



Supervisione e controllo XML-based
da Windows a Windows CE

Guida ai Driver di Comunicazione

Versione 11.6 - Ed. ott.2019

Sommario

1. DRIVER DI COMUNICAZIONE	1
1.1. INTRODUZIONE	ERRORE. IL SEGNALIBRO NON È DEFINITO.
1.2. INTRODUZIONE AI DRIVER	1
1.3. IL DRIVER NEL PROGETTO	1
1.4. LINK ALLE AREE DI MEMORIA	3
1.5. INDIRIZZAMENTO SU VARIABILI DINAMICHE	4
1.6. TASK: INCARICHI STATICI	5
1.7. INSTALLAZIONE DRIVER	6
1.7.1. File di Configurazione dei Driver di Comunicazione.....	7
1.8. DUPLICAZIONE DI DRIVER.....	7
1.9. ESECUZIONE DRIVER	9
1.10. ERRORI HARDWARE RS232	9
1.11. IMPOSTAZIONI GENERALI	10
1.12. GENERAL (DRIVER)	10
1.13. DEBUG	12
1.14. IMPOSTAZIONI STAZIONI	13
1.15. STATIONS	13
1.16. GENERAL (STATION)	14
1.17. SERIAL PORT SETTINGS	15
1.18. QUEUE SIZE.....	16
1.19. TIMEOUTS	16
1.20. BRIDGING SERVICE SETTINGS	16
1.21. TAPI SETTINGS	18
1.22. TCP/IP SETTINGS	20
1.23. RAS SETTINGS	21
1.24. CONFIGURAZIONI SPECIALI (TAPI E RAS).....	22
1.25. HALF DUPLEX	23
1.26. IMPOSTAZIONI TASK	23
1.27. TASKS	23
1.28. PROPRIETÀ GENERALI TASK STATICI	24
1.29. TASK DINAMICI	26
1.30. IMPORT DATABASE DISPOSITIVO	29
1.31. IMPORTAZIONE DATABASE DISPOSITIVO.....	29
1.32. API INTERFACCE BASIC SCRIPT	31
1.33. UTILIZZO DELLE INTERFACCE BASIC SCRIPT.....	31
1.33.1. DriverInterface	32
DriverInterface Events	32
1.34. FIREONDATASENT, DRIVERINTERFACE EVENT	32
1.35. FIREONNEWDATA, DRIVERINTERFACE EVENT	33
1.36. FIREONQUALITYCHANGED, DRIVERINTERFACE EVENT	33
DriverInterface Functions.....	34
1.37. ADDDRIVERTASK, DRIVERINTERFACE FUNCTION	34
1.38. ADDSTATION, DRIVERINTERFACE FUNCTION.....	36
1.39. CLOSEDRIVER, DRIVERINTERFACE FUNCTION.....	37
1.40. GETDRIVERSTATION, DRIVERINTERFACE FUNCTION	38
1.41. GETDRIVERTASK, DRIVERINTERFACE FUNCTION	38
1.42. INITDRIVER, DRIVERINTERFACE FUNCTION	39
1.43. OPENDRIVER, DRIVERINTERFACE FUNCTION	40
1.44. REMOVEDRIVERTASK, DRIVERINTERFACE FUNCTION	40
1.45. REMOVESTATION, DRIVERINTERFACE FUNCTION	41
1.46. SAVECONFIGDRIVER, DRIVERINTERFACE FUNCTION	41
1.47. SETDRIVER, DRIVERINTERFACE FUNCTION.....	42
1.48. STARTSTATION, DRIVERINTERFACE FUNCTION	43
1.49. TERMINATEDRIVER, DRIVERINTERFACE FUNCTION	43
DriverInterface Properties.....	44
1.50. DELAYEVENTS, DRIVERINTERFACE PROPERTY	44
1.50.1. TaskInterface.....	45

