

# MOTHER 用 DLL 関数仕様書

Rev 1.0

株式会社アコン

## 1) 概要

お客様のアプリケーション内で『MOTHER 用 DLL』を参照、利用することで、DLL 関数を通してアコン製レーザーマーカ-専用ソフト（Mother：マザー）を直接操作することが可能となります。

## 2) 対象アプリケーション

.NET Framework4.8 で作成されたアプリケーション ※その他のバージョンの場合はお問い合わせ下さい

## 3) 本 DLL の種類

C#ライブラリ（DLL）

## 4) 主な利用方法

- ・PC モードで必ず行って下さい。
- ・お客様側アプリケーションで MotherDLL.dll をインポート
- ・clsMother クラスをインスタンス化
- ・初期化处理（Init メソッド）
- ・予め作成した既存 mth ファイルを Load（FileLoad メソッド）
- ・各オブジェクトのオブジェクト番号を指定して個別操作（Move、SetTextString など）

```
clsMother mother = new clsMother();  
int ret;  
ret = mother.Init();  
ret = mother.FileLoad("d:¥¥text.mth");  
ret = mother.Move(0, 10, 20);
```

## 5) オブジェクトの種類

Mother 内のオブジェクトは表の通りです。

アイコン	オブジェクト名	主なメソッド
	直線	Move、 ChangeSize など
	連続線	Move、 ChangeSize など
	曲線	Move、 ChangeSize など
	四角形	Move、 ChangeSize など
	円	Move、 ChangeSize など
	楕円	Move、 ChangeSize など
	点	Move、 SetPointOutputTime など
	テキスト	Move、 SetTextStringSetTextFontKind など
	バーコード	Move、 SetBarcodeString など
	BMP(jpg,gif,png)	Move、 ChangeSize 、 ImageLoad など
	DXF	Move、 ChangeSize など

## 目次

初期化 .....	7
【Init】DLL の初期化.....	7
ファイル操作.....	8
【FileLoad】Mother ファイルの読み込み .....	8
オブジェクト操作／共通 .....	9
【Move】オブジェクトの移動.....	9
【ChangeSize】オブジェクトのサイズ変更.....	10
【ObjectCount】オブジェクトの全体数を取得 .....	11
ペン操作 .....	12
【GetObjectPenNumber】オブジェクトのペン番号を取得.....	12
【GetPenSpeed】ペンのスピードを取得.....	13
【SetPenSpeed】ペンのスピードを変更.....	14
【GetPenPower】ペンのパワーを取得 .....	15
【SetPenPower】ペンのパワーを変更.....	16
【GetPenFreq】ペンの周波数を取得 .....	17
【SetPenFreq】ペンのパワーを変更.....	18
【GetPenPulse】ペンのパルス幅を取得.....	19

【SetPenPulse】ペンのパルス幅を変更 .....	20
【GetPenCount】ペンの印字回数を取得 .....	21
【SetPenCount】ペンの印字回数を変更 .....	22
【GetPenJumpSpeed】ペンのジャンプスピードを取得 .....	23
【SetPenJumpSpeed】ペンのジャンプスピードを変更 .....	24
【GetPenJumpDeleyMax】ペンのジャンプ大移動後の待ち時間を取得 .....	25
【SetPenJumpDeleyMax】ペンのジャンプ大移動後の待ち時間を変更 .....	26
【GetPenJumpDeleyMin】ペンのジャンプ小移動後の待ち時間を取得 .....	27
【SetPenJumpDeleyMin】ペンのジャンプ小移動後の待ち時間を変更 .....	28
【GetPenDeleyStart】ペンの出力開始遅延時間を取得 .....	29
【SetPenDeleyStart】ペンの出力開始遅延時間を変更 .....	30
【GetPenDeleyLaseroff】ペンの出力停止前遅延時間を取得 .....	31
【SetPenDeleyLaseroff】ペンの出力停止前遅延時間を変更 .....	32
【GetPenDeleyEnd】ペンの出力停止後遅延時間を取得 .....	33
【SetPenDeleyEnd】ペンの出力停止前遅延時間を変更 .....	34
<b>オブジェクト操作／塗潰し .....</b>	<b>35</b>
【SetHatchKind】塗潰しのタイプを設定 .....	35
【SetHatchPen】塗潰しのペン番号を変更 .....	36

