得します。

引数の型	説明
なし	-

戻り値の型	説明
VT_ARRAY VT_VARIANT	-
0 VT_BOOL	エラー発生状態を示します。 false: エラーなし true: エラーあり
1 VT_BSTR	発生しているエラーをカンマ区切りで示します。 エラー番号: T000～T100 / W100～W200 / E001～E400 / S000～S100 例)T000,T001,S000,S100 *エラー発生状態が 0:エラーなしの場合, 空文字が格納されます。

使用例(CaoScript)

```
result = ctrl.Execute("GetError")
Dim varResult
varResult = DAT.ToVar(result)
dbg.output varResult(0)
dbg.output varResult(1)
```

⇒表示結果例:「True」

⇒表示結果例:「T000,T001,S000,S100」

2.2.6.11. CaoController: :Execute("ClearError")コマンド

発生しているエラーをクリアします。

引数の型	説明
なし	-

戻り値の型	説明
なし	-

使用例(CaoScript)

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

2.2.6.12. CaoController: :Execute("GetAllPosition")コマンド

機器の設定位置補正を取得します。

引数の型	説明
なし	-

戻り値の型	説明
VT_ARRAY VT_R8	X,Y,Z 座標補正量の値範囲は機器によって異なります。詳細は「LaserMarker 通信マニュアル」の「付録-2 機種別入力値一覧」(p.49)を参照ください。
0 VT_R8	X 回転角[°] 値範囲:-90.000~90.000
1 VT_R8	Y 回転角[°] 値範囲:-90.000~90.000
2 VT_R8	θ 角度[°] 値範囲:-180.000~180.000
3 VT_R8	X 座標補正量[mm] 値範囲:機種依存
4 VT_R8	Y 座標補正量[mm] 値範囲:機種依存
5 VT_R8	Z 座標補正量[mm] 値範囲:機種依存

使用例(CaoScript)

```

result = ctrl.Execute("GetAllPosition")
Dim varResult
varResult = DAT.ToVar(result)
dbg.output varResult(0)           '⇒表示結果例:「90」
dbg.output varResult(1)           '⇒表示結果例:「90」
dbg.output varResult(2)           '⇒表示結果例:「180」
dbg.output varResult(3)           '⇒表示結果例:「62.5」
dbg.output varResult(4)           '⇒表示結果例:「62.5」
dbg.output varResult(5)           '⇒表示結果例:「62.5」

```

2.2.6.13. CaoController: :Execute("SetAllPosition")コマンド

機器の設定位置補正を変更します。このコマンドは全プログラムに適用されます。

引数の型		説明
VT_ARRAY VT_R8		X, Y, Z 座標補正量は機種によって値の範囲が変わります。詳細は「LaserMarker 通信マニュアル」の「付録-2 機種別入力一覧」(p.49)を参照ください。
0	VT_R8	X 回転角[°] 値範囲:-90.000~90.000
1	VT_R8	Y 回転角[°] 値範囲:-90.000~90.000
2	VT_R8	θ 角度[°] 値範囲:-180.000~180.000
3	VT_R8	X 座標補正量[mm] 値範囲:機種依存
4	VT_R8	Y 座標補正量[mm] 値範囲:機種依存
5	VT_R8	Z 座標補正量[mm] 値範囲:機種依存

戻り値の型	説明
なし	-

使用例(CaoScript)

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

2.2.6.14. CaoController: :Execute("GetTime")コマンド

機器の内蔵時計の時刻を取得します。

引数の型	説明
なし	-

戻り値の型	説明
VT_DATE	機器の現在時刻を取得します。 値範囲:2000/01/01 00:00:00~2099/12/31 23:59:59

使用例(CaoScript)

```
result = ctrl.Execute("GetTime")
```

```
dbg.output result
```

⇒表示結果例:「2019/01/01 23:59:59」

2.2.6.15. CaoController: :Execute("SetTime")コマンド

機器の内蔵時計の時刻を変更します。

引数の型	説明
VT_DATE	設定する時刻を指定してください。 値範囲: 2000/01/01 00:00:00～2099/12/31 23:59:59

戻り値の型	説明
なし	-

使用例(CaoScript)

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

2.2.6.16. CaoController: :Execute("GetPowerOffset")コマンド

レーザパワーオフセットを取得します。

引数の型	説明
なし	-

戻り値の型	説明
VT_R4	パワーオフセット量[%] 値範囲: -100.0～100.0

使用例(CaoScript)

```
result = ctrl.Execute("GetPowerOffset")
```

```
dbg.output result
```