TaskInterface Functions	45
1.51. EXECUTE, TASKINTERFACE FUNCTION	45
1.52. GETFLATVARIABLELIST, TASKINTERFACE FUNCTION	45
1.53. GETMAXBYTESIZE, TASKINTERFACE FUNCTION	46
1.54. GETREADBYTEBUFFER, TASKINTERFACE FUNCTION	46
1.55. GETSTATIONOBJECT, TASKINTERFACE FUNCTION	47
1.56. GETTOTALSIZEBYTES, TASKINTERFACE FUNCTION	48
1.57. GETVARIABLELISTSEPARATOR, TASKINTERFACE FUNCTION	49
1.58. GETXMLSETTINGS, TASKINTERFACE FUNCTION	49
1.59. HASBEENEXECUTED, TASKINTERFACE FUNCTION	50
1.60. ISVALID, TASKINTERFACE FUNCTION	51
1.61. SETWRITEBYTEBUFFER, TASKINTERFACE FUNCTION	51
1.62. TOSTRING, TASKINTERFACE FUNCTION	52
TaskInterface Properties	52
1.63. AUTODELETE, TASKINTERFACE PROPERTY	52
1.64. CONDITIONALVARIABLE, TASKINTERFACE PROPERTY	53
1.65. INUSE, TASKINTERFACE PROPERTY	54
1.66. LASTEXECUTIONTIME, TASKINTERFACE PROPERTY	54
1.67. NAME, TASKINTERFACE PROPERTY	55
1.68. POLLINGTIMENoINUSE, TASKINTERFACE PROPERTY	55
1.69. SWAPWORDS, TASKINTERFACE PROPERTY	56
1.70. STATION, TASKINTERFACE PROPERTY	57
1.71. SWAPBYTES, TASKINTERFACE PROPERTY	57
1.72. POLLINGTIME, TASKINTERFACE PROPERTY	58
1.73. TYPE, TASKINTERFACE PROPERTY	58
1.74. WRITEOUTPUTSAtSTARTUP, TASKINTERFACE PROPERTY	59
1.74.1. StationInterface	60
StationInterface Functions	60
1.75. GETBASESTATIONINTERFACE, STATIONINTERFACE FUNCTION	60
1.76. GETSUBSTATIONINTERFACE, STATIONINTERFACE FUNCTION	60
1.77. GETTASK, STATIONINTERFACE FUNCTION	62
1.78. GETXMLSETTINGS, STATIONINTERFACE FUNCTION	62
1.79. ISINERROR, STATIONINTERFACE FUNCTION	63
StationInterface Properties	64
1.80. ERRORTHRESHOLD, STATIONINTERFACE PROPERTY	64
1.81. LASTERRORCODE, STATIONINTERFACE PROPERTY	64
1.82. LASTERRORSTRING, STATIONINTERFACE PROPERTY	65
1.83. NAME, STATIONINTERFACE PROPERTY	66
1.84. QUALITY, STATIONINTERFACE PROPERTY	66
1.85. STATECOMMANDVARIABLE, STATIONINTERFACE PROPERTY	67
1.86. STATIONTYPE, STATIONINTERFACE PROPERTY	68
1.86.1. SerialStationInterface	68
SerialStationInterface Functions	68
1.87. GETSUBSTATIONINTERFACE, SERIALSTATIONINTERFACE FUNCTION	68
1.88. GETXMLSETTINGS, SERIALSTATIONINTERFACE FUNCTION	69
SerialStationInterface Properties	70
1.89. BAUDRATE, SERIALSTATIONINTERFACE PROPERTY	70
1.90. BYTESIZE, SERIALSTATIONINTERFACE PROPERTY	71
1.91. CdTIMEOUT, SERIALSTATIONINTERFACE PROPERTY	71
1.92. CTS TIMEOUT, SERIALSTATIONINTERFACE PROPERTY	72
1.93. DSR TIMEOUT, SERIALSTATIONINTERFACE PROPERTY	72
1.94. FLOWCONTROL, SERIALSTATIONINTERFACE PROPERTY	73
1.95. KEEPPORTOPENED, SERIALSTATIONINTERFACE PROPERTY	73
1.96. PARITY, SERIALSTATIONINTERFACE PROPERTY	74
1.97. PORTID, SERIALSTATIONINTERFACE PROPERTY	74
1.98. RxTIMEOUT, SERIALSTATIONINTERFACE PROPERTY	75
1.99. SIZERECEIVINGQUEUE, SERIALSTATIONINTERFACE PROPERTY	76
1.100. SIZETRANSMISSIONQUEUE, SERIALSTATIONINTERFACE PROPERTY	77
1.101. STOPBITS, SERIALSTATIONINTERFACE PROPERTY	77
1.102. TxTIMEOUT, SERIALSTATIONINTERFACE PROPERTY	77
1.102.1. TAPISStationInterface	78
TAPISStationInterface Functions	78
1.103. GETBASESTATIONINTERFACE, TAPISSTATIONINTERFACE FUNCTION	78
1.104. GETSUBSTATIONINTERFACE, TAPISSTATIONINTERFACE FUNCTION	79
1.105. GETXMLSETTINGS, TAPISSTATIONINTERFACE FUNCTION	79

1.106. IsCONNECTED, TAPISTATIONINTERFACE FUNCTION	80
TAPIStationInterface Properties	81
1.107. DISCONNECTAFTERSECS, TAPISTATIONINTERFACE PROPERTY	81
1.108. ENABLETAPICALLONTHISSTATION, TAPISTATIONINTERFACE PROPERTY	82
1.109. ENDCONNECTIONTIME, TAPISTATIONINTERFACE PROPERTY	83
1.110. LASTCONNECTIONTIME, TAPISTATIONINTERFACE PROPERTY	83
1.111. LASTTAPIERROR, TAPISTATIONINTERFACE PROPERTY	84
1.112. LASTTAPIERRORSTRING, TAPISTATIONINTERFACE PROPERTY	84
1.113. PHONENUMBER, TAPISTATIONINTERFACE PROPERTY	84
1.114. PROMPTFORCONNECTION, TAPISTATIONINTERFACE PROPERTY	85
1.115. RETRIES, TAPISTATIONINTERFACE PROPERTY	85
1.116. RETRYAFTERSECS, TAPISTATIONINTERFACE PROPERTY	86
1.117. SHOWCONNECTIONDLG, TAPISTATIONINTERFACE PROPERTY	86
1.118. STARTCONNECTIONTIME, TAPISTATIONINTERFACE PROPERTY	86
1.119. TOTALCONNECTIONTIME, TAPISTATIONINTERFACE PROPERTY	88
1.119.1. <i>SerialTAPI</i>	88
SerialTAPIFunctions	88
1.120. GETBASESTATIONINTERFACE, SERIALTAPIHALFDUPLEXBASESTATIONINTERFACE FUNCTION	88
1.121. GETSUBSTATIONINTERFACE, SERIALTAPIHALFDUPLEXBASESTATIONINTERFACE FUNCTION	89
1.122. GETXMLSETTINGS, SERIALTAPIHALFDUPLEXBASESTATIONINTERFACEFUNCTION	90
SerialTAPIProperties	91
1.123. DELAYBEFOREFIRSTCHAR, SERIALTAPIHALFDUPLEXBASESTATIONINTERFACE PROPERTY	91
1.124. DELAYAFTERLASTCHAR, SERIALTAPIHALFDUPLEXBASESTATIONINTERFACE PROPERTY	92
1.125. HALFDUPLEXSIGNAL, SERIALTAPIHALFDUPLEXBASESTATIONINTERFACE PROPERTY	93
1.125.1. <i>TCPIPStationInterface</i>	93
TCPIPStationInterface Functions	93
1.126. GETBASESTATIONINTERFACE, TCPIPSTATIONINTERFACE FUNCTION	93
1.127. GETSUBSTATIONINTERFACE, TCPIPSTATIONINTERFACE FUNCTION	94
1.128. GETXMLSETTINGS, TCPIPSTATIONINTERFACE FUNCTION	95
TCPIPStationInterface Properties	95
1.129. BACKUPSERVERADDRESS, TCPIPSTATIONINTERFACE PROPERTY	95
1.130. LOCALBOUNDADDRESS, TCPIPSTATIONINTERFACE PROPERTY	96
1.131. LOCALBOUNDPORT, TCPIPSTATIONINTERFACE PROPERTY	97
1.132. MAXRECEIVE, TCPIPSTATIONINTERFACE PROPERTY	97
1.133. MAXSEND, TCPIPSTATIONINTERFACE PROPERTY	98
1.134. RXTIMEOUT, TCPIPSTATIONINTERFACE PROPERTY	98
1.135. SERVERADDRESS, TCPIPSTATIONINTERFACE PROPERTY	99
1.136. SERVERPORT, TCPIPSTATIONINTERFACE PROPERTY	100
1.137. SWITCHSERVERTIMEOUT, TCPIPSTATIONINTERFACE PROPERTY	101
1.138. TXTIMEOUT, TCPIPSTATIONINTERFACE PROPERTY	101
1.138.1. <i>RASStationInterface</i>	102
RASStationInterface Functions	102
1.139. GETBASESTATIONINTERFACE, RASSTATIONINTERFACE FUNCTION	102
1.140. GETSUBSTATIONINTERFACE, RASSTATIONINTERFACE FUNCTION	103
1.141. GETXMLSETTINGS, RASSTATIONINTERFACE FUNCTION	103
1.142. IsCONNECTED, RASSTATIONINTERFACE FUNCTION	104
RASStationInterface Properties	105
1.143. DISCONNECTAFTERSECS, RASSTATIONINTERFACE PROPERTY	105
1.144. ENABLERASCALLONTHISSTATION, RASSTATIONINTERFACE PROPERTY	105
1.145. ENDCONNECTIONTIME, RASSTATIONINTERFACE PROPERTY	105
1.146. LASTCONNECTIONTIME, RASSTATIONINTERFACE PROPERTY	106
1.147. LASTRASERRORNUMBER, RASSTATIONINTERFACE PROPERTY	106
1.148. LASTRASERRORSTRING, RASSTATIONINTERFACE PROPERTY	107
1.149. NUMRETRIES, RASSTATIONINTERFACE PROPERTY	107
1.150. PASSWORD, RASSTATIONINTERFACE PROPERTY	107
1.151. PHONEBOOKENTRY, RASSTATIONINTERFACE PROPERTY	108
1.152. PHONENUMBER, RASSTATIONINTERFACE PROPERTY	109
1.153. PROMPTFORCONNECTION, RASSTATIONINTERFACE PROPERTY	109
1.154. RETRYAFTERSECS, RASSTATIONINTERFACE PROPERTY	110
1.155. SHOWCONNECTIONDLG, RASSTATIONINTERFACE PROPERTY	110

