

GT プロバイダ SUNX プログラマブル表示器

Version 1.0.0

ユーザーズ ガイド

January 23, 2015

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2015-01-23	初版.

目次

1. はじめに.....	4
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.....	6
2.2.1. CaoWorkspace::AddController メソッド.....	6
2.2.1.1. Conn パラメータ.....	8
2.2.2. CaoController::Execute メソッド.....	8
2.3. エラーコード.....	8
3. コマンドリファレンス.....	10
3.1. コントローラクラスの Execute コマンド.....	11
3.1.1. CaoController::Execute(“WDR”) コマンド.....	11
3.1.2. CaoController::Execute(“SRR”) コマンド.....	11
3.1.3. CaoController::Execute(“PRR”) コマンド.....	11
3.1.4. CaoController::Execute(“WRR”) コマンド.....	12
3.1.5. CaoController::Execute(“WDW”) コマンド.....	12
3.1.6. CaoController::Execute(“BDW”) コマンド.....	13
3.1.7. CaoController::Execute(“DDW”) コマンド.....	13
3.1.8. CaoController::Execute(“SDW”) コマンド.....	14
3.1.9. CaoController::Execute(“WRW”) コマンド.....	14
3.1.10. CaoController::Execute(“BRW”) コマンド.....	15
3.1.11. CaoController::Execute(“DRW”) コマンド.....	15
3.1.12. CaoController::Execute(“SRW”) コマンド.....	16
3.1.13. CaoController::Execute(“AllRead”) コマンド.....	16
3.1.14. CaoController::Execute(“FDR”) コマンド.....	17
3.1.15. CaoController::Execute(“FDW”) コマンド.....	17
3.1.16. CaoController::Execute(“CDR”) コマンド.....	18
3.1.17. CaoController::Execute(“CDW”) コマンド.....	18
3.1.18. CaoController::Execute(“LDR”) コマンド.....	19
3.1.19. CaoController::Execute(“LDW”) コマンド.....	19

1. はじめに

本書は、パナソニック デバイス SUNX 製 プログラマブル表示器 GT シリーズ用 CAO プロバイダである GT プロバイダのユーザーズガイドです。

GT プロバイダは、CAO のインタフェースを用いてプログラマブル表示器 GT シリーズとの通信を行います。

2. プロバイダの概要

2.1. 概要

GT プロバイダは、コントローラクラスでコマンドの生成及び送受信を行います。これにより、クライアントはGTのワードエリア/ビットエリア領域へ値の設定、取得を容易に操作することができます。

GTプロバイダのファイル形式はDLL(Dynamic Link Library)となっており、その詳細は表 2-1 のようになっています。

表 2-1 GT プロバイダ

ファイル名	CaoProvGT.dll
ProgID	CaoProv.SUNX.GT
レジストリ登録	regsvr32 CaoProvGT.dll
レジストリ登録の抹消	regsvr32 /u CaoProvGT.dll

<p>[CharEndianWord= <文字列のワードのデータ参照順位(エンディアン)>]</p>	<p>文字列のワードのデータ参照順位を設定します。 (デフォルト:ビッグエンディアン)</p> <p>Big:ビッグエンディアン(上位→下位) Little:リトルエンディアン(下位→上位)</p> <p>上記以外の文字列が設定された場合, Big:ビッグエンディアン(上位→下位)を設定します.</p>
--	---

※Mode=<通信モード>の設定では, GT の通信条件で設定した通信モードを設定する必要があります.

GT と異なる通信モードを設定した場合, 動作の保証はされません.

※Mode=<通信モード>の 2: 双方向1モード, 3: 双方向2モードは未実装の為使用できません.

※[CharEndianAddress=<文字列のアドレスのデータ参照順位(エンディアン)>]と,

[CharEndianWord=<文字列のワードのデータ参照順位(エンディアン)>]の設定は,

下記の GTWIN のオプション設定に合わせる必要があります.

部品: データ部品, データ形式: ASCII か文字列, オプション設定: データ参照順位のアドレスとワード.

