

ASTMotionLib 1.0

July 02, 2004

Copyright © 1995-2004 Astel electronics and industrial automation

Classe **CMotionLib**

Properties

' Imposta / Ritorna la modalità di funzionamento della libreria.
' * **Manual** - Funzioni di Comandi abilitate, Funzioni Ex disabilitate.
' * **Automatic** - Funzioni Ex abilitate, Funzioni di Comandi disabilitate.
object.LibMode = *value*

il parametro *value* è una costante tipo **LibModeEnum** che può assumere la seguente lista di valori:

0 = **eLibManual**
1 = **eLibAutomatic**

' Imposta / Ritorna i settaggi della porta seriale (Baud Rates, Parity, Data Bit, Stop Bit).
object.COMSettings = *value*

il parametro *value* è un tipo **String** formata dai seguenti parametri separati da virgola:

- **Baud Rates**
- **Parity**
- **Data Bit**
- **Stop Bit**

Esempio: `MotionLib.COMSettings = "9600,n,8,1"`

' Imposta / Ritorna il numero della porta di comunicazione da usare.
object.COMPort = *value*

il parametro *value* è un **Integer** che rappresenta il numero identificativo della COM da usare.

' Imposta / Ritorna lo stato della porta di comunicazione selezionata (Aperta (TRUE) / Chiusa (FALSE))
object.COMPortOpen = *value*

il parametro *value* è un **Boolean** che specifica se aprire (TRUE) o chiudere (FALSE) la comunicazione.

' Ritorna la classe **CStepper** per la gestione delle schede di tipo Stepper (ISC/INDWonM/INDWonD).
object.Stepper

.vedere la classe **CStepper**.

' Ritorna la classe **CIOController** per la gestione delle schede di tipo CIOController (EIOC/PIOC).
object.IOController

.vedere la classe **CIOController**.

' Aggancia un eventuale controllo **ProgressBar** utilizzato da **RS485_ScanNetEx** per mostrare lo stato della scansione.

Set *object*.**HookProgressBarForRS485** = *control*

il parametro *control* è un controllo di tipo **ProgressBar**.

' Imposta / Ritorna il valore della flag globale per fermare la routine di wait se in esecuzione.

object.**StopWaitEx** = *value*

il parametro *value* è un Boolean che specifica se fermare (TRUE) o lasciare continuare (FALSE) il metodo *WaitEx*.

Methods

' Mostra la finestra di About contenente tutte le informazioni relative alla versione della libreria.

object.About ()

' Ritorna una stringa contenente la versione della libreria.

object.Info_GetVersion ()

ritorna un valore di tipo **String**.

' Ritorna una stringa contenente il nome interno della libreria.

object.Info_GetLibName ()

ritorna un valore di tipo **String**.

' Ritorna una stringa contenente la data dell' ultima modifica apportata alla libreria.

object.Info_GetUpdateDate ()

ritorna un valore di tipo **String**.

' Attende per un numero di millisecondi specificato.

object.WaitEx (*sngMilliseconds*)

- **sngMilliseconds** indica il numero di millisecondi che si desidera far attendere.

1000 = 1 secondo

Questo parametro è di tipo *Single*.

' [**^C**]- Reset tutti i controlli e assegna un indirizzo identificativo ad ogni scheda. Mode RS232.

object.RS232_ResetBoards (*ByRef strLastDevAddress*, [*strFirstDevAddress* = "A"])

- **strLastDevAddress** viene riempito dalla funzione con l'indirizzo dell'ultimo dispositivo trovato. Questo parametro è di tipo *String*.

- **strFirstDevAddress** indica l'indirizzo che deve essere assegnato alla prima scheda della rete.

Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [**^C**]- Reset tutti i controlli. Mode RS485.

object.RS485_ResetBoards ()

' Esegue la scansione delle schede presenti in rete Mode RS485.

object.RS485_ScanNetEx (*ByRef dicNetList*, [*ByRef sngTimeScan*])

- **dicNetList** viene riempito con la mappa delle schede presenti in rete.

Nel metodo **Keys**(*indice*) viene memorizzato il nome del dispositivo interrogato.

Nel metodo **Items**(*indice*) viene memorizzata la stringa "PRESENT" se il dispositivo è presente altrimenti conterra la stringa vuota "".

Questo parametro è di tipo *Dictionary*.

NOTA: per passare questo parametro includere nel progetto la libreria

"Microsoft Scripting Runtime"

relativa al file "C:\Windows\System\SCRUN.dll".

Selezionare dal menu "Project" la voce "References..."

- **sngTimeScan** viene riempito con il tempo impiegato dalla funzione per la scansione della rete.

Questo parametro è di tipo *Single*, non obbligatorio.

' Ferma la scansione avviata con *RS485_ScanNetEx*.

object.RS485_StopScanNetEx ()

se chiamato durante l'esecuzione del metodo *RS485_ScanNetEx*, la scansione delle schede nella rete viene interrotta.