1.156. STARTCONNECTIONTIME, RASSTATIONINTERFACE PROPERTY	110
1.157. TOTALCONNECTIONTIME, RASSTATIONINTERFACE PROPERTY	111
1.158. USERNAME, RASSTATIONINTERFACE PROPERTY	112
1.159. ERRORI	112
1.160. DESCRIZIONE ERRORI	112
1.161. ABOUT	119

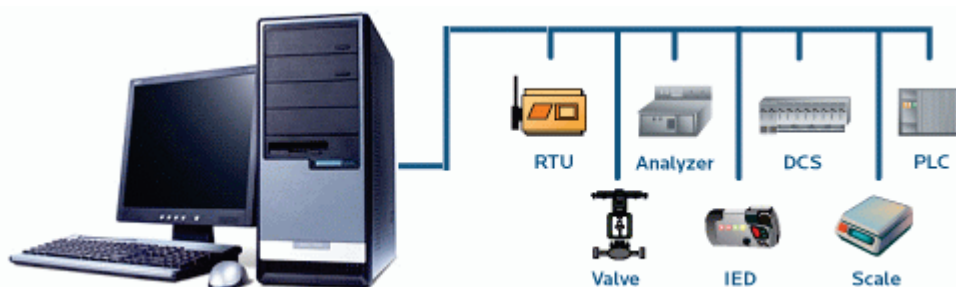
1. Driver di Comunicazione

1.1. Introduzione ai Driver

Gli argomenti trattati in questa sezione riguardano le impostazioni che devono essere effettuate per i Driver di Comunicazione del Supervisore, presenti nella risorsa Real Time DB di ogni progetto di supervisione.

I driver sono costituiti da librerie dinamiche (file .DLL) che, secondo una logica "exception-based", provvedono a notificare le informazioni ricevute dalle aree di memoria del dispositivo collegato alle aree di memoria del Supervisore e viceversa, secondo le impostazioni predefinite.

Tramite le impostazioni del Driver si determina l'associazione tra le variabili del campo e le variabili del Supervisore. Il sistema, tramite la porta seriale, il fieldbus o la rete utilizzata, provvederà a leggere e scrivere le variabili "da" e "verso" l'impianto, secondo le modalità impostate.



La tecnologia adottata nei driver del Supervisore prevede sofisticati meccanismi di thread pooling, tali per cui le informazioni sono scambiate con il campo nel modo più efficiente possibile, gestendo una ottimizzazione automatica della comunicazione secondo l'utilizzo effettivo delle variabili nel progetto in esecuzione.

Infatti, solo le variabili **effettivamente in uso** dal sistema saranno scambiate con il campo, lasciando al driver il compito di ottimizzare e rendere efficiente la comunicazione.

I Driver del Supervisore includono inoltre potenzialità tali da rendere la comunicazione del Supervisore estremamente potente e flessibile.

Grazie alle proprie caratteristiche, comuni a tutti i driver, è possibile ottenere:

1. Comunicazione efficiente ed ottimizzata
2. Link agli indirizzi PLC gestibile direttamente nei Tag oppure indirettamente tramite "incarichi"
3. Configurabilità runtime tramite interfaccia Script VBA
4. Importazione automatica del database del dispositivo
5. Funzione di Bridging per consentire l'accesso trasparente al dispositivo dall'esterno via modem (es. teleservice)
6. Funzioni TAPI per consentire la chiamata automatica ai dispositivi remoti in seriale via modem
7. Funzioni RAS per consentire la chiamata automatica ai dispositivi remoti ethernet via modem
8. Funzioni avanzate di debugging
9. Test cablaggio e comunicazione immediato e diretto

1.2. Il Driver nel Progetto

Un progetto di supervisione è costituito dall'insieme delle risorse e degli oggetti e una volta compilato ed elaborato in Runtime, provvederà a gestire le informazioni, le logiche e l'interfaccia utente secondo quanto desiderato.