⇒表示結果例: 「-99.9」

2.2.6.17. CaoController: :Execute("SetPowerOffset")コマンド

レーザパワーオフセットを変更します。

引数の型	説明
VT_R4	パワーオフセット量[%] 値範囲: -100.0～100.0

戻り値の型	説明
なし	-

使用例(CaoScript)

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

2.2.6.18. CaoController : Execute("GetCharacterString")コマンド

指定したプログラム No, ブロック No の文字列, 二次元コードおよびロゴファイルなどを取得します。

引数の型		説明
VT_ARRAY VT_UI2		-
0	VT_UI2	プログラム No 値範囲:0~1999
1	VT_UI2	ブロック No 値範囲:0~255

戻り値の型	説明
VT_BSTR	読み込んだ文字列/ロゴファイル名

使用例(CaoScript)

```
result = ctrl.Execute("GetCharacterString",Array(100,100))
```

```
dbg.output result
```

⇒表示結果例:「test」

2.2.6.19. CaoController : Execute("SetCharacterString")コマンド

指定したプログラム No, ブロック No の文字列, 二次元コードおよびロゴファイルなどを変更します。

引数の型		説明
VT_ARRAY VT_VARIANT		-
0	VT_UI2	プログラム No 値範囲:0~1999
1	VT_UI2	ブロック No 値範囲:0~255
2	VT_BSTR	設定する文字列/ロゴファイル名

戻り値の型	説明
なし	-

使用例(CaoScript)

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

2.2.6.20. CaoController: :Execute("GetOperatingTime")コマンド

レーザマーカの累積稼働時間[h]を取得します。

引数の型	説明
なし	-

戻り値の型	説明
VT_UI4	累積稼働時間[h] 値範囲:0～99999

使用例(CaoScript)

```
result = ctrl.Execute("GetOperatingTime")
dbg.output result          ⇨表示結果例:1000[h]
```

2.2.6.21. CaoController: :Execute("GetLaserOperatingTime")コマンド

レーザマーカの励起時間[h]を取得します。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型	説明
なし	-

戻り値の型	説明
VT_UI4	レーザ励起時間[h] 値範囲:0～99999

使用例(CaoScript)

```
result = ctrl.Execute("GetLaserOperatingTime")
dbg.output result          ⇨表示結果例:1000[h]
```

2.2.6.22. CaoController: :Execute("GetScannerOperatingTime")コマンド

スキャナー稼働時間 [h]を取得します。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型	説明
なし	-

戻り値の型	説明
VT_UI4	スキャナー稼働時間[h] 値範囲:0～99999

使用例(CaoScript)

```
result = ctrl.Execute("GetScannerOperatingTime")
dbg.output result
```

⇒表示結果例:1000[h]

2.2.6.23. CaoController: :Execute("GetShutterOperatingCount")コマンド

シャッター動作回数を取得します。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型	説明
なし	-

戻り値の型	説明
VT_UI4	シャッター動作回数 値範囲:0～4294967295

使用例(CaoScript)

```
result = ctrl.Execute("GetShutterOperatingCount")
dbg.output result
```

⇒表示結果例:1000[h]

2.2.6.24. CaoController: :Execute("GetContactorOperatingCount")コマンド

コンタクター動作回数を取得します。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型	説明
なし	-

戻り値の型	説明
VT_UI4	コンタクター動作回数 値範囲:0～4294967295

使用例(CaoScript)

```
result = ctrl.Execute("GetContactorOperatingCount")
dbg.output result
```

⇒表示結果例:1000[回]

2.2.6.25. CaoController: :Execute("GetMarkingUnitTemperature")コマンド

ヘッド温度[°C]を取得します。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型	説明
なし	-

戻り値の型	説明
VT_R4	ヘッド温度[°C] 値範囲: -999.9~999.9

使用例(CaoScript)

```
result = ctrl.Execute("GetMarkingUnitTemperature")
dbg.output result          ⇨表示結果例:100.0 [°C]
```

2.2.6.26. CaoController: :Execute("GetControllerTemperature")コマンド

コントローラ温度[°C]を取得します。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型	説明
なし	-

戻り値の型	説明
VT_R4	ヘッド温度[°C] 値範囲: -999.9~999.9

使用例(CaoScript)

```
result = ctrl.Execute("GetControllerTemperature")
dbg.output result          ⇨表示結果例:100.0 [°C]
```