' Invia il comando specificato.

object.SendCMDEX (*CMDDevAdd*, *CMDId*, *CMDParams*, *ByRef* *CMDAnswer*,
[*boolSpaceBeforeCMDId* = True], [*boolWaitAnswer* = True], [*sngTimeoutAnswer* = 3])

- ***CMDDevAdd*** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*.
- ***CMDId*** specifica il nome del comando da inviare.
Questo parametro è di tipo *String*.
- ***CMDParams*** specifica i parametri relativi al comando.
Questo parametro è di tipo *String*.
- ***CMDAnswer*** viene riempito dall'eventuale risposta ricevuta dalla scheda interrogata.
Questo parametro è di tipo *String*.
- ***boolSpaceBeforeCMDId*** flag che se impostato a True inserisce uno spazio prima del nome del comando all'interno della stringa completa da inviare alla scheda.
Questo parametro è di tipo *Boolean*, non obbligatorio e se non specificato assume come default True.
- ***boolWaitAnswer*** flag che se impostato a True indica alla routine di invio comando che deve aspettare una risposta dalla scheda
L'attesa dura al massimo il tempo specificato dal parametro *sngTimeoutAnswer*.
Questo parametro è di tipo *Boolean*, non obbligatorio e se non specificato assume come default True.
- ***sngTimeoutAnswer*** indica alla routine di invio comando il tempo che deve utilizzare per ricevere la risposta oltre il quale viene segnalato un errore di timeout. Il tempo viene espresso in secondi e viene tenuto in considerazione solo se il parametro *boolWaitAnswer* è stato impostato a True.
Questo parametro è di tipo *Single*, non obbligatorio e se non specificato assume come default 3 sec.

Events

' Evento per il debug dell' invio di un comando e della ricezione.

DebugComm (ByRef *strRequest* As String, ByRef *strAnswer* As String)

- ***strRequest*** ritorna la stringa relativa alla richiesta che è stata inviata. Questo parametro è di tipo *String*.
- ***strAnswer*** ritorna la stringa relativa alla risposta che è stata ricevuta. Questo parametro è di tipo *String*.

Classe **CStepper**

Methods

' [ESC] Interrompi il movimento
object.Kill

' [@] - Stop dolce
object.AxisStop([*strDevAddress* = "A"])

- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [A] - Lettura Scrittura porte di I/O

' Ritorna la stringa in formato binario del valore decimale passato alla funzione.
object.GetBinaryString(*intDecimalValue*) As *String*

- ***intDecimalValue*** specifica il valore decimale che si desidera convertire in formato binario. Questo parametro è di tipo *Integer*.

Esempio:

```
Dim StrBin As String  
StrBin = Stepper.GetBinaryString(11)
```

La variabile *StrBin* conterrà il valore "1011".

' Ritorna nel parametro <*intPortStatus*> lo stato delle 5 porte (P5,P4,P3,P2,P1) per ISC oppure lo stato dell'unica porta P1 per INDW (M/D).

object.GetPort(*intPortStatus*, [*strDevAddress*= "A"], [*intBoardType* = *eBoard_ISC*])

- ***intPortStatus*** viene riempito con lo stato delle porte presenti sulla scheda specificata. Questo parametro è di tipo *Integer*.
- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- ***intBoardType*** specifica il tipo di scheda a cui si sta inviando il comando. Questo parametro è di tipo *BoardTypeEnum*, non obbligatorio e se non specificato assume come default *eBoard_ISC*. Può assumere la seguente lista di valori:

0 = ***eBoard_ISC***
1 = ***eBoard_INDW_on_M***
2 = ***eBoard_INDW_on_D***

' Imposta lo stato delle 5 porte (P5,P4,P3,P2,P1), sulla base del parametro <intMaskPort> e <intPortStatus> e della costante <MASK_OUTPUT_FOR_*> data dal parametro <intBoardType>.

object.SetPort([*strDevAddress* = "A"], [*intBoardType* = eBoard_ISC], [*intMaskPort* = 0], [*intPortToSet* = 0])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intBoardType** specifica il tipo di scheda a cui si sta inviando il comando.
Questo parametro è di tipo *BoardTypeEnum*, non obbligatorio e se non specificato assume come default *eBoard_ISC*. Può assumere la seguente lista di valori:
 - 0 = **eBoard_ISC**
 - 1 = **eBoard_INDW_on_M**
 - 2 = **eBoard_INDW_on_D**
- **intMaskPort** specifica la maschera software di input/output per ogni porta.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.
- **intPortToSet** specifica lo stato che ciascuna porta deve assumere.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.

' [Funzione di utilità]

' Ritorna (FALSE/TRUE) lo stato, (0/1), del bit specificato dal parametro <intBit>.

object.GetBit(*intPortStatus*, *intBit*) As Boolean

- **intPortStatus** specifica lo stato di ciascuna porta deve assumere.
Questo parametro è di tipo *Integer*.
- **intBit** specifica la posizione del bit che si vuole leggere nel parametro *intPortStatus*.
Questo parametro è di tipo *Integer*.

' [Funzione di utilità]

' Imposta il bit specificato dal parametro <intBit> utilizzando l' eventuale MaskPort.