La comunicazione con il campo è determinata dall'utilizzo dei Driver di Comunicazione (**oltre che dalle tecnologie OPC**). I Driver, sotto forma di librerie dinamiche, hanno il compito di leggere o scrivere le aree di memoria del dispositivo collegato alle aree di memoria gestite dal progetto.



Ad esempio, in caso di apparecchiatura PLC, il driver utilizzerà il protocollo di comunicazione previsto dal dispositivo per leggere o scrivere le aree di

memoria dal PLC al Supervisore e viceversa, secondo la configurazione e le associazioni di dati impostate nelle proprietà del driver e/o nelle proprietà delle variabili del progetto di supervisione.

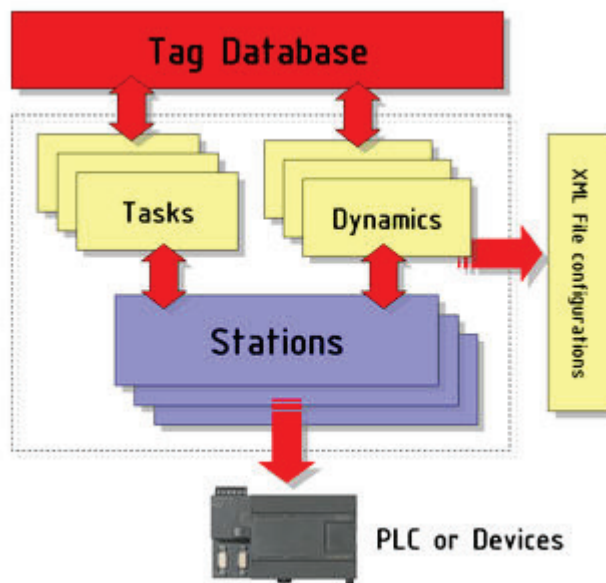
Le aree di memoria nelle quali il driver potrà leggere o scrivere e le modalità di trasmissione sono determinate dal costruttore del dispositivo, pertanto è necessario prima di tutto attenersi alle istruzioni previste per lo specifico dispositivo al quale ci si desidera collegare.



Leggere ed attenersi attentamente alle specifiche del costruttore hardware del dispositivo per il collegamento e l'associazione dei dati tra dispositivo e Supervisore.

Indipendentemente dal protocollo e dal costruttore hardware, i driver del Supervisore permettono la configurazione visuale al programmatore, consentendo di gestire la comunicazione secondo le possibili configurazioni previste.

La struttura di un driver di comunicazione del Supervisore è descritta nello schema illustrativo:



Secondo lo schema illustrato in figura, il driver gestisce a basso livello il protocollo di comunicazione.

1. Tramite il concetto di "**Stazione**", il driver richiede di definire i parametri principali della comunicazione (a seconda se si tratti di driver seriali o di rete, occorre definire i relativi parametri).
2. Tramite il concetto di "**Task**", il driver permette di definire l'associazione "indiretta" tra gli indirizzi del dispositivo e le variabili del progetto del Supervisore. I Task offrono la possibilità quindi definire come comunicare per blocchi di dati, definendo una variabile (o un gruppo di variabili) in associazione ad un indirizzo (o indirizzo di partenza) del dispositivo. In tal modo l'utente può configurare, in modo indiretto e quindi indipendente dal progetto, i link alle aree di memoria del dispositivo.
3. Tramite il concetto di "**Indirizzi Dinamici**" il driver permette di consentire l'associazione diretta dell'indirizzo di memoria nelle proprietà della variabile nel progetto (Tag). In questo modo, la variabile punta direttamente all'indirizzo nel dispositivo, lasciando al driver il compito di creare "dinamicamente" degli incarichi di comunicazione che gestirà sempre in maniera ottimizzata.
4. Il driver ha sempre a che fare quindi con il "**Realtime Database**" del progetto del Supervisore. Le variabili sono associabili quindi direttamente (proprietà dei Tag) oppure indirettamente (Task). In ogni caso la comunicazione è ottimizzata secondo il concetto di "variabili in uso".
5. La configurazione del driver è salvata su appositi **file XML** nella cartella "Risorse" del progetto". I file sono basati sul metalinguaggio XML, come tutto il progetto, per la massima trasparenza. I file del driver sono:
<nome_driver>.drvsettings = file contenente le impostazioni generali del driver
<nome_driver>.dynsettings = file generato all'avvio runtime con le caratteristiche degli incarichi dinamici calcolati

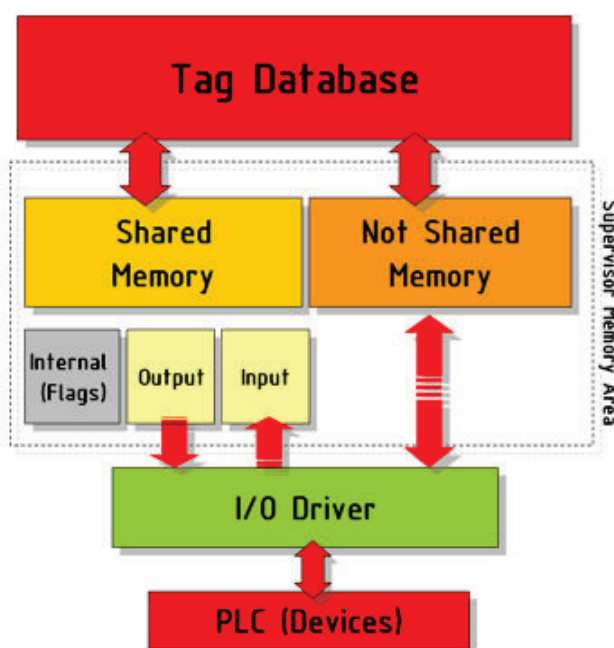


I Driver del Supervisore sono librerie che possono essere anche utilizzate indipendentemente dai progetti del Supervisore, ovvero possono essere inserite anche in altri ambienti di programmazione compatibili con la tecnologia ActiveX.