2.2.6.27. CaoController: :Execute("GetLaserPowerCalibrationResult")コマンド

レーザパワーキャリブレーションの実施の有無と、その結果[W]を取得します。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型	説明
なし	-

戻り値の型		説明
VT_ARRAY VT_VARIANT		-
0	VT_BOOL	キャリブレーション実施の有無. false: なし true: あり
1	VT_R4	キャリブレーション結果[W](キャリブレーション実施が false の場合, 0 が格納されます.) 値範囲: 0~999.99

使用例(CaoScript)

```

result = ctrl.Execute("GetLaserPowerCalibrationResult")
Dim varResult
varResult = DAT.ToVar(result)
dbg.output result(0)           ⇒表示結果例: True
dbg.output result(1)           ⇒表示結果例: 100.0 [W]

```

2.2.6.28. CaoController::Execute("GetCumulativeMarkingCount")コマンド

累積印字回数 1, 2 を取得します.

引数の型	説明
なし	-

戻り値の型		説明
VT_ARRAY VT_UI4		-
0	VT_UI4	累積印字回数 1[回] 値範囲: 0~4294967295
1	VT_UI4	累積印字回数 2[回] 値範囲: 0~4294967295

使用例(CaoScript)

```

result = ctrl.Execute("GetCumulativeMarkingCount")
Dim varResult
varResult = DAT.ToVar(result)
dbg.output varResult (0)       ⇒表示結果例: 100[回]
dbg.output varResult (1)       ⇒表示結果例: 200[回]

```

2.2.6.29. CaoController: :Execute(“SetCumulativeMarkingCount”)コマンド

累積印字回数 1, 2 を変更します。

引数の型		説明
VT_ARRAY VT_UI4		-
0	VT_UI4	累積印字回数 1[回] 値範囲: 0～4294967295
1	VT_UI4	累積印字回数 2[回] 値範囲: 0～4294967295

戻り値の型	説明
なし	-

使用例(CaoScript)

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

2.2.6.30. CaoController: :Execute(“GetDesiccantExpirationDate”)コマンド

ヘッド乾燥剤交換期限を取得します。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型	説明
なし	-

戻り値の型	説明
VT_DATE	ヘッド乾燥剤交換期限 値範囲: 2000/01/01 ～2099/12/31

使用例(CaoScript)

```
result = ctrl.Execute("GetDesiccantExpirationDate")
```

```
dbg.output result
```

⇒表示結果例: 「2022/05/01 00:00:00」

2.2.6.31. CaoController: :Execute(“GetSafetyShutterAOperatingCount”)コマンド

安全シャッターA の開閉回数を取得します。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型	説明
なし	-

戻り値の型	説明
VT_UI4	安全シャッターAの開閉回数 値範囲:0~4294967295

使用例(CaoScript)

```
result = ctrl.Execute("GetSafetyShutterAOperatingCount")
dbg.output result          ⇨表示結果例:1000[回]
```

2.2.6.32. CaoController::Execute("GetSafetyShutterBOperatingCount")コマンド

安全シャッターBの開閉回数を取得します。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型	説明
なし	-

戻り値の型	説明
VT_UI4	安全シャッターBの開閉回数 値範囲:0~4294967295

使用例(CaoScript)

```
result = ctrl.Execute("GetSafetyShutterBOperatingCount")
dbg.output result          ⇨表示結果例:1000[回]
```

2.2.6.33. CaoController::Execute("GetMarkingLaserOscillatorTemperature")コマンド

レーザ発振管温度[℃]を取得します。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

引数の型	説明
なし	-

戻り値の型	説明
VT_R4	ヘッド温度[℃] 値範囲:-999.9~999.9

使用例(CaoScript)

```
result = ctrl.Execute("GetMarkingUnitTemperature")
dbg.output result          ⇨表示結果例:100.0 [℃]
```


2.2.6.34. CaoController : Execute(“StartRetryMarking”)コマンド

本コマンドは実際の生産設備で使いやすいように複数のコマンドをラップしたものです。図 2-3 は StartRetryMarking コマンドの動作を示しています。実際の生産現場では、1 回ではうまく印字できない場合があります。その場合に設定する印字品質基準 (AIM-DPM の総合グレード) を満たすように繰り返し印字できるのが本コマンドです。フローチャートを図 2-4 に示します。本コマンドは非同期処理で実行されますので、レーザ印字中も次の処理に移ることができます。また、本コマンドを使用できない機種があります。対応機種は表 2-5 を参照してください。