GTWIN のオプション設定 - 文字列のエンディアンを設定

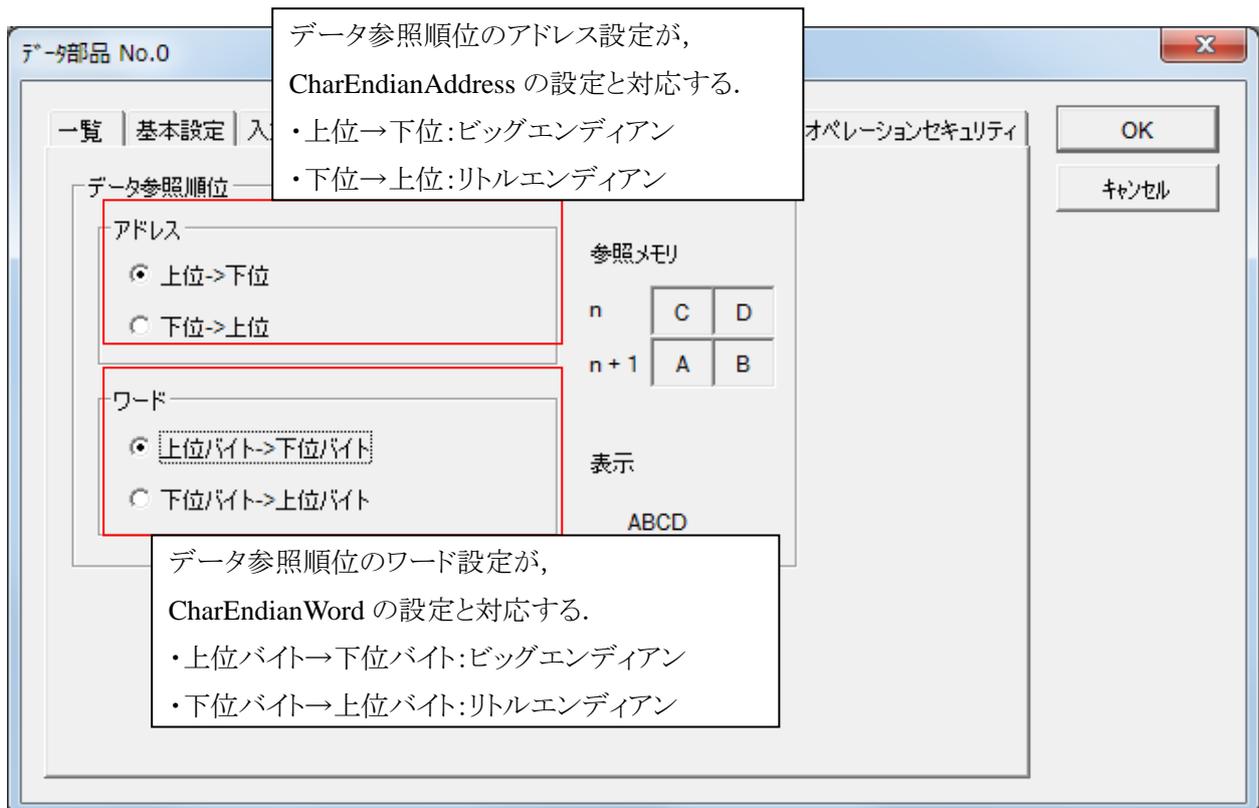


表 2-4 独自エラーコード一覧

エラー名	エラー番号	説明
E_COMMAND_ARG_NUM	0x80100001	コマンド引数の個数エラー
E_RANGE_OUTSIDE	0x80100002	コマンド引数の設定値の範囲エラー
E_RECV_MSG_SIZE	0x80100003	返信メッセージのサイズエラー
E_RECV_MSG_CMD	0x80100004	返信メッセージのコマンドコードエラー
E_RECV_MSG_BCC	0x80100005	返信メッセージの BCC エラー
E_TRANS_MODE_NOT_CMD	0x80100006	通信モード設定エラー
E_SEMAPHORE_TIMEOUT	0x80100007	セマフォのタイムアウトエラー
E_SEMAPHORE_TIMEOUT_EX	0x80100008	セマフォのタイムアウト以外のエラー
E_RECV_SIZE_ZERO	0x80100009	返信メッセージのサイズが 0
E_DISCONNECT	0x80100010	AddController 時に機器と未接続
E_RECV_MSG_ERR_BCC	0x80110000	返信メッセージのエラー: BCC
E_RECV_MSG_ERR_FRM	0x80110001	返信メッセージのエラー: コマンドのフォーマット
E_RECV_MSG_ERR_NO_CMD	0x80110002	返信メッセージのエラー: 使用している GT が対応外のコマンド
E_RECV_MSG_ERR_ADDRESS	0x80110003	返信メッセージのエラー: アドレス範囲外
E_RECV_MSG_ERR_BUF_OVER	0x80110004	返信メッセージのエラー: 受信バッファオーバー
E_RECV_MSG_ERR_READ	0x80110005	返信メッセージのエラー: 読み出しバッファオーバー
E_RECV_MSG_ERR_SET	0x80110006	返信メッセージのエラー: GT と相手機器の通信条件
E_RECV_MSG_ERR_WRITE	0x80110007	返信メッセージのエラー: 書き込み禁止アドレス
E_RECV_MSG_ERR_UNDEFIND	0x80110008	返信メッセージのエラー: 未定義

3. コマンドリファレンス

表 3-1 コントローラクラスの Execute コマンド一覧¹

コマンド	機能	
WDR	データエリアのワード単位読み出し	P. 11
SRR	リレーの単点読み出し	P. 11
PRR	リレーの複数点読み出し	P. 11
WRR	リレーのワード単位読み出し	P. 12
WDW	データエリアのワード単位書き込み	P. 12