【SetHatchPattern】塗潰しのパターンを設定 .....	37
【SetHatchAngle】塗潰しの線の角度を設定 .....	38
【SetHatchPitch】塗潰しの線のピッチ幅を設定 .....	39
【SetHatchDir】塗潰しの線の印字方向を設定 .....	40
<b>オブジェクト操作／テキスト.....</b>	<b>41</b>
【SetTextString】テキストオブジェクトの文字列を設定 .....	41
【SetTextFontKind】テキストオブジェクトのフォント種類を設定 .....	42
【SetTextCharHeight】テキストオブジェクトの文字高さを設定 .....	43
【SetTextCharWidthPer】テキストオブジェクトの文字幅を設定 .....	44
【SetTextBetween】テキストオブジェクトの文字間隔を設定.....	45
<b>オブジェクト操作／ポイント .....</b>	<b>46</b>
【SetPointOutputTime】ポイントオブジェクトの照射時間を設定 .....	46
<b>オブジェクト操作／バーコード.....</b>	<b>47</b>
【SetBarcodeString】バーコードオブジェクトの文字列を設定 .....	47
【SetQRcodeParameter】バーコードオブジェクト(QR)のパラメータを設定 .....	48
【SetDataMatrixParameter】バーコードオブジェクト(DMC)のパラメータを設定.....	49
<b>オブジェクト操作／BMP .....</b>	<b>50</b>
【ImageLoad】BMP オブジェクトのファイルを設定.....	50

<b>印字</b> .....	<b>51</b>
【MarkingStart】印字を開始する .....	51
【MarkingStop】印字を停止する .....	53
【MarkingState】印字状態のステータスを取得 .....	54
【MarkingEnd】印字終了をイベントで通知する .....	55
<b>ガイド光</b> .....	<b>56</b>
【GuideStart】ガイド光の出力を開始する .....	56
【GuideStop】ガイド光を停止する .....	58
【GuideEndState】ガイド光の出力完了を取得 .....	59
<b>状態取得</b> .....	<b>60</b>
【GetLaserState】レーザー装置の状態を取得 .....	60

## 初期化

### 【Init】DLL の初期化

---

DLL を初期化します。必ず最初に実行して下さい。

```
public int Init()
```

#### パラメータ

変数	内容	備考
なし		

#### 戻り値

結果	内容	備考
成功	0	
失敗	-1	



## ファイル操作

### 【FileLoad】Mother ファイルの読み込み

---

mth ファイルのファイルパスを指定し Mother にデータを展開します。

```
public int FileLoad(string path)
```

#### パラメータ

変数	内容	備考
Path	Mth ファイルの保存されたパスを含むファイル名	

#### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## オブジェクト操作 / 共通

### 【Move】オブジェクトの移動

---

x、y 座標を指定してオブジェクトを移動します。

```
public int Move(int obj_number, double x, double y)
```

#### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	
x	X 座標 (mm)	
y	Y 座標 (mm)	

#### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【ChangeSize】オブジェクトのサイズ変更

---

オブジェクトのサイズ（幅、高さ）を変更します。

```
public int ChangeSize(int obj_number, double width, double height)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	テキストオブジェクトは未対応（SetTextCharHeight などを利用
width	オブジェクトの幅（mm）	
height	オブジェクトの高さ（mm）	

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【ObjectCount】オブジェクトの全体数を取得

---

オブジェクトの全体数（総数）を取得します。

```
public int ObjectCount()
```

### パラメータ

変数	内容	備考
なし		

### 戻り値

結果	内容	備考
成功	0~255	オブジェクトの総数
失敗	-1	

## ペン操作

### 【GetObjectPenNumber】オブジェクトのペン番号を取得

---

指定したオブジェクトに設定されているペンの番号（0~31）を取得します。