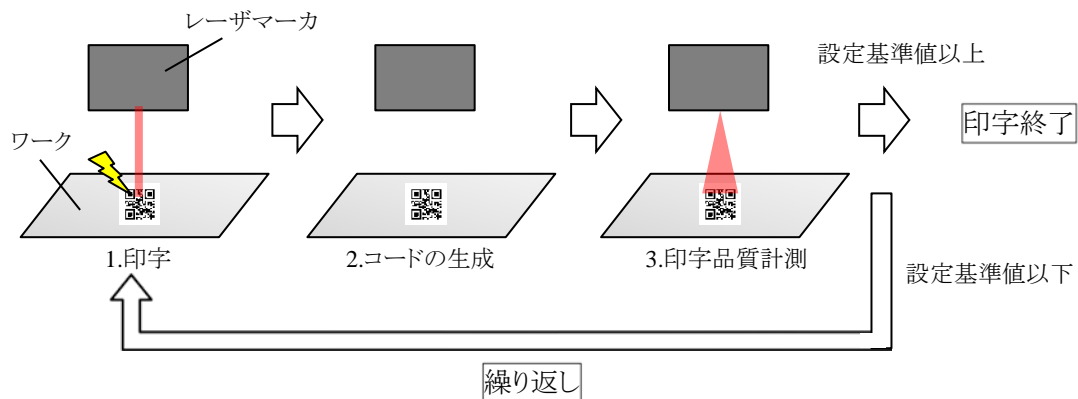


図 2-3 StartRetryMarking コマンドの動作

引数の型		説明
VT_ARRAY VT_VARIANT		-
0	VT_UI2	印字 1 回目で使用するプログラム No 値範囲: 0~1999
1	VT_UI4	印字 1 回目の StartMarking コマンドのタイムアウト[ms]. (StartMarking コマンド以外は AddController 時の Option 文字列で設定したタイムアウトを使用します。) 値範囲: 0~ 4294967295
2	VT_UI2	印字 2 回目以降で使用するプログラム No 値範囲: 0~1999
3	VT_UI4	印字 2 回目以降の StartMarking コマンドのタイムアウト[ms]. (StartMarking コマンド以外は AddController 時の Option 文字列で設定したタイムアウトを使用します。) 値範囲: 0~ 4294967295
4	VT_UI1	印字品質をチェックするブロック No

		値範囲:0~255
5	VT_BSTR	印字品質の基準. AIM-DPM の総合グレードに基づき, A~D の範囲で指定してください. 印字した文字列などが設定した印字品質基準に満たない場合, 再度印字が実行されます. 値範囲:A~D
6	VT_UI1	印字した文字列などが, 4 で設定した印字品質基準に満たない場合, 繰り返す最大実行回数を設定します. 値範囲:1~255