1.3. Link alle aree di memoria

Le variabili Tag del Realtime Database del Supervisore vengono mappate dinamicamente nella memoria del PC, a prescindere dalle tecniche di comunicazione utilizzate. La mappatura nelle aree di memoria dei Tag del Supervisore è definibile nelle "proprietà generali" di ogni singolo tag.

Le variabili possono essere mappate in due aree dati differenti, l'area dati definita "**Non Condivisa**" (Not Shared Memory, proposta per default) e l'area dati definita "**Condivisa**" (Shared memory).



Area Supervisore "Non Condivisa"

L'area dati "Non Condivisa" è l'area proposta per default quando si creano le variabili Tag nel progetto di supervisione. Utilizzando l'area Non Condivisa, si lascia che il Supervisore decida come e dove allocare nella memoria le proprie variabili Tag.

In tal modo, l'utente non deve preoccuparsi di allocare le variabili nè è in grado di sapere dove il Supervisore allocherà in runtime le proprie variabili. Questo permette all'utente di non preoccuparsi di assegnare indirizzi oltre quello relativo all'eventuale link al dispositivo.

E' di norma preferibile utilizzare quindi questa selezione, evitando così potenziali rischi di allocazioni di dati non corrette internamente al Supervisore provvederà automaticamente a gestire le proprie informazioni internamente, comunicando con il driver solo se la variabile risulterà connessa al dispositivo.

Area Supervisore "Condivisa"

L'area dati "Condivisa", se selezionata, permette di specificare l'allocazione del dato nelle aree di memoria interna del Supervisore, a prescindere dalla comunicazione del driver. La selezione di quest'area richiede **obbligatoriamente di assegnare il tipo di area e l'indirizzo assoluto del Tag nella memoria del Supervisore**.

E' quindi richiesto al programmatore di verificare la corretta assegnazione dell'indirizzo, onde evitare la sovrapposizione non desiderata delle variabili in memoria.

L'area Condivisa prevede di selezionare il **Tipo di Area**, scegliendo tra:

1. **Area di Input**
2. **Area di Output**
3. **Area Flag (Interna)**

Indipendentemente dal tipo di area selezionata sarà possibile scambiare le variabili dell'Area Condivisa con il campo utilizzando incarichi di Input, Output o Input/Output.

Link al dispositivo

Nei Task viene definito il collegamento tra le variabili del Supervisore e le aree di memoria del dispositivo. Il Supervisore offre la possibilità di realizzare i Task di comunicazione in due modi differenti: Task Statici e Task Dinamici:

Task Statici: Task definiti dal programmatore che dovrà definire i pacchetti di variabili da scambiare e i parametri di comunicazione

Task Dinamici: Task creati automaticamente dal Driver all'avvio del progetto in base al collegamento inserito nella proprietà "Indirizzo Fisico I/O" delle variabili del progetto. In questo caso sarà il Driver a gestire il raggruppamento delle variabili al fine di ottimizzare al meglio la comunicazione

1.4. Indirizzamento su Variabili Dinamiche

Tag Dinamici

Il Supervisore permette di specificare l'indirizzo del dispositivo direttamente nelle proprietà generali della variabile Tag, nella risorsa RealTime Database del progetto. Utilizzando questa tecnica, driver provvederà a generare in modo dinamico gli incarichi di comunicazione, secondo i concetti di ottimizzazione e raggruppamento dei dati predefiniti. Infatti il driver, all'avvio del progetto, provvederà a generare un numero di incarichi adeguato, raggruppando i dati a blocchi e comunicando solo se le variabili andranno in uso nel progetto.

Per assegnare l'indirizzo in modo dinamico, occorre:

1. Dalle proprietà Generali della variabile, selezionare "Indirizzo Fisico I/O"
2. Dalla finestra Esploratore Tag, selezionare il Tabs Driver
3. Fare doppio clic sul driver desiderato nella lista. Apparirà la finestra di assegnazione dell'indirizzo
4. Specificare l'indirizzo nell'area di memoria assoluta del dispositivo collegato
5. Specificare il tipo di gestione dei dati (lettura, scrittura o entrambe) nel campo Type

La selezione dell'indirizzo relativo al dispositivo può avvenire, oltre che per configurazione, anche specificando direttamente la sintassi di indirizzamento, che dovrà generalmente prevedere:

[DRV]<Nome del driver>.Sta=< Nome della stazione>|Addr=< Indirizzo del dispositivo>



Occorre tuttavia riferirsi alle informazioni specifiche relative agli indirizzamenti di ogni specifico driver.

Concetti sugli Incarichi Dinamici

Come indicato precedentemente, il Supervisore provvederà a creare dinamicamente, all'avvio del progetto, il numero di incarichi necessario a gestire la comunicazione.

Il parametro determinante per la generazione automatica degli incarichi, nelle proprietà del driver, è il **"Minimum Threshold"**. Tramite questo parametro si stabilisce al driver il numero minimo come soglia di frammentazione nella generazione dell'incarico.

Ad esempio:

VAR00001 è collegata all'indirizzo word 0 del dispositivo
VAR00002 è collegata all'indirizzo word 3 del dispositivo
VAR00003 è collegata all'indirizzo word 12 del dispositivo
VAR00004 è collegata all'indirizzo word 18 del dispositivo

Di default, il parametro "Minimum Threshold" del driver è impostato a 5. In questo esempio, all'avvio del progetto, il driver creerà dinamicamente il file XML ".dynsettings" nella cartella "Risorse" del progetto, nel quale provvederà a calcolare la generazione di nr. 2 incarichi dinamici al driver.