```
public int GetObjectPenNumber(int obj_number)
```

#### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	

#### 戻り値

結果	内容	備考
成功	0~31	ペン番号
失敗	-1	

## 【GePenSpeed】ペンのスピードを取得

---

指定したペンのスピード値を取得します。

```
public int GetPenSpeed(int pen_number)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	

### 戻り値

結果	内容	備考
成功	1~10000	スピード値(mm/s)
失敗	-1	

## 【SetPenSpeed】ペンのスピードを変更

---

指定したペンのスピード値を変更します。

```
public int SetPenSpeed(int pen_number, int speed)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
speed	スピード値(mm/s)	1~10000

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【GetPenPower】ペンのパワーを取得

---

指定したペンの POWER 値を取得します。

```
public int GetPenPower(int pen_number)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	

### 戻り値

結果	内容	備考
成功	0~100	パワー値 (%)
失敗	-1	



## 【SetPenPower】ペンのパワーを変更

---

指定したペンの POWER 値を変更します。

```
public int SetPenPower(int pen_number, int power)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
power	パワー値 (%)	0~100

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【GetPenFreq】ペンの周波数を取得

---

指定したペンの周波数を取得します。

```
public int GetPenFreq(int pen_number)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	

### 戻り値

結果	内容	備考
成功	0~4000	周波数 (kHz)
失敗	-1	

## 【SetPenFreq】ペンのパワーを変更

---

指定したペンの周波数を変更します。

```
public int SetPenFreq(int pen_number, int freq)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
freq	周波数 (kHz)	0~4000

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【GetPenPulse】ペンのパルス幅を取得

---

指定したペンのパルス幅を取得します。

```
public int GetPenPulse(int pen_number)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	

### 戻り値

結果	内容	備考
成功	1~350	パルス幅 (ns, $\mu$ s) ※ファイバーレーザーは ns 単位、UV レーザーは $\mu$ s 単位
失敗	-1	

## 【SetPenPulse】ペンのパルス幅を変更

---

指定したペンのパルス幅を変更します。

```
public int SetPenPulse(int pen_number, int pulse)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
pulse	パルス幅 (ns, $\mu$ s)	1~350 ※ファイバーレーザーは ns 単位、UV レーザーは $\mu$ s 単位

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【GetPenCount】ペンの印字回数を取得

---

指定したペンの印字回数を取得します。

```
public int GetPenCount(int pen_number)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	

### 戻り値

結果	内容	備考
成功	1~255	印字回数 (回)
失敗	-1	

## 【SetPenCount】ペンの印字回数を変更

---

指定したペンの印字回数を変更します。

```
public int SetPenCount(int pen_number, int count)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
count	印字回数 (回)	1~255

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【GetPenJumpSpeed】ペンのジャンプスピードを取得

---

指定したペンのジャンプスピードを取得します。

```
public int GetPenJumpSpeed(int pen_number)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	

### 戻り値

結果	内容	備考
成功	1~10000	ジャンプスピード (mm/s)
失敗	-1	



## 【SetPenJumpSpeed】ペンのジャンプスピードを変更

---

指定したペンのジャンプスピードを変更します。

```
public int SetPenJumpSpeed(int pen_number, int jump_speed)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
Jump_speed	ジャンプスピード (mm/s)	1~10000

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【GetPenJumpDeleyMax】ペンのジャンプ大移動後の待ち時間を取得

---

指定したペンのジャンプ大移動後の待ち時間を取得します。

```
public int GetPenJumpDeleyMax(int pen_number)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	

### 戻り値

結果	内容	備考
成功	0~1000	待ち時間 (μs)
失敗	-1	

## 【SettPenJumpDeleyMax】ペンのジャンプ大移動後の待ち時間を変更

---

指定したペンのジャンプ大移動後の待ち時間を変更します。

```
public int SettPenJumpDeleyMax(int pen_number, int deley_time)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
deley_time	待ち時間 (μs)	0~1000

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【GetPenJumpDeleyMin】ペンのジャンプ小移動後の待ち時間を取得

---

指定したペンのジャンプ小移動後の待ち時間を取得します。