' Solo se la MaskPort indica che il bit è un Output lo cambia.

object.SetBit(ByRef *intBitStatus*, *intBit*, [*intMaskPort%* = 0])

- **intBitStatus** ritorna .
Questo parametro è di tipo *Integer*.
- **intBit** specifica la posizione del bit che si vuole leggere nel parametro *intPortStatus*.
Questo parametro è di tipo *Integer*.

object.MovementStates(ByRef *boolMoving*, ByRef *boolTripPointExceed*, ByRef *boolRotationCW*, [*strDevAddress* = "A"])

- **boolMoving** viene modificato indicando lo stato del bit di motore in movimento.
Questo parametro è di tipo *Boolean*.
- **boolTripPointExceed** viene modificato indicando lo stato del bit di punto di tappa passato.
Questo parametro è di tipo *Boolean*.
- **boolRotationCW** viene modificato indicando lo stato del bit di rotazione in senso antiorario CW.
Questo parametro è di tipo *Boolean*.
- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

object.**InvertTrip**([*strDevAddress* = "A"[, [*intBoardType* = *eBoard_ISC*])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intBoardType** specifica il tipo di scheda a cui si sta inviando il comando. Questo parametro è di tipo *BoardTypeEnum*, non obbligatorio e se non specificato assume come default *eBoard_ISC*. Può assumere la seguente lista di valori:

0 = **eBoard_ISC**
1 = **eBoard_INDW_on_M**
2 = **eBoard_INDW_on_D**

' [B] - Impostazioni velocità manuali

object.**SetJogSpeed**([*strDevAddress* = "A"], [*intBoardType* = *eBoard_ISC*], [*vSpeed* = 0])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intBoardType** specifica il tipo di scheda a cui si sta inviando il comando. Questo parametro è di tipo *BoardTypeEnum*, non obbligatorio e se non specificato assume come default *eBoard_ISC*. Può assumere la seguente lista di valori:

0 = **eBoard_ISC**
1 = **eBoard_INDW_on_M**
2 = **eBoard_INDW_on_D**

- **vSpeed** specifica la velocità che deve essere impostata per il Jog. Questo parametro è di tipo *Variant*, non obbligatorio e se non specificato assume come default 0.

' [D] - Imposta risoluzione velocità

object.**SetSpeedDivisor**([*strDevAddress* = "A"], [*intBoardType* = *eBoard_ISC*], [*intFactor* = 1])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intBoardType** specifica il tipo di scheda a cui si sta inviando il comando. Questo parametro è di tipo *BoardTypeEnum*, non obbligatorio e se non specificato assume come default *eBoard_ISC*. Può assumere la seguente lista di valori:

0 = **eBoard_ISC**
1 = **eBoard_INDW_on_M**
2 = **eBoard_INDW_on_D**

- **intFactor** specifica il valore del fattore di divisione che deve essere impostato. Questo parametro è di tipo *Variant*, non obbligatorio e se non specificato assume come default 0.

' [E] - Controllo delle correnti

object.**SetHWPParameters**([*strDevAddress* = "A"], [*intBoardType* = eBoard_ISC], [*intAutoPowerDown* As AutoPowerDownEnum = eAutoPowerDown_Active], [*intEnablePolarity* As EnablePolarityEnum = eEnablePolarity_ActiveLow], [*intLimitSwitchPolarity* As LimitSwitchPolarityEnum = eLimitSwitchPolarity_ActiveLow])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intBoardType** specifica il tipo di scheda a cui si sta inviando il comando. Questo parametro è di tipo *BoardTypeEnum*, non obbligatorio e se non specificato assume come default *eBoard_ISC*. Può assumere la seguente lista di valori:
 - 0 = **eBoard_ISC**
 - 1 = **eBoard_INDW_on_M**
 - 2 = **eBoard_INDW_on_D**
- **intAutoPowerDown** specifica lo stato del bit di *AutoPowerDown*. Questo parametro è di tipo *AutoPowerDownEnum*, non obbligatorio e se non specificato assume come default *eAutoPowerDown_Active*. Può assumere la seguente lista di valori:
 - 0 = **eAutoPowerDown_Active**
 - 1 = **eAutoPowerDown_NotActive**
- **intEnablePolarity** specifica lo stato del bit di *EnablePolarity*. Questo parametro è di tipo *EnablePolarityEnum*, non obbligatorio e se non specificato assume come default *eEnablePolarity_ActiveLow*. Può assumere la seguente lista di valori:
 - 0 = **eEnablePolarity_ActiveLow**
 - 1 = **eEnablePolarity_ActiveHigh**
- **intLimitSwitchPolarity** specifica lo stato del bit di *LimitSwitchPolarity*. Questo parametro è di tipo *LimitSwitchPolarityEnum*, non obbligatorio e se non specificato assume come default *eLimitSwitchPolarity_ActiveLow*. Può assumere la seguente lista di valori:
 - 0 = **eLimitSwitchPolarity_ActiveLow**
 - 1 = **eLimitSwitchPolarity_ActiveHigh**