Nel primo incarico si leggeranno le word da 0 a 3, nel secondo incarico (necessario perchè la word successiva da leggere è ad un indirizzo superiore a 5 byte come indicato dal parametro "Minimum Threshold") si leggeranno le word da 12 a 18.

Ipotizziamo ora di impostare al valore 20 il parametro "Minimum Threshold". Al successivo avvio del progetto, il driver creerà dinamicamente un solo incarico, leggendo dalla word 0 alla word 18.



Il numero di incarichi generato automaticamente dipenderà quindi dal valore impostato come parametro "Minimum Threshold".

Le variabili associate al driver in modo "**dinamico**" saranno di default gestite in lettura-scrittura automaticamente dal driver. Il driver deciderà, a seconda delle possibilità dell'area del dispositivo associata, se gestire in sola lettura, in sola scrittura o in lettura-scrittura. E' possibile tuttavia modificare questa impostazione agendo nel file XML ".dynsettings" generato automaticamente all'avvio del progetto in run. Il dato nel file XML riguardante il tipo di esecuzione è il seguente:

<NameType> dove, al valore 1 assegnato per default, è possibile modificare con i seguenti valori:

- 0 = Input
- 1 = Input/Output
- 2 = Output su eccezione
- 3 = Output continuo

1.5. Task: Incarichi statici

Il Supervisore permette di assegnare in modo indiretto, tramite i Task, l'associazione tra indirizzi del dispositivo e variabili del Realtime Database del progetto. Utilizzando questa tecnica, il driver richiederà di specificare, nelle proprietà di configurazione di ogni singolo Task, l'associazione tra le variabili definite del progetto e le aree di memoria del dispositivo. Occorrerà inoltre definire il tipo di comunicazione (lettura, scrittura o entrambi).

L'elenco dei task generati sarà salvato sul file XML ".drvsettings" e resterà pertanto indipendente dal progetto.

Per inserire e configurare i Task, occorre:

1. Dalle impostazioni del driver, selezionare la scheda Task
2. Inserire un nuovo incarico Task utilizzando il pulsante "Add"
3. Configurare le proprietà dell'incarico secondo le necessità di scambio dati
4. Confermare con OK e procedere con gli eventuali incarichi successivi



Nota importante: per definire, in un incarico, una lista di variabili collegate a partire da un indirizzo del dispositivo, occorre digitare manualmente, nella proprietà Variabile, i nomi delle variabili separati dal carattere " ; ".

Un Task creato con questa procedura può includere una o più variabili, che dovranno essere dello stesso tipo (es. tutte Word, tutte Float, ecc.) e leggere/scrivere un'area consecutiva della memoria del PLC. L'indirizzo di partenza da specificare nel task sarà l'indirizzo della prima variabile da leggere sul PLC che verrà in questo modo associata alla prima variabile della lista del Task.

Soltanto pochi driver consentono la creazione di Task che includono variabili di diverso tipo (es. Byte, Word, ecc.). Per maggiori informazioni consultare la guida specifica dei singoli driver.

Occorre tenere presente i seguenti principi di esecuzione dei Task:

Task di Lettura	Leggono dal dispositivo collegato e scrivono sulle variabili del Supervisore. Possono essere soggetti all'esecuzione su evento. In tal caso l'incarico sarà eseguito solo quando le logiche interne del Supervisore attiveranno le variabili corrispondenti all'esecuzione dell'incarico. In tal caso la lettura dei dati verso il dispositivo verrà eseguita, in caso contrario sarà ignorata. Se l'incarico di lettura non è soggetto ad esecuzione su evento, l'incarico sarà eseguito con tecnologia "polling" insieme agli altri incarichi di lettura.
Task di Scrittura	Scrivono i dati delle variabili del Supervisore sul dispositivo collegato. Gli incarichi di scrittura sono eseguiti dal Supervisore con tecnologia "Event-driven", cioè solo quando un dato dell'incarico è variato nel Supervisore ed è quindi necessario notificarlo al PLC o al dispositivo collegato. Il driver consente comunque la scrittura con aggiornamento continuo dei dati.
Task di Lettura e Scrittura	Gli incarichi di lettura-scrittura sono gestiti in "polling" per mantenere aggiornati i dati letti dal dispositivo collegato, mentre la scrittura dei valori, se modificati nel Supervisore, viene eseguita solo su evento, generando la riscrittura del dato cambiato verso il dispositivo collegato. Gli incarichi di Lettura/Scrittura eseguono comunque sempre prima la lettura del dato dal dispositivo e poi se necessario eseguono la scrittura.
Task COM (OLE2)	Il Supervisore può gestire anche incarichi non configurati direttamente nel driver, ma generati in runtime da logiche Basic Script. In tal caso gli

incarichi saranno comunque eseguiti dal driver secondo la programmazione effettuata, eseguendo incarichi di lettura o scrittura in modo sincrono o asincrono.

L'utilizzo dei Task richiede una maggiore attenzione da parte del programmatore, ma lascia in compenso una maggiore flessibilità di impiego, consentendo al programmatore di decidere lui stesso (e non lasciarlo decidere al Supervisore) come scambiare i dati.

Gli incarichi di comunicazione vengono eseguiti solo per Nome della variabile e non per Indirizzo assoluto.



Gli incarichi, ricordiamo, consentono anche l'utilizzo dell'interfaccia COM (Component Object Model) per l'eventuale manipolazione o generazione degli incarichi di comunicazione al driver tramite **script VBA**.