```
public int GetPenJumpDeleyMin(int pen_number)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	

### 戻り値

結果	内容	備考
成功	0~1000	待ち時間 (μs)
失敗	-1	

## 【SettPenJumpDeleyMin】ペンのジャンプ小移動後の待ち時間を変更

---

指定したペンのジャンプ小移動後の待ち時間を変更します。

```
public int SettPenJumpDeleyMin(int pen_number, int deley_time)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
deley_time	待ち時間 (μs)	0~1000

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【GetPenDeleyStart】ペンの出力開始遅延時間を取得

---

指定したペンの出力開始遅延時間を取得します。本時間はマイナス値もあるため out 引数で取得します。

```
public int GetPenDeleyStart(int pen_number, out int time)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
out time	出力開始遅延時間（戻り値）	-100~100

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【SettPenDeleyStart】ペンの出力開始遅延時間を変更

---

指定したペンの出力開始遅延時間を変更します。マイナスの値も設定することができます。

```
public int SettPenDeleyStart(int pen_number, int time)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
time	出力開始遅延時間 (μs)	-100~100

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【GetPenDeleyLaseroff】ペンの出力停止前遅延時間を取得

---

指定したペンの出力停止前遅延時間を取得します。

```
public int GetPenDeleyLaserOff(int pen_number)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	

### 戻り値

結果	内容	備考
成功	0~1000	待ち時間 (μs)
失敗	-1	



## 【SettPenDeleyLaseroff】ペンの出力停止前遅延時間を変更

---

指定したペンの出力停止前遅延時間を変更します。

```
public int SettPenDeleyLaserOff(int pen_number, int time)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
time	待ち時間 (μs)	0~1000

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【GetPenDeleyEnd】ペンの出力停止後遅延時間を取得

指定したペンの出力停止後遅延時間を取得します。

```
public int GetPenDeleyEnd(int pen_number)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	

### 戻り値

結果	内容	備考
成功	0~1000	待ち時間 (μs)
失敗	-1	

## 【SettPenDeleyEnd】ペンの出力停止前遅延時間を変更

---

指定したペンの出力停止前遅延時間を変更します。

```
public int SettPenDeleyEnd (int pen_number, int time)
```

### パラメータ

変数	内容	備考
pen_number	ペン番号	
time	待ち時間 (μs)	0~1000

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## オブジェクト操作 / 塗潰し

### 【SetHatchKind】塗潰しのタイプを設定

指定したオブジェクトの塗潰しタイプを設定します。

```
public int SetHatchKind(int obj_number, HatchKinds hatch_kind)
```

#### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	・点、直線、BMP オブジェクトは除外 ・グループは「グループ単位でペン、塗りつぶし設定を行う」にチェックされている場合のみ有効
hatch_kind	塗潰し種類	0 : HatchKinds.Autoline アウトラインのみ 1 : HatchKinds.AutolineAndHatch アウトライン+塗潰し 2 : HatchKinds.OnlyHatch 塗潰しのみ

#### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【SetHatchPen】塗潰しのペン番号を変更

---

指定したオブジェクトの塗潰しのペン番号を変更します。

```
public int SetHatchPen(int obj_number, int hatchpen_number)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	・点、直線、BMP オブジェクトは除外 ・グループは「グループ単位でペン、塗りつぶし設定を行う」にチェックされている場合のみ有効
hatchpen_number	塗潰しのペン番号	0~31

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【SetHatchPattern】塗潰しのパターンを設定

---

指定したオブジェクトの塗潰しパターンを設定します。

```
public int SetHatchPattern(int obj_number, HatchPatterns hatch_pattern)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	・点、直線、BMP オブジェクトは除外 ・グループは「グループ単位でペン、塗りつぶし設定を行う」にチェックされている場合のみ有効
hatch_pattern	塗潰しパターン	0 : HatchPatterns.Single 単線 1 : HatchPatterns.Cross クロス線