' [F] - Ricerca posizione di zero

object.**FindHome**([*strDevAddress* = "A"], [*intBoardType* = eBoard_ISC], [*lngSpeed* = 18], [*intDirection* = 0])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intBoardType** specifica il tipo di scheda a cui si sta inviando il comando. Questo parametro è di tipo *BoardTypeEnum*, non obbligatorio e se non specificato assume come default *eBoard_ISC*. Può assumere la seguente lista di valori:
 - 0 = **eBoard_ISC**
 - 1 = **eBoard_INDW_on_M**
 - 2 = **eBoard_INDW_on_D**
- **lngSpeed** specifica il valore della velocità che deve essere utilizzata per la ricerca dell'Home. Questo parametro è di tipo *Long*, non obbligatorio e se non specificato assume come default 18. Se il parametro *intBoardType* viene impostato a *eBoard_ISC*, questo parametro può assumere tutti i valori compresi tra 18 e 23.000. Se il parametro *intBoardType* viene impostato a *eBoard_INDW_on_(x)*, questo parametro può assumere tutti i valori compresi tra 37 e 51.000.
- **intDirection** specifica la direzione del movimento: 0 = CCW = dir oraria; 1 = CW = dir antioraria. Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.

' Funzione generica per calcolare il valore decimale del parametro Resolution del comando "SetStepResolution"

object.**GetDecValueByStepResolution**(intResolution As StepSizeEnum) As Integer

- **intResolution** specifica la il valore della risoluzione da calcolare in decimale. Questo parametro e di tipo *StepSizeEnum*. Può assumere la seguente lista di valori:

0 = **eFull_Step**
 1 = **eHalf_Step**
 2 = **eQuarter_Step**
 3 = **e8_Step**
 4 = **e16_Step**
 5 = **e32_Step**
 6 = **e64_Step**
 7 = **e128_Step**
 8 = **e256_Step**
 9 = **e5_Step**
 10 = **e10_Step**
 11 = **e25_Step**
 12 = **e50_Step**
 13 = **e125_Step**
 14 = **e250_Step**

' [H] - Risoluzione (passo intero / mezzo passo / quarto di passo)

object.**SetStepResolution**([strDevAddress = "A"], [intBoardType = eBoard_ISC], [intResolution = eFull_Step])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro e di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intBoardType** specifica il tipo di scheda a cui si sta inviando il comando. Questo parametro e di tipo *BoardTypeEnum*, non obbligatorio e se non specificato assume come default *eBoard_ISC*. Può assumere la seguente lista di valori:

0 = **eBoard_ISC**
 1 = **eBoard_INDW_on_M**
 2 = **eBoard_INDW_on_D**

- **intResolution** specifica la il valore della risoluzione da impostare. Questo parametro e di tipo *StepSizeEnum*. Può assumere la seguente lista di valori:

con *intBoardType* è impostato a **eBoard_ISC**:

0 = **eFull_Step**
 1 = **eHalf_Step**
 2 = **eQuarter_Step**
 3 = **e8_Step**

con *intBoardType* è impostato a **eBoard_INDW_on_M**:

1 = **eHalf_Step**
 2 = **eQuarter_Step**
 3 = **e8_Step**
 4 = **e16_Step**
 5 = **e32_Step**
 6 = **e64_Step**
 7 = **e128_Step**
 8 = **e256_Step**
 9 = **e5_Step**
 10 = **e10_Step**
 11 = **e25_Step**

12 = **e50_Step**
 13 = **e125_Step**
 14 = **e250_Step**

con *intBoardType* è impostato a **eBoard_INDW_on_D**:

0 = **eFull_Step**
 1 = **eHalf_Step**
 2 = **eQuarter_Step**

' [I] - Imposta la velocità iniziale

object.**SetStartSpeed**([*strDevAddress* = "A"], [*intBoardType* = eBoard_ISC], [*IngSpeed* = 18])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intBoardType** specifica il tipo di scheda a cui si sta inviando il comando. Questo parametro è di tipo *BoardTypeEnum*, non obbligatorio e se non specificato assume come default *eBoard_ISC*. Può assumere la seguente lista di valori:

0 = **eBoard_ISC**
 1 = **eBoard_INDW_on_M**
 2 = **eBoard_INDW_on_D**

- **IngSpeed** specifica il valore della velocità iniziale che deve essere impostata. Questo parametro è di tipo *Long*, non obbligatorio e se non specificato assume come default 18. Se il parametro *intBoardType* viene impostato a *eBoard_ISC*, questo parametro può assumere tutti i valori compresi tra 18 e 23.000. Se il parametro *intBoardType* viene impostato a *eBoard_INDW_on_(x)*, questo parametro può assumere tutti i valori compresi tra 37 e 51.000.

' [K] - Pendenza delle rampe

object.**SetRampSlope**([*strDevAddress* = "A"], [*intSlope* = 0])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intSlope** specifica il valore delle rampe di accelerazione/decelerazione che devono essere impostate. Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.

' [M] - Movimento a velocità costante

object.**MoveAtConstantSpeed**([*strDevAddress* = "A"], [*intBoardType* = eBoard_ISC], [*IngSpeed* = 18])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
 - **intBoardType** specifica il tipo di scheda a cui si sta inviando il comando. Questo parametro è di tipo *BoardTypeEnum*, non obbligatorio e se non specificato assume come default *eBoard_ISC*. Può assumere la seguente lista di valori:
- 0 = **eBoard_ISC**
 1 = **eBoard_INDW_on_M**
 2 = **eBoard_INDW_on_D**
- **IngSpeed** specifica il valore della velocità costante a cui il motore deve essere muoversi. Questo parametro è di tipo *Long*, non obbligatorio e se non specificato assume come default 18. Se il parametro *intBoardType* viene impostato a *eBoard_ISC*, questo parametro può assumere tutti i valori compresi tra 18 e 23.000. Se il parametro *intBoardType* viene impostato a *eBoard_INDW_on_(x)*, questo parametro può assumere tutti i valori compresi tra 37 e 51.000.