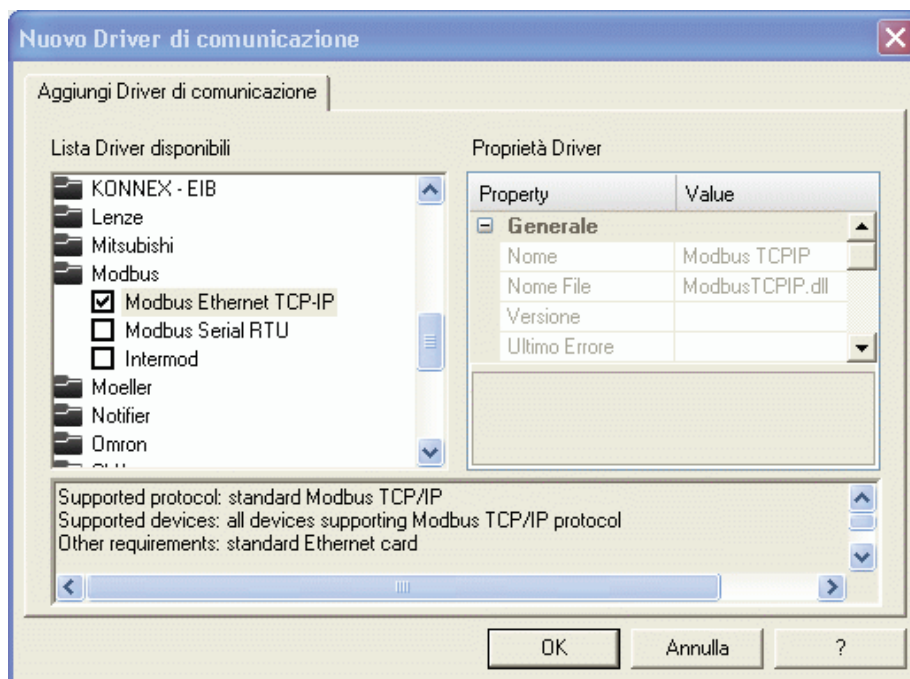
1.6. Installazione Driver

L'installazione del Supervisore provvede ad installare automaticamente la libreria dei driver disponibili, insieme alla piattaforma.

I Driver del Supervisore, essendo costituiti da semplici file .DLL, possono essere facilmente ampliati o aggiornati in modo indipendente dalla piattaforma di sviluppo. Per aggiornare o installare un nuovo driver, è sufficiente copiare il file .DLL relativo nella cartella Driver, che si trova all'interno della cartella di installazione del Supervisore (tipicamente C:\Programmi\Progea\Movicon11.3\Drivers).

Il programmatore potrà scegliere di volta in volta, nelle varie applicazioni, quale driver di comunicazione inserire e configurare, in funzione delle proprie necessità, selezionandolo dalla lista a disposizione.

L'inserimento e l'impostazione di un driver di comunicazione avviene in modalità programmazione del Supervisore, tramite la risorsa "Lista Comm.Driver" del gruppo "Real Time DB" nella finestra "Esploratore Progetto". Attivando il comando "Aggiungi Nuovo Driver" verrà visualizzata l'apposita finestra di dialogo contenente l'elenco dei driver disponibili.



La finestra per la selezione di un driver di comunicazione riporta, nella parte bassa, alcune informazioni importanti legate al driver quali:

- Breve descrizione del protocollo
- Elenco dei dispositivi supportati
- Schede o librerie necessarie
- Eventuali limitazioni del driver: aree non supportate, connessioni con più PLC non possibile, ecc.

Una volta inserito il driver, sarà possibile configurarlo tramite la "Finestra delle Proprietà" del Supervisore.

E' naturalmente possibile inserire più di un driver di comunicazione in un singolo progetto, compatibilmente con le opzioni definite sulla chiave hardware.

Nell'elenco delle 'features' dei drivers di comunicazione è stata aggiunta una nuova voce "Renaming Manager" che consente di sapere se il driver supporta la gestione delle rinomine. I drivers che supportano la gestione delle rinomine, e quindi hanno tale opzione a 'true', consentono di visualizzare le variabili con il nome modificato all'interno delle loro finestre di configurazione; e di supportare il comando di applicazione delle rinomine.

1.6.1. File di Configurazione dei Driver di Comunicazione

Quando in un progetto viene aggiunto un nuovo driver di comunicazione è possibile fare in modo che il driver vada a caricare un file ".drvsettings" personalizzato e già esistente. In questo modo il driver potrà avere delle impostazioni iniziali predefinite dal programmatore e diverse da quelle di default. Durante l'installazione di Visu+ verranno create tutte o solo alcune delle cartelle sotto indicate all'interno della cartella di installazione di Visu+ e all'interno di tali cartelle saranno già presenti alcuni file di configurazione "drvsettings" dei driver:

```
[Install Folder]\drvsettings\WinCE\  
[Install Folder]\drvsettings\Win32\  
[Install Folder]\drvsettings\ClientCE\  
[Install Folder]\drvsettings\ClientWin32\  
[Install Folder]\drvsettings\ClientJ2SE\  
[Install Folder]\drvsettings\ClientJ2ME\
```

Come si può notare sono presenti sei sottocartelle, una per ogni piattaforma selezionabile nel progetto. In questo modo è possibile definire diversi file di configurazione per uno stesso driver a seconda del tipo di piattaforma selezionata. Se una o più cartelle tra quelle sopra indicate non sono presenti è possibile aggiungerle manualmente.

Quando nel progetto viene aggiunto un driver di comunicazione, Visu+ andrà a ricercare se esiste un file con lo stesso nome della dll del driver ed estensione ".drvsettings" (ad esempio "ModbusTCPIP.drvsettings") all'interno della cartella corrispondente alla piattaforma selezionata per il progetto. Se il file viene trovato allora verrà copiato nella cartella delle risorse del progetto (la data e ora del file verranno impostate con la data e ora attuali). Se il file non è invece presente la finestra di impostazione del driver verrà aperta utilizzando i valori di default.

Ad esempio, se il progetto è stato impostato con la piattaforma "Windows CE", e nel progetto viene aggiunto il driver "Modbus TCPIP", Visu+ verificherà la presenza di un file nominato "ModbusTCPIP.drvsettings" all'interno della cartella "[Install Folder]\drvsettings\WinCE \", e nel caso questo venga trovato lo copierà nella cartella delle risorse del progetto.

Se nel progetto sono state abilitate più piattaforme l'ordine di priorità con cui verrà selezionato il file ".drvsettings" da copiare sarà il seguente: WinCE, Win32, ClientCE, ClientWin32, ClientJ2SE, ClientJ2ME.