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【SetHatchAngle】塗潰しの線の角度を設定

---

指定したオブジェクトの塗潰しの線の角度（°）を変更します。

```
public int SetHatchAngle(int obj_number, double angle)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	・点、直線、BMP オブジェクトは除外 ・グループは「グループ単位でペン、塗りつぶし設定を行う」にチェックされている場合のみ有効
angle	塗潰しの線の角度	0~360

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【SetHatchPitch】塗潰しの線のピッチ幅を設定

---

指定したオブジェクトの塗潰しの線のピッチ幅(mm)を設定します。

```
public int SetHatchPitch(int obj_number, double hatch_pitch)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	・点、直線、BMP オブジェクトは除外 ・グループは「グループ単位でペン、塗りつぶし設定を行う」にチェックされている場合のみ有効
hatch_pitch	塗潰しの線のピッチ幅	mm

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	



## 【SetHatchDir】塗潰しの線の印字方向を設定

---

指定したオブジェクトの塗潰しの線の印字方向を変更します。

```
public int SetHatchDir(int obj_number, HatchDirs hatch_dir)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	・点、直線、BMP オブジェクトは除外 ・グループは「グループ単位でペン、塗りつぶし設定を行う」にチェックされている場合のみ有効
angle	塗潰しの線の印字方向	0 : HatchDirs.Right 左→右 1 : HatchDirs.Left 右→左 2 ; HatchDirs.Right_Left 左→右、右→左を交互 3 : HatchDirs.Left_Right 右→左、左→右を交互

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## オブジェクト操作 / テキスト

### 【SetTextString】テキストオブジェクトの文字列を設定

---

指定したテキストオブジェクト文字列を設定します。

```
public int SetTextString(int obj_number, string str)
```

#### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	テキストオブジェクトのみ有効
str	文字列	

#### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【SetTextFontKind】テキストオブジェクトのフォント種類を設定

---

指定したテキストオブジェクトのフォント種類（TrueType、ストロークフォント、オリジナルフォント）を変更します。

TrueType の場合はフォント名、Blod、Italic を設定します。オリジナルフォントの場合はオリジナルフォントのファイル名（拡張子は除く）を設定します。

```
public int SetTextFontKind(int obj_number, FontTypes font_type, string font_name = "", bool bold = false, bool italic = false)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	テキストオブジェクトのみ有効
fontType	フォント種類	0 : FontTypes.Stroke ストロークフォント 1 : FontTypes.TrueType TrueType フォント 2 ; FontTypes.Orijinal オリジナルフォント
font_name	フォント名	(ストロークフォントは引数省略可)
bold	太文字	TrueType フォントの場合のみ有効 (引数省略可)
Italic	斜体	TrueType フォントの場合のみ有効 (引数省略可)

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【SetTextCharHeight】テキストオブジェクトの文字高さを設定

---

指定したテキストオブジェクトの文字高さ(mm)を設定します。

```
public int SetTextCharHeight(int obj_number, double text_height)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	テキストオブジェクトのみ有効
text_height	文字高さ	m m

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【SetTextCharWidthPer】テキストオブジェクトの文字幅を設定

---

指定したテキストオブジェクトの文字幅(%)を設定します。

```
public int SetTextCharWidthPer(int obj_number, double text_width_per)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	テキストオブジェクトのみ有効
text_width_per	文字幅 (%)	1~500(%) 標準幅=100% 100 より小さければ文字幅が縮み、大きければ横長になります。

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【SetTextBetween】テキストオブジェクトの文字間隔を設定

---

指定したテキストオブジェクトの文字間隔を設定します。

```
public int SetTextBetween(int obj_number, BetweenNormalTypes between_n_type, double sub_param)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	テキストオブジェクトのみ有効
between_n_type	文字間隔のタイプ	0 : BetweenNormalTypes.BetweenEdge 文字端 1 : BetweenNormalTypes.BetweenPitch 文字ピッチ 2 : BetweenNormalTypes.BetweenTotalLength 全体長
sub_param	各タイプ別の値(mm)	

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## オブジェクト操作／ポイント

### 【SetPointOutputTime】ポイントオブジェクトの照射時間を設定

---

指定したポイントオブジェクトのレーザー照射時間（ms）を設定します。  
指定した時間だけレーザーを照射し続けます。