BDW	データエリアのバイト単位書き込み	P. 13
DDW	データエリアのデジット単位書き込み	P. 13
SDW	データエリアのビット単位書き込み	P. 14
WRW	リレーのワード単位書き込み	P. 14
BRW	リレーのバイト単位書き込み	P. 15
DRW	リレーのデジット単位書き込み	P. 15
SRW	リレーのビット単位書き込み	P. 16
AllRead	全てのメッセージを受信	P. 16
FDR	データエリアの Float 単位読み出し	P. 17
FDW	データエリアの Float 単位書き込み	P. 17
CDR	データエリアの文字列読み出し	P. 18
CDW	データエリアの文字列書き込み	P. 18
LDR	データエリアの Long 単位読み出し	P. 19
LDW	データエリアの Long 単位書き込み	P. 19

¹ CDR/CDW は、GT の表示データ桁数を奇数で設定した場合、表示に使用されない領域は使用不可能とする

3.1. コントローラクラスの Execute コマンド

3.1.1. GaoController::Execute(“WDR”) コマンド

データエリア領域からワード単位分読み出します。

書式 “WDR”, (<WordAddress>, <WordNum>)

<WordAddress> : [in] ワードエリア先頭アドレス:0~2047

<WordNum> : [in] ワード数:1~2048

vntRet : [out]VT_I2|VT_ARRAY

引数が 2 個より少ない場合, エラー値を返す.

先頭アドレスの範囲が 0~2047 でない場合, エラー値を返す.

ワード数の範囲が 1~2048 でない場合, エラー値を返す.

先頭アドレス + ワード数 - 1 が 2047 を超える場合, エラー値を返す.

使用例 Execute(“WDR”, Array(0, 1)) アドレス 0 から 1 ワード分取得
Execute(“WDR”, Array(11, 10)) アドレス 11 から 10 ワード分取得

3.1.2. GaoController::Execute(“SRR”) コマンド

ビットエリア領域からビット単位分読み出します。

書式 “SRR”, (<BitAddress>, <BitPos>)

<BitAddress> : [in] ビットエリア指定アドレス:0~255

<BitPos> : [in] 指定ビット:0~15 (0: 下位~15: 上位)

vntRet : [out]VT_I2

引数が 2 個より少ない場合, エラー値を返す.

指定アドレスの範囲が 0~255 でない場合, エラー値を返す.

指定ビットの範囲が 0~15 でない場合, エラー値を返す.