' [O] - Imposta origine

object.SetOrigin([*strDevAddress* = "A"], [*vPosition* = 0])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **vPosition** specifica il valore della di origine (lo 0 relativo) che deve essere impostato.
Questo parametro è di tipo *Variant*, non obbligatorio e se non specificato assume come default 0.

' [R] - Movimento relativo all' origine

object.GoToPosition([*strDevAddress* = "A"], [*vPosition* = 0])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **vPosition** specifica la posizione che deve essere raggiunta.
Questo parametro è di tipo *Variant*, non obbligatorio e se non specificato assume come default 0.

' *****

' **WARNING: Non usare frequentemente questo comando per evitare l' usura di durata della memoria.**

' *****

' [S] - Memorizzazione parametri

object.StoreParameters([*strDevAddress* = "A"])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [T] - Punto di tappa

object.SetTripPoint([*strDevAddress* = "A"], [*vPosition* = 0], [*bNegativeSignForTheZero* = False])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **vPosition** specifica il valore della posizione in cui deve essere indicato il punto di tappa.
Questo parametro è di tipo *Variant*, non obbligatorio e se non specificato assume come default 0.
- **bNegativeSignForTheZero** specifica se aggiungere o no il segno allo zero.
Questo parametro è di tipo *Boolean*, non obbligatorio e se non specificato assume come default False.

' [V] - Imposta la velocità di movimento (quella finale che viene raggiunta dopo l' esecuzione della rampa di accelerazione)

object.SetSlewSpeed([*strDevAddress* = "A"], [*intBoardType* = *eBoard_ISC*], [*lngSpeed* = 18])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intBoardType** specifica il tipo di scheda a cui si sta inviando il comando.
Questo parametro è di tipo *BoardTypeEnum*, non obbligatorio e se non specificato assume come default *eBoard_ISC*. Può assumere la seguente lista di valori:

0 = **eBoard_ISC**

1 = **eBoard_INDW_on_M**

2 = **eBoard_INDW_on_D**

- **lngSpeed** specifica il valore della velocità finale che deve essere impostata.
Questo parametro è di tipo *Long*, non obbligatorio e se non specificato assume come default 18.
Se il parametro *intBoardType* viene impostato a *eBoard_ISC*, questo parametro può assumere tutti i valori compresi tra 18 e 23.000.
Se il parametro *intBoardType* viene impostato a *eBoard_INDW_on_(x)*, questo parametro può assumere tutti i valori compresi tra 37 e 51.000.

' [W] - **Attesa, il controller rimane in stato di inattività per il tempo in millisecondi**

object.Wait([*strDevAddress* = "A"], [*IngTimeMillisec* = 0])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **IngTimeMillisec** specifica tempo, espresso in millisecondi, per il quale mettere il sistema in attesa.
Questo parametro è di tipo *Long*, non obbligatorio e se non specificato assume come default 0.

' [X] - **Esamina i parametri**

object.QueryParameters(ByRef *strQueryResults*, [*strDevAddress* = "A"])

- **strQueryResults** viene riempito con i parametri impostati nella scheda:
kk = rampa, **ii** = velocità iniziale, **vv** = velocità finale, [**tt** = punto di tappa (solo se impostato)].
Questo parametro è di tipo *String*.
- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [Z] - **Lettura della posizione corrente**

object.ReadPosition(ByRef *vCurrentPosition*, [*strDevAddress* = "A"])

- **vCurrentPosition** viene riempito con la posizione corrente letta.
Questo parametro è di tipo *Variant*.
- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' **[[[]]]** - Lettura finecorsa per schede ISC

```
object.ReadLimitSwitch_for_ISC([ByRef boolHalfStepOutLevel], [ByRef boolEnableOutLevel],  
                                [ByRef boolQuarterStepOutLevel], [ByRef boolHomeSwitchOutLevel],  
                                [ByRef boolLimitAOutLevel], [ByRef boolLimitBOutLevel],  
                                [strDevAddress = "A"])
```

- **boolHalfStepOutLevel** viene riempito con lo stato del bit di output *HalfStep*. Questo parametro è di tipo *Boolean*.
- **boolEnableOutLevel** viene riempito con lo stato del bit di output *Enable*. Questo parametro è di tipo *Boolean*.
- **boolQuarterStepOutLevel** viene riempito con lo stato del bit di output *QuarterStep*. Questo parametro è di tipo *Boolean*.
- **boolHomeSwitchOutLevel** viene riempito con lo stato del bit di output *HomeSwitch*. Questo parametro è di tipo *Boolean*.
- **boolLimitAOutLevel** viene riempito con lo stato del bit di output *LimitA*. Questo parametro è di tipo *Boolean*.
- **boolLimitBOutLevel** viene riempito con lo stato del bit di output *LimitB*. Questo parametro è di tipo *Boolean*.
- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' **[A]** - Lettura della risoluzione per schede INDW su [M/D]

```
object.ReadResolution_for_INDW([ByRef boolRes0], [ByRef boolRes1], [ByRef boolRes2],  
                                [ByRef boolRes3], [strDevAddress = "A"])
```