戻り値の型	説明
なし	-

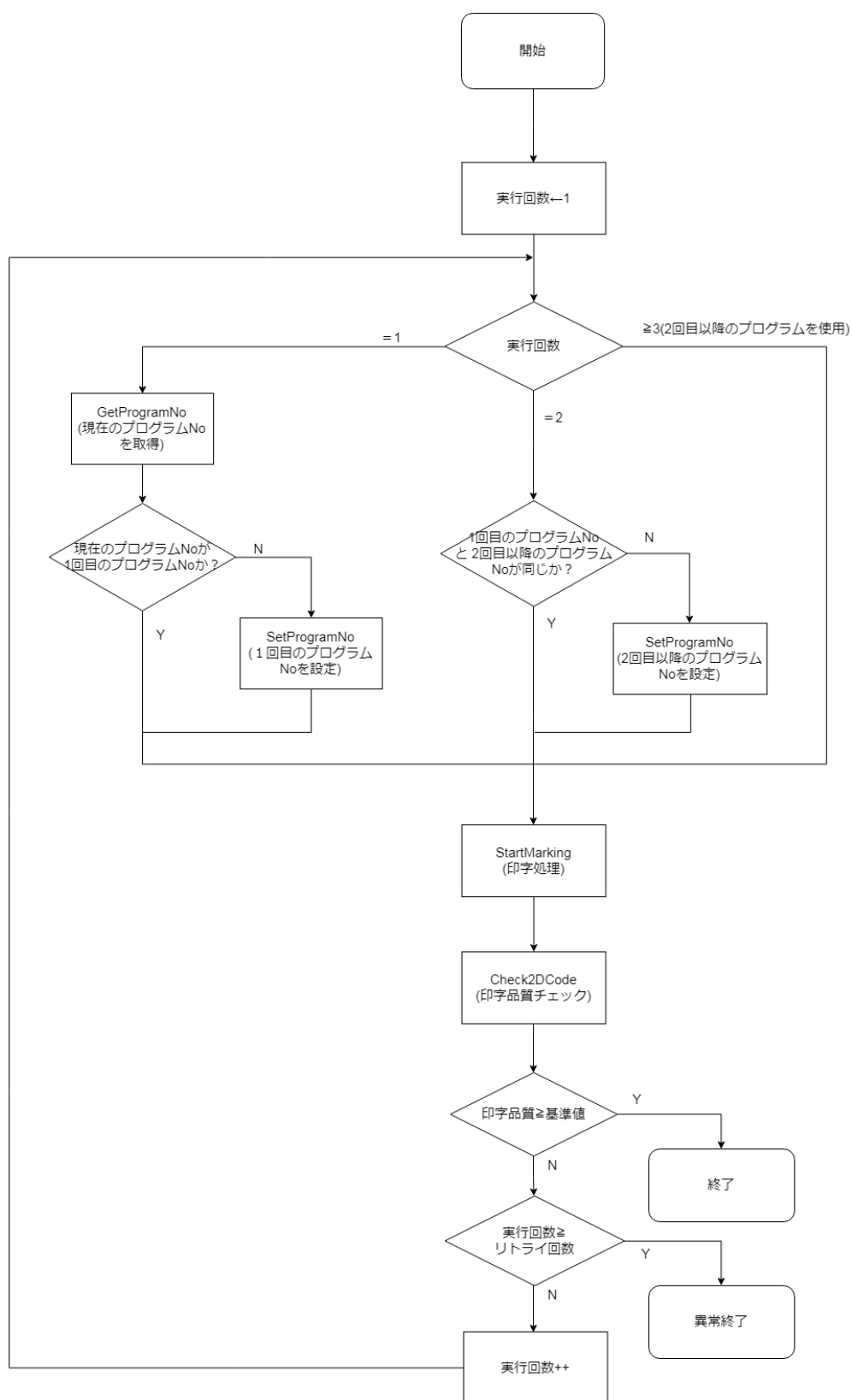


図 2-4 StartRetryMarking のフローチャート

2.2.6.35. CaoController : Execute("GetRetryMarking")コマンド

StartRetryMarking コマンドと組み合わせて使用し, StartRetryMarking コマンドの結果を取得します. 印字が終了していない場合, 印字が終了するまで待機します. 事前に非同期処理で StartRetryMarking コマンドを使用していない場合, エラーを返します.

引数の型	説明
なし	-

戻り値の型	説明
VT_ARRAY VT_VARIANT	-
0	VT_BSTR 最後に読みとった AIM-DPM 総合グレード(A~D/F) 値範囲:A~D / F / (空白) * 印字品質を 1 度も判定していない場合, (空白)が格納されます.
1	VT_BSTR 印字した文字列の読み取り結果 例)文字列が「test」の QR コードを印字した場合, 「test」が格納されます.
2	VT_UI1 印字を実行した回数. 値範囲:1~255
3	VT_BSTR エラーが発生した関数名が格納されます. 途中でエラーが発生した場合, どのコマンドで失敗したかが分かります. ただし, エラーが発生していない場合, (空白)が格納されます. 例)GetProgramNo コマンドで失敗した場合, 「GetProgramNo」が格納されます.
4	VT_I4 エラーコードを示します. 0 : 成功 0 以外 : エラー (エラー内容は表 2-6 固有エラーコードを参照してください)

使用例(CaoScript)

‘1 回目で使用するプログラム No:100

‘1 回目の印字タイムアウト:2000 ms

‘2 回目以降で使用するプログラム No:101

‘2 回目以降の印字タイムアウト:5000 ms

‘指定するブロック No:255

‘印字品質の基準:A 以上

‘繰り返し最大実行回数:10 回

```
result = ctrl.Execute("StartRetryMarking",Array(100,2000,101,5000,255,"A",10))
```

‘ここに印字と並行して行う動作を記述

.....

```
result = ctrl.Execute("GetRetryMarking","")
```

```
Dim vntResult
```

```
vntResult = DAT.ToVar(result)
```

```
dbg.output vntResult(0)          ‘⇒表示結果例:「B」
```

```
dbg.output vntResult(1)          ‘⇒表示結果例:「test」
```

```
dbg.output vntResult(2)          ‘⇒表示結果例:「5」
```

```
dbg.output vntResult(3)          ‘⇒表示結果例:「Check2DCode2」
```

```
dbg.output Hex(vntResult(4))     ‘⇒表示結果例:「80110001」(エラーコード:S001)
```