使用例 Execute(“SRR”, Array(0, 0)) アドレス 0 から 0 ビット目を取得
Execute(“SRR”, Array(11, 11)) アドレス 11 から 11 ビット目を取得

3.1.3. GaoController::Execute(“PRR”) コマンド

ビットエリア領域からビット複数分読み出します。

書式 “PRR”, (<BitAddress1>, <BitPos1>, ..., <BitAddressN>, <BitPosN>)

<BitAddressN> : [in] ビットエリア指定アドレス:0~255
Nの最大:8
<BitPosN> : [in] 指定ビット:0~15(0:下位~15:上位)
Nの最大:8
vntRet : [out]VT_I2|VT_ARRAY

引数が2個より少ない場合, エラー値を返す.

引数が16個を超える場合, エラー値を返す.

引数が奇数の場合, 最後の指定アドレスが切り捨てられた結果を返す.

指定アドレスの範囲が0~255でない場合, エラー値を返す.

指定ビットの範囲が0~15でない場合, エラー値を返す.

使用例 Execute(“PRR”, Array(0, 0)) アドレス0から0ビット目を取得
Execute(“PRR”, Array(11, 1, 11)) アドレス11から1ビット目を取得

3.1.4. CaoController::Execute(“WRR”) コマンド

ビットエリア領域からワード単位分読み出します.

書式 “WRR”, (<BitAddress>, <WordNum>)

<BitAddress> : [in] ビットエリア先頭アドレス:0~255
<WordNum> : [in] ワード数:1~256
vntRet : [out]VT_I2|VT_ARRAY

引数が2個より少ない場合, エラー値を返す.

先頭アドレスの範囲が0~255でない場合, エラー値を返す.

ワード数の範囲が1~256でない場合, エラー値を返す.

先頭アドレス + ワード数 - 1 が255を超える場合, エラー値を返す.

使用例 Execute(“WRR”, Array(0, 1)) アドレス0から1ワード分取得
Execute(“WRR”, Array(11, 11)) アドレス11から11ワード分取得

3.1.5. CaoController::Execute(“WDW”) コマンド

ワードエリア領域にワード単位分書き込みます.

書式 “WDW” , (<WordAddress>, <WordData1>, <WordData2>, ... , <WordDataN>)

<WordAddress> : [in] ワードエリア先頭アドレス:0～2047
 <WordDataN> : [in] ワードデータ:-32768～32767
 Nの最大:2048
 vntRet : [out]VT_I2(正常:true, 異常:値なし)

引数が2個より少ない場合, エラー値を返す.

先頭アドレスの範囲が0～2047でない場合, エラー値を返す.

ワードデータの範囲が-32768～32767でない場合, エラー値を返す.

ワードデータの個数が, 1～2048でない場合, エラー値を返す.

先頭アドレス + データの個数 - 1 が 2047 を超える場合, エラー値を返す.

使用例 Execute(“WDW”, Array(0, 1)) アドレス0に1を書き込む
 Execute(“WDW”, Array(11, 1, 1234)) アドレス11に1, 続けてアドレス12に1234を書き込む

3.1.6. CaoController::Execute(“BDW”) コマンド

ワードエリア領域にバイト単位分書き込みます.

書式 “BDW” , (<WordAddress>, <BytePos>, <ByteData>)

<WordAddress> : [in] ワードエリア先頭アドレス:0～2047
 <BytePos> : [in] 上位下位指定:0 or 1 (0:下位バイト, 1:上位バイト)
 <ByteData> : [in] バイトデータ:-128～127
 vntRet : [out]VT_I2(正常:true, 異常:値なし)

引数が3個でない場合, エラー値を返す.

先頭アドレスの範囲が0～2047でない場合, エラー値を返す.

上位下位指定が0, 1以外の場合, エラー値を返す.

バイトデータの範囲が-128～127でない場合, エラー値を返す.

使用例 Execute(“BDW”, Array(0, 0, 0)) アドレス0の下位バイト位置に0を書き込む
 Execute(“BDW”, Array(11, 1, 12)) アドレス11の上位バイト位置に12を書き込む

3.1.7. CaoController::Execute(“DDW”) コマンド

ワードエリア領域にデジット単位分書き込みます.

書式 “DDW” , (<WordAddress>, <DigitPos>, <DigitData>)

<WordAddress> : [in] ワードエリア先頭アドレス:0~2047
 <DigitPos> : [in] デジット指定:0~3 (0:下位~3:上位)
 <DigitData> : [in] デジットデータ:0~15
 vntRet : [out]VT_I2(正常:true, 異常:値なし)

引数が3個でない場合, エラー値を返す.