```
public int SetPointOutputTime(int obj_number, double out_time)
```

#### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	ポイントオブジェクトのみ有効
out_time	照射時間	m s

#### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## オブジェクト操作／バーコード

### 【SetBarcodeString】バーコードオブジェクトの文字列を設定

---

指定したバーコードオブジェクトの文字列を設定します。  
文字列を変更すると自動的にバーコードが再生成されます。

```
public int SetBarcodeString(int obj_number, string text)
```

#### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	バーコードオブジェクトのみ有効
text	文字列	

#### 戻り値

結果	内容	備考
成功	0	
失敗	-1	



## 【SetQRcodeParameter】バーコードオブジェクト(QR)のパラメータを設定

---

指定したバーコードオブジェクトで QR コードの場合の各種パラメータを設定します。  
設定を変更すると自動的にバーコードが再生成されます。

```
public int SetQRcodeParameter(int obj_number, double cellsize, QRErrTypes errtype, QRVersions version)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	バーコードオブジェクトで QR コードの場合のみ有効
cellsize	セルサイズ	mm
errtype	誤り訂正率	0 : QRErrTypes.L 誤り訂正能力約 7% 1 : QRErrTypes.M 誤り訂正能力約 15% 2 : QRErrTypes.Q 誤り訂正能力約 25% 3 : QRErrTypes.H 誤り訂正能力約 30%
version	バージョン	0 : QRVersions.AUTO 自動 1~40 : QRVersions.VERSION1~VERSION40

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【SetDataMatrixParameter】バーコードオブジェクト(DMC)のパラメータを設定

---

指定したバーコードオブジェクトで DataMatrix (DMC) の場合の各種パラメータを設定します。設定を変更すると自動的にバーコードが再生成されます。

```
public int SetDataMatrixParameter(int obj_number, double cellsize, DmcSymbols symbol)
```

### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	バーコードオブジェクトで DMC の場合のみ有効
cellsize	セルサイズ	mm
symbol	シンボル	0 : DmcSymbols.AUTO 自動 1~30 : DmcSymbols.SYMBOL10x10 ~ SYMBOL16x48

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## オブジェクト操作／BMP

### 【ImageLoad】BMP オブジェクトのファイルを設定

---

指定した BMP オブジェクトのファイル名をフルパスで設定します。  
ファイルは bmp,png,jpeg,gif に対応しています。

```
public int ImageLoad(int obj_number, string path)
```

#### パラメータ

変数	内容	備考
obj_number	オブジェクト番号	BMP オブジェクトのみ有効
path	ファイル名	パスも含む

#### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

# 印字

## 【MarkingStart】印字を開始する

レーザーを出力し印字を開始します。  
※印字開始後、印字中にレーザーマーカと通信が発生する操作を行うと印字は停止されます。

```
public int MarkingStart(int marking_mode)
```

### パラメータ

変数	内容	備考
marking_mode	印字モード	0：通常（1回で終了） 1：連続（STOPされるまで連続印字を繰り返す）

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	通信異常、オブジェクトデータなし、その他
	-2	ペンデータの転送に失敗
	-3	オブジェクトデータの転送に失敗
	-4	インターロック中
	-5	非常停止ボタン押下中 ※機種による
	-6	コンソール非常停止ボタン押下中 ※機種による
	-7	レーザー発振器異常
	-8	シュミレーション中
	-9	レーザー停止中 ※機種による
	-10	シャッター閉じ中 ※機種による
	-11	レーザーコア起動中
	-12	簡易レーザー停止信号中
	-13	トリガーロック中
	-14	印字範囲逸脱中
	-15	プログラム展開エラー中
	-16	時計電池異常中（PCレスモードのみ）
	-17	システムエラー中
	-18	品種未選択で印字開始（PCレスモード時）中