- **boolRes0** viene riempito con lo stato del bit 0 di risoluzione. Questo parametro è di tipo *Boolean*.
- **boolRes1** viene riempito con lo stato del bit 1 di risoluzione. Questo parametro è di tipo *Boolean*.
- **boolRes2** viene riempito con lo stato del bit 2 di risoluzione. Questo parametro è di tipo *Boolean*.
- **boolRes3** viene riempito con lo stato del bit 3 di risoluzione. Questo parametro è di tipo *Boolean*.
- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' **[]** - Lettura finecorsa per schede INDW su **[M/D]**

object.**ReadLimitSwitch_for_INDW**([ByRef *boolEnableOutLevel*], [ByRef *boolHomeSwitchOutLevel*],
[ByRef *boolLimitAOutLevel*], [ByRef *boolLimitBOutLevel*],
[*strDevAddress* = "A"])

- ***boolEnableOutLevel*** viene riempito con lo stato del bit di output *Enable*. Questo parametro è di tipo *Boolean*.
- ***boolHomeSwitchOutLevel*** viene riempito con lo stato del bit di output *HomeSwitch*. Questo parametro è di tipo *Boolean*.
- ***boolLimitAOutLevel*** viene riempito con lo stato del bit di output *LimitA*. Questo parametro è di tipo *Boolean*.
- ***boolLimitBOutLevel*** viene riempito con lo stato del bit di output *LimitB*. Questo parametro è di tipo *Boolean*.
- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' **[+]** - Movimento in senso orario - **[CW = Clock Wise]**

object.**IndexPlus**([*strDevAddress* = "A"], [*vDistance* = 0])

- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- ***vDistance*** specifica quanta distanza deve percorrere in senso orario. Questo parametro è di tipo *Variant*, non obbligatorio e se non specificato assume come default 0.

' **[-]** - Movimento in senso antiorario - **[CCW = Counter Clock Wise]**

object.**IndexMinus**([*strDevAddress* = "A"], [*vDistance* = 0])

- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- ***vDistance*** specifica quanta distanza deve percorrere in senso anti-orario. Questo parametro è di tipo *Variant*, non obbligatorio e se non specificato assume come default 0.

' **[^]** - Lettura stato del movimento

object.**ReadStatus**([ByRef *boolPositioning*], [ByRef *boolConstantSpeed*], [ByRef *boolFindingHome*],
[ByRef *boolRamping*], [*strDevAddress* = "A"])

- ***boolPositioning*** viene riempito con lo stato del bit di posizionamento. Questo parametro è di tipo *Boolean*.
- ***boolConstantSpeed*** viene riempito con lo stato del bit di velocità costante. Questo parametro è di tipo *Boolean*.
- ***boolFindingHome*** viene riempito con lo stato del bit di ricerca home. Questo parametro è di tipo *Boolean*.
- ***boolRamping*** viene riempito con lo stato del bit di generazione rampa. Questo parametro è di tipo *Boolean*.
- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

Classe **CIOController**

Methods

' [M] - Richiede Nome e Versione del dispositivo

' Valido per: EIOC; PIOC

object.**GetDevInfo**(ByRef *strDevInfo*, [*strDevAddress* = "A"])

- **strDevInfo** viene riempito con il nome e la versione della scheda.
Questo parametro è di tipo *String*.
- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [I] - Richiede lo stato degli ingressi

' Valido per: EIOC (8 ingressi); PIOC (4 ingressi)

object.**GetInput**(ByRef *intInputState*, [*strDevAddress* = "A"])

- **intInputState** viene riempito con lo stato degli input.
Questo parametro è di tipo *Integer*.
- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [C] - Richiede i cambiamenti di stato (High e Low) degli ingressi

' Valido per: EIOC (8 ingressi); PIOC (4 ingressi)

object.**GetInputChange**(ByRef *intRiseEdge*, ByRef *intFallEdge*, [*strDevAddress* = "A"])

- **intRiseEdge** viene riempito con i cambiamenti di stato *alti* degli input.
Questo parametro è di tipo *Integer*.
- **intFallEdge** viene riempito con i cambiamenti di stato *bassi* degli input.
Questo parametro è di tipo *Integer*.
- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [A] - Richiede il valore presente sul canale analogico specificato

' Valido per: EIOC (2 ingressi); PIOC (2 ingressi)

object.**GetAnalogInput**(ByRef *intVoltageRelative*, [*strDevAddress* = "A"], [*intAINChannel* = 0])

- **intVoltageRelative** viene riempito con il valore dell'input analogico specificato dal parametro *intAINChannel*.
Questo parametro è di tipo *Integer*.
- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intAINChannel** specifica il canale analogico cui si vuole leggere il canale.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.