先頭アドレスの範囲が0~2047でない場合, エラー値を返す.

デジット指定の範囲が0~3でない場合, エラー値を返す.

デジットデータの範囲が0~15でない場合, エラー値を返す.

使用例 Execute(“DDW”, Array(0, 0, 0)) アドレス0のデジット位置0に0を書き込む
 Execute(“DDW”, Array(11, 1, 12)) アドレス11のデジット位置1に12を書き込む

3.1.8. CaoController::Execute(“SDW”) コマンド

ワードエリア領域にビット単位分書き込みます.

書式 “SDW” , (<WordAddress>, <BitPos>, <BitData>)

<WordAddress> : [in] ワードエリア先頭アドレス:0~2047
 <BitPos> : [in] ビット指定:0~15 (0:下位~15:上位)
 <BitData> : [in] データ:0 or 1 (0:OFF, 1:ON)
 vntRet : [out]VT_I2(正常:true, 異常:値なし)

引数が3個でない場合, エラー値を返す.

先頭アドレスの範囲が0~2047でない場合, エラー値を返す.

ビット指定の範囲が0~15でない場合, エラー値を返す.

データが0 or 1でない場合, エラー値を返す.

使用例 Execute(“SDW”, Array(0, 0, 0)) アドレス0のビット位置0に0を書き込む
 Execute(“SDW”, Array(11, 1, 1)) アドレス11のビット位置1に1を書き込む

3.1.9. CaoController::Execute(“WRW”) コマンド

ビットエリア領域にワード単位分書き込みます.

書式 “WRW” , (<BitAddress>, <WordData1>, <WordData2>, ... , <WordDataN>)

<BitAddress> : [in] ビットエリア先頭アドレス:0~255
 <WordDataN> : [in] ワードデータ:-32768~32767
 Nの最大:256
 vntRet : [out]VT_I2(正常:true, 異常:値なし)

引数が2個でない場合, エラー値を返す.

先頭アドレスの範囲が0~255でない場合, エラー値を返す.

ワードデータの範囲が-32768~32767でない場合, エラー値を返す.

ワードデータの個数が, 1~256でない場合, エラー値を返す.

先頭アドレス + データの個数 - 1 が 255 を超える場合, エラー値を返す.

使用例 Execute("WRW", Array(0, 0)) アドレス0に0を書き込む
 Execute("WRW", Array(11, 1, 1234)) アドレス11に1, 続けてアドレス12に1234を書き込む

3.1.10. CaoController::Execute("BRW") コマンド

ビットエリア領域にバイト単位分書き込みます.

書式 "BRW", (<BitAddress>, <BytePos>, <ByteData>)

<BitAddress> : [in] ビットエリア先頭アドレス:0~255
 <BytePos> : [in] 上位下位指定:0 or 1 (0:下位バイト, 1:上位バイト)
 <ByteData> : [in] バイトデータ:-128~127
 vntRet : [out]VT_I2(正常:true, 異常:値なし)

引数が3個でない場合, エラー値を返す.

先頭アドレスの範囲が0~255でない場合, エラー値を返す.

上位下位指定が0 or 1以外の場合, エラー値を返す.

バイトデータの範囲が-128~127でない場合, エラー値を返す.

使用例 Execute("BRW", Array(0, 0, 0)) アドレス0の下位バイト位置に0を書き込む
 Execute("BRW", Array(11, 1, 12)) アドレス11の上位バイト位置に12を書き込む

3.1.11. CaoController::Execute("DRW") コマンド

ビットエリア領域にデジット単位分書き込みます.

書式 “DRW” , (<BitAddress>, <DigitPos>, <DigitData>)

<BitAddress> : [in] ビットエリア先頭アドレス:0~255
 <DigitPos> : [in] デジット指定:0~3 (0:下位~3:上位)
 <DigitData> : [in] デジットデータ:0~15
 vntRet : [out]VT_I2(正常:true, 異常:値なし)

引数が3個でない場合, エラー値を返す.

先頭アドレスの範囲が0~255でない場合, エラー値を返す.