-19	I/O で印字トリガー中
-20	I/O で印字中
-21	Mother から印字中
-22	IO でガイド光操作中
-23	Mother からガイド光中
-24	Mother でシミュレーション中
-25	リモートコマンド、またはコンソールからガイド光中
-26	リモートコマンド、またはコンソールから印字中
-27	READY 信号 OFF 中
-28	ホスト PC に印字開始要求中 (PC モードのみ)
-29	ホスト PC にガイド開始要求中 (PC モードのみ)
-30	プログラム展開中
-31	運転停止中

## 【MarkingStop】印字を停止する

---

印字中であれば印字を停止します。

```
public int MarkingStop()
```

### パラメータ

変数	内容	備考
なし		

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【MarkingState】印字状態のステータスを取得

---

本メソッドをポーリングすることで印字中かどうか判断することができます。  
MarkingStart を行ってから本コマンドを利用して下さい。  
またイベントで印字完了を知りたい場合は、MarkingEnd イベントハンドラを利用して下さい。

```
public int MarkingState()
```

### パラメータ

変数	内容	備考
なし		

### 戻り値

結果	内容	備考
成功	0 または 1	0 : 印字完了、停止中 1 : 印字中
失敗	-1	

## 【MarkingEnd】印字終了をイベントで通知する

---

印字完了をイベントで取得したい場合に利用します。  
印字完了時のイベント関数を用意しイベントハンドラーとして設定します。

```
mother.MarkingEnd += OnMarkingEndReceived;
```

```
public void OnMarkingEndReceived(object sender, EventArgs e)
{
    MessageBox.Show("マーキング終了しました！");
}
```



## ガイド光

### 【GuideStart】ガイド光の出力を開始する

ガイド光のモードを指定して出力を開始します。  
※ガイド開始後、ガイド中にレーザーマーカと通信が発生する操作を行うとガイドは停止されます。

```
public int GuideStart(int guide_mode)
```

#### パラメータ

変数	内容	備考
guide_mode	ガイド光モード	0：四角形枠 1：輪郭 2：WD ガイド ※AL2000F-M20 のみ

#### 戻り値

結果	内容	備考
成功	0	
失敗	-1	通信異常、オブジェクトデータなし、その他
	-2	ペンデータの転送に失敗
	-3	オブジェクトデータの転送に失敗
	-4	インターロック中
	-5	非常停止ボタン押下中 ※機種による
	-6	コンソール非常停止ボタン押下中 ※機種による
	-7	レーザー発振器異常
	-8	シュミレーション中
	-9	レーザー停止中 ※機種による
	-10	シャッター閉じ中 ※機種による
	-11	レーザーコア起動中
	-12	簡易レーザー停止信号中
	-13	トリガーロック中
	-14	印字範囲逸脱中
	-15	プログラム展開エラー中
	-16	時計電池異常中 (PC レスモードのみ)
	-17	システムエラー中

-18	品種未選択で印字開始（PC レスモード時） 中
-19	I/O で印字トリガー中
-20	I/O で印字中
-21	Mother から印字中
-22	IO でガイド光操作中
-23	Mother からガイド光中
-24	Mother でシミュレーション中
-25	リモートコマンド、またはコンソールからガイド光中
-26	リモートコマンド、またはコンソールから印字中
-27	READY 信号 OFF 中
-28	ホスト PC に印字開始要求中（PC モードのみ）
-29	ホスト PC にガイド開始要求中（PC モードのみ）
-30	プログラム展開中
-31	運転停止中

## 【GuideStop】ガイド光を停止する

---

ガイド光を出力中であれば停止します。

```
public int GuideStop()
```

### パラメータ

変数	内容	備考
なし		

### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

## 【GuideEndState】ガイド光の出力完了を取得

---

本メソッドをポーリングすることでガイド光出力が停止されたかどうかを判断することができます。  
GuideStart を行ってから本メソッドを利用して下さい。

```
public int GuideEndState()
```

### パラメータ

変数	内容	備考
なし		

### 戻り値

結果	内容	備考
成功	0 または 1	0：ガイド光出力中 1：ガイド光停止中
失敗	-1	

## 状態取得

### 【GetLaserState】レーザー装置の状態を取得

レーザー装置の現在の状態を返します。

```
public int GetLaserState (out int danger, out int warning, out int other, out int ready, out int memory_number)
```