' [O] - Imposta lo stato delle uscite in blocco secondo la maschera indicata

' Valido per: EIOC (8 output); PIOC (4 output)

object.**SetOutput**([*strDevAddress* = "A"], [*intOutStateMask* = 0])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intOutStateMask** specifica quali stati devono assumere le uscite.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.

' [R] - Richiede lo stato delle uscite

' Valido per: EIOC (8 output); PIOC (4 output)

object.**GetOutput**(ByRef *intOutState*, [*strDevAddress* As *String* = "A"])

- **intOutState** viene riempito con lo stato degli output.
Questo parametro è di tipo *Integer*.
- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [D] - Imposta lo stato di una singola uscita

' Valido per: EIOC (8 output); PIOC (4 output)

object.**SetSingleOutput**([*strDevAddress* = "A"], [*intOUTChannel* = 0], [*intState* = 0])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intOUTChannel** specifica quale uscita deve essere impostata.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.
- **intState** specifica quale stato deve assumere l'uscita.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.

' [D] - Commuta lo stato di un uscita dopo un tempo T1, partendo da uno stato iniziale specifico

' Valido per: EIOC (8 output); PIOC (4 output)

object.**PulseGen**([*strDevAddress* = "A"], [*intOUTChannel* = 0], [*intInitState* = 0], [*IngMillisecondsT1* = 100])

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intOUTChannel** specifica quale uscita deve essere impostata.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.
- **intInitState** specifica in quale stato, l'uscita, deve iniziare l'impulso.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.
- **IngMillisecondsT1** specifica dopo quanto tempo, espresso in millisecondi, l'uscita deve commutare di stato.
Questo parametro è di tipo *Long*, non obbligatorio e se non specificato assume come default 100 millisecondi.

' [D] - Genera un treno di impulsi sull' uscita specificata, partendo da uno stato iniziale
' I tempi di salita e discesa sono impostabili tramite i tempi T1 e T2
' Valido per: EIOC (8 output); PIOC (4 output)

```
object.SquareWaveGen([strDevAddress = "A"], [intOUTChannel = 0], [intInitState = 0],  
                    [IngMillisecondsT1 = 100], [IngMillisecondsT2 = 100])
```

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intOUTChannel** specifica quale uscita deve essere impostata.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.
- **intInitState** specifica in quale stato, l'uscita, deve iniziare l'impulso.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.
- **IngMillisecondsT1** specifica dopo quanto tempo, espresso in millisecondi, l'uscita deve commutare di stato da quello iniziale al successivo.
Questo parametro è di tipo *Long*, non obbligatorio e se non specificato assume come default 100 millisecondi.
- **IngMillisecondsT2** specifica dopo quanto tempo, espresso in millisecondi, l'uscita deve nuovamente commutare allo stato iniziale.
Questo parametro è di tipo *Long*, non obbligatorio e se non specificato assume come default 100 millisecondi.

' [D] - Genera un' onda quadra sull' uscita specificata, partendo da uno stato iniziale
' I tempi di salita e discesa sono impostabili tramite i tempi T1 e T2
' Il numero di impulsi è impostabile

' Valido per: EIOC (8 output); PIOC (4 output)

```
object.SquareWavePulseTrainGen([strDevAddress = "A"], [intOUTChannel = 0], [intInitState = 0],  
                               [IngMillisecondsT1 = 100], [IngMillisecondsT2 = 100], [intCounter = 3])
```

- **strDevAddress** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- **intOUTChannel** specifica quale uscita deve essere impostata.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.
- **intInitState** specifica in quale stato, l'uscita, deve iniziare l'impulso.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.
- **IngMillisecondsT1** specifica dopo quanto tempo, espresso in millisecondi, l'uscita deve commutare di stato da quello iniziale al successivo.
Questo parametro è di tipo *Long*, non obbligatorio e se non specificato assume come default 100 millisecondi.
- **IngMillisecondsT2** specifica dopo quanto tempo, espresso in millisecondi, l'uscita deve nuovamente commutare allo stato iniziale.
Questo parametro è di tipo *Long*, non obbligatorio e se non specificato assume come default 100 millisecondi.
- **intCounter** specifica quanti impulsi devono essere generati.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 3.

' [K] - Richiede il primo carattere premuto sul tastierino

' Valido per: EIOC; PIOC

object.**GetFirstKey**(ByRef *strKeyPressed*, [*strDevAddress* = "A"])

- ***strKeyPressed*** viene riempito con un carattere, il primo premuto, presente nel buffer di tastiera. Questo parametro è di tipo *String*.
- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [K] - Richiede un numero [n] caratteri contenuti nel buffer di tastierino

' Valido per: EIOC; PIOC

object.**GetKeyboardBuffer**(ByRef *strBufferKeysPressed*, [*strDevAddress* = "A"], [*intNumOfKeys* = 16])

- ***strBufferKeysPressed*** viene riempito con tutti i caratteri premuti presenti nel buffer di tastiera. Questo parametro è di tipo *String*.
- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [L] - Cancella tutto il Display

' Valido per: EIOC (4 linee x 20 caratteri); PIOC (2 linee x 16 caratteri)

object.**ClearDisplay**([*strDevAddress* = "A"])

- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [L] - Cancella una linea del Display

' Valido per: EIOC (4 linee x 20 caratteri); PIOC (2 linee x 16 caratteri)

object.**ClearLineDisplay**([*strDevAddress* = "A"], [*intLineNumber* = 0])

- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- ***intLineNumber*** specifica quale linea del display deve essere cancellata. Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.