デジット指定の範囲が0~3でない場合, エラー値を返す.

デジットデータの範囲が0~15でない場合, エラー値を返す.

使用例 Execute(“DRW”, Array(0, 0, 0)) アドレス0のデジット位置0に0を書き込む
 Execute(“DRW”, Array(11, 1, 12)) アドレス11のデジット位置1に12を書き込む

3.1.12. CaoController::Execute(“SRW”) コマンド

ビットエリア領域にビット単位分書き込みます.

書式 “SRW” , (<BitAddress>, <BitPos>, <BitData>)

<BitAddress> : [in] ビットエリア先頭アドレス:0~255
 <BitPos> : [in] ビット指定:0~15 (0:下位~15:上位)
 <BitData> : [in] データ:0 or 1 (0:OFF, 1:ON)
 vntRet : [out]VT_I2(正常:true, 異常:値なし)

引数が3個でない場合, エラー値を返す.

先頭アドレスの範囲が0~255でない場合, エラー値を返す.

ビット指定の範囲が0~15でない場合, エラー値を返す.

データが0 or 1でない場合, エラー値を返す.

使用例 Execute(“SRW”, Array(0, 0, 0)) アドレス0のビット位置0に0を書き込む
 Execute(“SRW”, Array(11, 1, 1)) アドレス11のビット位置1に1を書き込む

3.1.13. CaoController::Execute(“AllRead”) コマンド

全てのメッセージを受信します. 取得したメッセージなどは返しません.

書式 “AllRead”

戻り値 : なし

使用例 Execute(“AllRead”)

3.1.14. CaoController::Execute(“FDR”) コマンド

データエリア領域から 2 ワード分 FLOAT として読み出します。

書式 “FDR”, (<WordAddress>, <FloatNum>)

<WordAddress> : [in] ワードエリア先頭アドレス:0~2046
 <FloatNum> : [in] データ数(FLOAT の数):1~1024
 vntRet : [out]VT_R4|VT_ARRAY

引数が 2 個より少ない場合, エラー値を返す.

先頭アドレスの範囲が 0~2046 でない場合, エラー値を返す.

データ数の範囲が 1~1024 でない場合, エラー値を返す.

先頭アドレス + (データ数 - 1) * 2 が 2046 を超える場合, エラー値を返す.

使用例 Execute(“FDR”, Array(0, 1)) アドレス 0 から 1 データ分 (2 ワード) 取得
 Execute(“FDR”, Array(12, 10)) アドレス 12 から 10 データ分 (20 ワード) 取得

3.1.15. CaoController::Execute(“FDW”) コマンド

データエリア領域に 2 ワード分 FLOAT として書き込みます。

書式 “FDW”, (<WordAddress>, <FloatData1>, <FloatData2>, ... , <FloatDataN>)

<WordAddress> : [in] ワードエリア先頭アドレス:0~2046
 <FloatDataN> : [in] Float データ:-3.402823466e+38~3.402823466e+38
 N の最大:1024
 vntRet : [out]VT_I2 (正常:true, 異常:値なし)

引数が 2 個より少ない場合, エラー値を返す.

先頭アドレスの範囲が 0~2046 でない場合, エラー値を返す.

Float データの範囲が-3.402823466e+38~3.402823466e+38 でない場合, エラー値を返す.

Float データの個数が 1~1024 でない場合, エラー値を返す.

先頭アドレス + (データの個数 - 1) * 2 が 2046 を超える場合, エラー値を返す.

- 使用例** Execute(“FDW”, Array(0, 10.5)) アドレス 0 から 2 ワードに 10.5 を書き込む
 Execute(“FDW”, Array(12, 0.1, 1234.5)) アドレス 12 から 2 ワードに 0.1, アドレス 14 から 2 ワードに 1234.5 を書き込む

3.1.16. CaoController::Execute(“CDR”) コマンド

データエリア領域から文字列を表示桁数分読み出します。

書式 “CDR”, (<WordAddress>, <GtMemNum>)

<WordAddress> : [in] ワードエリア先頭アドレス:0~2047
 <GtMemNum> : [in] GT の表示データ桁数(バイト):1~20
 (GTWIN で設定した表示データバイト数を設定)
 vntRet : [out]VT_BSTR (末尾のスペースを削除した文字列)

引数が 2 個より少ない場合, エラー値を返す.