2.2.6.36. CaoController: :Execute(“SendCommand”)コマンド

「LaserMarker 通信マニュアル」に記載されており、本プロバイダで実装されていないコマンドを使用したい場合に使用します。

引数の型	説明
VT_BSTR	LaserMarker 通信マニュアルの通信コマンドを参照し、送信したいデータコマンドを入力します。 例)印字開始受付を取得したい場合。 " RX,TriggerLock "

戻り値の型	説明
VT_BSTR	受信したデータコマンド 例) 送信した" RX,TriggerLock "コマンドが成功した場合 "RX,OK,0" (0:印字開始可能, 1:印字開始禁止)

使用例(CaoScript)

```
result = ctrl.Execute("SendCommand","RX,TriggerLock")
```

```
dbg.output result                ‘⇒表示結果例:「RX,OK,0」
```

2.3. エラーコード

本プロバイダでは固有エラーコードを定義しています。固有エラーコードを表 2-6 に示します。ORiN2 共通エラーについては、「ORiN2 プログラミングガイド」のエラーコードの章を参照してください。

表 2-6 固有エラーコード

エラー番号	説明
0x8011xxxx	レーザマーカのエラーコードで「S〇〇〇」から始まるエラーです。下 4 桁「xxxx」を 10 進数に変換した値が「〇〇〇」に対応しています。使用するユーザーズマニュアルからエラー原因を調査してください。 ⁷
0x8012 xxxx	レーザマーカのエラーコードで「E〇〇〇」から始まるエラーです。下 4 桁「xxxx」を 10 進数に変換した値が「〇〇〇」に対応しています。使用するユーザーズマニュアルからエラー原因を調査してください。 ⁷
0x8013 xxxx	レーザマーカのエラーコードで「T〇〇〇」から始まるエラーです。下 4 桁「xxxx」を 10 進数に変換した値が「〇〇〇」に対応しています。使用するユーザーズマニュアルからエラー原因を調査してください。 ⁷
0x8014 xxxx	レーザマーカのエラーコードで「W〇〇〇」から始まるエラーです。下 4 桁「xxxx」を 10 進数に変換した値が「〇〇〇」に対応しています。使用するユーザーズマニュアルからエラー原因を調査してください。 ⁷
0x80150000	レーザマーカのエラーコードで S, E, T, W 以外から始まるエラーです。WireShark などに応答パケットをキャプチャし、使用するユーザーズマニュアルからエラー原因を調査してください。
0x80160000	StartMarking コマンドを使用する前に GetStartMarking コマンドを使用した、または、StartRetryMarking コマンドを使用する前に GetStartMarking コマンドを使用した場合に出力されるエラーコードです。
0x80170000	StartRetryMarking コマンドで、実行回数が設定した繰り返し最大実行回数を上回った場合に出力されるエラーコードです。
0x80180000	StartMarking(非同期)、StartRetryMarking が動作しているときに他のコマンドを使用した場合に出力されるエラーです。AddController 時の Timeout を長めに設定することで、設定値分 StartMarking(非同期)および StartRetryMarking の終了を待機できます。
0x80190000	StartRetryMarking コマンド実行中に ProviderCancel コマンドを実行した場合に出力されるエラーです。再度 StartRetryMarking コマンドを実行する場合は、ProviderClear コマンドを実行してください。

⁷ 下位 4 桁が「FFFF」の場合、不正な応答パケットを受信した可能性があります。WireShark などに応答パケットをキャプチャし、使用するユーザーズマニュアルからエラー原因を調査してください。

0x801A0000	不正なパケットを受信したときに出力されるエラーコードです。WireSharkなどで応答パケットをキャプチャし、使用するユーザーズマニュアルからエラー原因を調査してください。
------------	--

3. 通信プロトコルコマンド対応表

本プロバイダで実装している Execute メソッド、Variable 変数と LaserMarker 通信マニュアルの通信コマンドの対応表を表 3-1 に示します。

表 3-1 通信コマンド対応表

Execute メソッド	変数名	get/put	通信コマンド
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
GetShutterOperating	@SHUTTER_OPERATING	get	[RX]ShutterOperatingCount

Count	G_TIME		
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