' [L] - Scrive una linea del Display

' Valido per: EIOC (4 linee x 20 caratteri); PIOC (2 linee x 16 caratteri)

object.**WriteLineDisplay**([*strDevAddress* = "A"], [*intLineNumber* = 0], [*strLineDisplayText* = ""])

- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- ***intLineNumber*** specifica quale linea del display deve essere scritta. Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.
- ***strLineDisplayText*** specifica la stringa da scrivere sulla linea del display. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "".

' [B] - Accende/Spegne la Backlight del Display

' Valido solo per EIOC (4 linee x 20 caratteri)

object.**Backlight**([*strDevAddress* = "A"], [*intTime* = 0])

- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando. Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- ***intTime*** specifica il tempo, espresso in millisecondi, per cui la retroilluminazione del display deve rimanere accesa. Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 0.

' [B] - Fa suonare il Buzzer (se presente)

' Valido solo per EIOC (4 linee x 20 caratteri)

object.**Buzzer**([*strDevAddress* = "A"], [*intTime* = 1])

- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- ***intTime*** specifica il tempo, espresso in decimi di secondo, per cui il buzzer deve suonare.
Questo parametro è di tipo *Integer*, non obbligatorio e se non specificato assume come default 1.

' [E] - Richiede la posizione dell' encoder

' Valido solo per EIOC

object.**GetEncoderPosition**(ByRef *IngPosition*, [*strDevAddress* = "A"])

- ***IngPosition*** viene riempita con la posizione in cui si trova l'encoder.
Questo parametro è di tipo *Long*.
- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

' [E] - Imposta la posizione di arrivo per l' encoder

' Valido solo per EIOC

object.**SetEncoderPosition**([*strDevAddress* = "A"], [*IngPosition* = 0])

- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".
- ***IngPosition*** specifica la posizione che l'encoder deve raggiungere.
Questo parametro è di tipo *Long*, non obbligatorio e se non specificato assume come default 0

' [V] - Richiede la velocità dell' encoder

' Valido solo per EIOC

object.**GetEncoderSpeed**(ByRef *IngSpeed*, [*strDevAddress* = "A"])

- ***IngSpeed*** viene riempita con la velocità con cui si sta muovendo l'encoder.
Questo parametro è di tipo *Long*.
- ***strDevAddress*** specifica l'indirizzo della scheda a cui inviare il comando.
Questo parametro è di tipo *String*, non obbligatorio e se non specificato assume come default "A".

Costanti utilizzate da alcuni metodi

' Utilizzata per selezionare la modalità di funzionamento della libreria.

Tipo: **LibModeEnum**

Valori:

eLibNormal = 0

eLibExport = 1

' Utilizzata per selezionare il tipo di scheda a cui il comando deve essere inviato.

Tipo: **BoardTypeEnum**

Valori:

eBoard_ISC = 0

eBoard_INDW_on_M = 1

eBoard_INDW_on_D = 2

' Utilizzata per selezionare la risoluzione che deve essere impostata.

Tipo: **StepSizeEnum**

Valori:

eFull_Step = 0

eHalf_Step = 1

eQuarter_Step = 2

e8_Step = 3

e16_Step = 4

e32_Step = 5

e64_Step = 6

e128_Step = 7

e256_Step = 8

e5_Step = 9

e10_Step = 10

e25_Step = 11

e50_Step = 12

e125_Step = 13

e250_Step = 14

' Utilizzata per selezionare attivare / disattivare l'autopowerdown.

Tipo: **AutoPowerDownEnum**

Valori:

eAutoPowerDown_Active = 0

eAutoPowerDown_NotActive = 1

' Utilizzata per impostare la polarità dell'enable, attivo basso o attivo alto.

Tipo: **EnablePolarityEnum**

Valori:

eEnablePolarity_ActiveLow = 0

eEnablePolarity_ActiveHigh = 1

' Utilizzata per impostare la polarità dei limiti, attivi bassi o attivi alti.

Tipo: **LimitSwitchPolarityEnum**

Valori:

eLimitSwitchPolarity_ActiveLow = 0

eLimitSwitchPolarity_ActiveHigh = 1

Codici di errore

Il metodo chiamato ha avuto successo.

ERR_SUCCESS = 0

Il metodo chiamato indica che non può comunicare con la rete perché la porta di comunicazione non è stata aperta.

ERR_NOT_PORT_OPEN = -1

Il metodo chiamato indica che c'è già un'altro metodo che sta ancora utilizzando la porta di comunicazione, quindi non può inviare il comando.

ERR_TIMEOUT_BUSY = -2

Il metodo chiamato indica che la risposta attesa non è stata ricevuta entro il tempo specificato.

ERR_TIMEOUT_WAIT_RECEIVE_ANSWER = -3

Il metodo chiamato indica che la risposta ricevuta al comando inviato non è valida.

ERR_ANSWER_NOT_VALID = -4