先頭アドレスの範囲が 0~2047 でない場合, エラー値を返す.

GT の表示データ桁数(バイト)の範囲が 1~20 でない場合, エラー値を返す.

先頭アドレス + (GT の表示データ桁数(バイト) + 1) / 2 - 1 が 2047 を超える場合, エラー値を返す.

- 使用例** GTWIN で表示桁数を 10 に設定した場合,
 Execute(“CDR”, Array(0, 10)) アドレス 0 から 10 バイト分取得
 Execute(“CDR”, Array(12, 10)) アドレス 12 から 10 バイト分取得

3.1.17. CaoController::Execute(“CDW”) コマンド

データエリア領域から文字列を表示桁数分書き込みます。

書式 “CDW”, (<WordAddress>, <GtMemNum>, <bstrData>)

<WordAddress> : [in] ワードエリア先頭アドレス:0~2047
 <GtMemNum> : [in] GT の表示データ桁数(バイト):1~20
 (GTWIN で設定した表示データバイト数を設定)
 <bstrData> : [in] 文字列:MAX で 20 バイト(半角文字:20, 全角文字:10)
 vntRet : [out]VT_I2 (正常:true, 異常:値なし)

エラー引数が 3 個より少ない場合, エラー値を返す.

先頭アドレスの範囲が 0~2047 でない場合, エラー値を返す.

GT の表示データ桁数の範囲が 1~20 でない場合, エラー値を返す.

文字列数が GT の表示データ桁数(バイト)より長い場合, エラー値を返す.

先頭アドレス + (GT の表示データ桁数(バイト) + 1) / 2 - 1 が 2047 を超える場合, エラー値を返す.

使用例 GTWIN で表示桁数を 10 に設定した場合,
 Execute(“CDW”, Array(0, 10, あ)) アドレス 0 からの表示領域分に ‘あ’ を書き込む
 Execute(“CDW”, Array(12, 10, ABCD)) アドレス 12 から表示領域分に ABCD を書き込む

3.1.18. CaoController::Execute(“LDR”) コマンド

データエリア領域から 2 ワード分 LONG として読み出します.

書式 “LDR”, (<WordAddress>, <LongNum>)

<WordAddress> : [in] ワードエリア先頭アドレス:0~2046
 <LongNum> : [in] データ数(LONG の数):1~1024
 vntRet : [out]VT_I4|VT_ARRAY

引数が 2 個より少ない場合, エラー値を返す.

先頭アドレスの範囲が 0~2046 でない場合, エラー値を返す.

データ数の範囲が 1~1024 でない場合, エラー値を返す.

先頭アドレス + (データ数 - 1) * 2 が 2046 を超える場合, エラー値を返す.

使用例 Execute(“LDR”, Array(0, 1)) アドレス 0 から 1 データ分(2 ワード)取得
 Execute(“LDR”, Array(12, 10)) アドレス 12 から 10 データ分(20 ワード)取得

3.1.19. CaoController::Execute(“LDW”) コマンド

データエリア領域に LONG 単位分(2 ワード毎)書き込みます.

書式 “LDW”, (<WordAddress>, <LongData1>, <LongData2>, ... , <LongDataN>)

<WordAddress> : [in] ワードエリア先頭アドレス:0~2046
 <LongDataN> : [in] Long データ:-2147483648~2147483647
 N の最大:1024
 vntRet : [out]VT_I2(正常:true, 異常:値なし)

引数が 2 個より少ない場合, エラー値を返す.

先頭アドレスの範囲が 0~2046 でない場合, エラー値を返す.

Long データの範囲が-2147483648~2147483647 でない場合, エラー値を返す.

Long データの個数が 1~1024 でない場合, エラー値を返す.

先頭アドレス + (データの個数 - 1) * 2 が 2046 を超える場合, エラー値を返す.

使用例 Execute("LDW", Array(0, 105)) アドレス 0 から 2 ワードに 105 を書き込む
Execute("LDW", Array(12, 1, 12345)) アドレス 12 から 2 ワードに 1, アドレス 14 から 2 ワードに 12345 を書き込む