#### パラメータ

変数	内容	備考
out danger	異常アラームフラグ	下表参照 ビットフラグ（複数アラームあり）D00 はビット 0 が ON
out warning	注意アラームフラグ	下表参照 ビットフラグ（複数アラームあり）W00 はビット 0 が ON
out other	その他エラーフラグ	下表参照 ビットフラグ（複数アラームあり）N00 はビット 0 が ON
out ready	READY 状態	0 : READY OFF 状態 1:READY ON 状態
out memory_number	現在の品種№	0～1999、または 9999 ※9999 の場合は品種未選択状態 ※PC レスモードの場合のみ使用

#### 戻り値

結果	内容	備考
成功	0	
失敗	-1	

■異常アラームフラグ

エラー№	異常アラーム種別	エラー内容、解除方法
D00	レーザー発振器異常	レーザー発振器に異常が発生している可能性があります。何度も発生する場合は購入先に連絡して下さい。
D01	印字範囲逸脱	印字エリアをオーバーした場合に発生します。主に移動マーキングで発生する場合があります。エラー解除ボタンで解除することができます。
D02	印字データ展開エラー	PCレスモードで、レーザーマーカ側に転送した印字データを内部でデータ展開する際にメモリーエラーなどで発生する場合があります。印字データを再度転送するか、レーザー本体を再起動して再度試してみてください。
D03	内蔵時計エラー	内蔵時計に異常が発生している場合に表示されます。主に時計用のボタン電池が切れている可能性があります。販売元にご相談下さい。
D04	システムエラー	レーザー本体内部で予期せぬエラーが発生しました。レーザー本体を再起動して再度操作してみてください。
D05	品種メモリ№未選択	PCレスモードで、品種を未選択状態で1/Oで印字開始を実行した場合に発生します。
D06	ACK通信エラー	レーザー機器側から送信の際、Motherが受信応答がない場合に発生します。

■注意アラームフラグ

エラー№	異常アラーム種別	エラー内容、解除方法
W00	レーザーパワー低下	印字の度にレーザーパワーを測定し、予め設定したレーザーパワー閾値を下回った場合に警告表示されます。パワーメーターを内蔵している機種に限ります。(2024.5月時点未対応)

■その他のエラーフラグ

エラー№	異常アラーム種別	エラー内容、解除方法
N00	インターロック中	インターロック信号が入力されて緊急停止されている状態です。同時に異常アラームが出力されます。解除するにはインターロック信号をOFFにしてください。
N01	シャッター閉	シャッターが閉まっている時に発生します。I/Oのシャッター制御を行っている場合はシャッター制御I/OをONにしてください。またはシャッター閉じる原因となったエラーを取り除いて下さい。
N02	印字トリガーロック中	I/Oの印字トリガーロック信号がONになっている状態です。印字トリガーロック信号をOFFにしてください。
W03	レーザー停止中	I/Oのレーザー停止信号がONになっている状態です。レーザー停止信号をOFFにしてください。
W04	レーザー停止中 (簡易信号)	I/Oの簡易レーザー停止信号がONになっている状態です。簡易レーザー停止信号をOFFにしてください。

W05	非常停止ボタン押下（本体）	レーザーマーカ本体の非常停止ボタンが押された状態です。非常停止ボタンを元に戻して下さい。
W06	非常停止ボタン押下（コンソール）	別売りのコンソールの非常停止ボタンが押された状態です。コンソールの非常停止ボタンを元に戻して下さい。
W07	レーザー発振器スタンバイ中	レーザーマーカの起動準備中です。起動完了するまで待機して下さい。
W08	シミュレーション中	I/O モニタ画面でシミュレーション中です。
W09	品種データメモリ展開中	PC レスモード時に、品種データをメモリ展開している最中です。しばらくお待ちください。
W10	運転停止中	別売りのコンソール操作で運転停止中の場合に表示されます。コンソール操作で運転開始を行って下さい。

■改定履歴

Rev	日付	内容
1.0	2024.05.22	初版