Nel caso in cui all'inserimento di un driver nel progetto esista già il suo relativo file ".drvsettings" nella cartella delle risorse, tale file verrà lasciato inalterato e non verrà quindi sovrascritto.

I file di configurazione creati dall'installazione di Visu+ potranno essere eventualmente modificati a seconda delle specifiche esigenze.

1.7. Duplicazione di Driver

Il Supervisore prevede normalmente la possibilità di installare due o più driver nel sistema, selezionandoli dalla lista di quelli disponibili come visto nei paragrafi precedenti. Qualora si desiderasse però installare ed utilizzare due o più driver identici tra di loro (quindi per PLC o dispositivi dello stesso tipo), occorre seguire alcune semplici indicazioni, di seguito riportate.

Ad esempio può verificarsi la necessità di installare due o più driver dello stesso tipo qualora si desiderasse comunicare contemporaneamente, nell'ambito dello stesso sistema, con lo stesso protocollo per il dispositivo dello stesso tipo e non sia sufficiente definire semplicemente due Stazioni diverse.

Esempio: si desidera che il PC sul quale è installato il Supervisore comunichi con due dispositivi identici su due canali distinti di comunicazione del PC. Per eseguire una duplicazione del Driver di Comunicazione seguire i seguenti passi:

1. Duplicare la DLL del driver interessato che si trova nella cartella "Drivers" della cartella di installazione del Supervisore. Il file dovrà avere naturalmente un nome diverso da quello originale, ad esempio lo stesso nome più un indice incrementale
2. Aggiornare il file "Driver.xml" indicando il nuovo file .DLL appena duplicato ed indicandone il nome descrittivo. In tal modo nella lista dei Driver del Supervisore appariranno i due driver identici ma con nome e descrizione diversi



Quando un driver viene duplicato questo risulta essere a tutti gli effetti un nuovo driver, quindi sarà necessario prevedere un driver aggiuntivo anche nelle opzioni della licenza. In pratica utilizzando due driver, seppur dello stesso tipo perchè duplicati, la licenza dovrà essere abilitata per due driver di comunicazione.

Esempio Pratico:

Si desidera gestire un progetto dove si possano utilizzare due driver "Modbus Serial" distinti. i passi da seguire sono:

Duplicazione della DLL

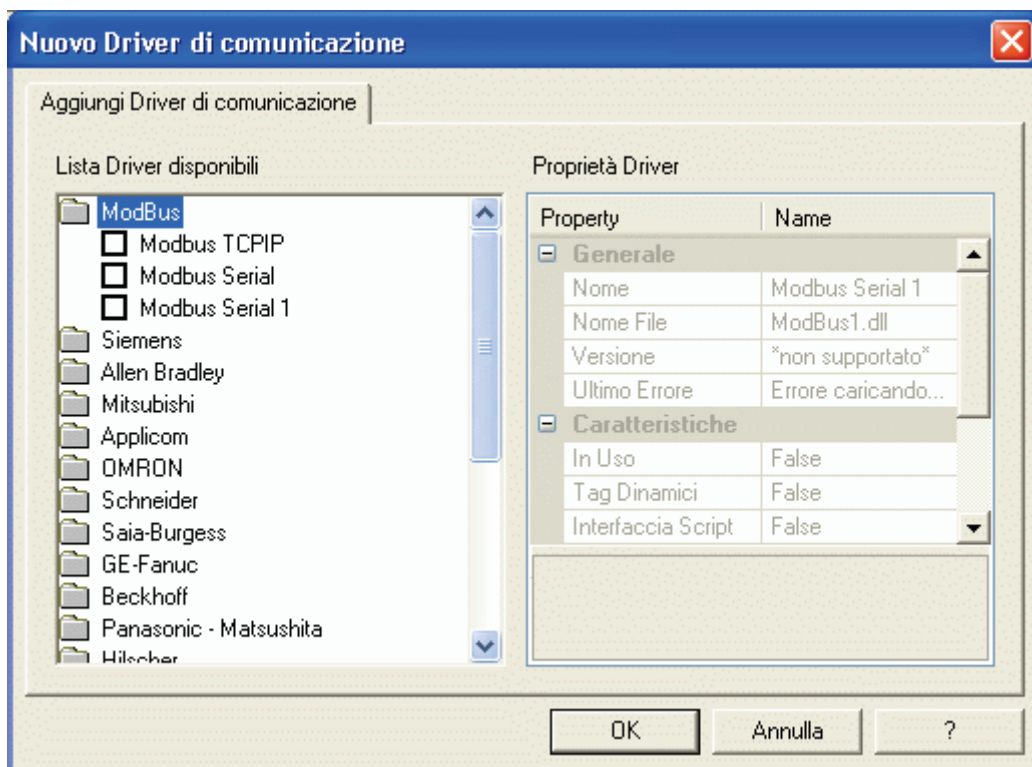
Nella sottocartella "Driver" di installazione del Supervisore eseguire una copia della dll originale "**Modbus.dll**" e nominarla ad esempio "**Modbus1.dll**"

Modifica del file Driver.xml

Dalla sottocartella "Driver" di installazione del Supervisore aprire con un editor di testo il file "Driver.xml". Nel tag <DriverFactory Factory="ModBus"> aggiungere la voce per il nuovo driver aggiunto, specificando una descrizione a piacere e il nome della nuova dll. Ad esempio <Driver Name="Modbus Serial 1">ModBus1.dll</Driver>. Il file modificato risulterà essere:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DriverList>
  <DriverFactory Factory="ModBus">
    <Driver Name="Modbus TCPIP">ModBusTCPIP.dll</Driver>
    <Driver Name="Modbus Serial">ModBus.dll</Driver>
    <Driver Name="Modbus Serial 1">ModBus1.dll</Driver>
  </DriverFactory>
  ...
  ...
```

In questo modo quando verrà aperta la finestra con la lista dei driver disponibili sarà presente anche quello appena duplicato.



1.8. Esecuzione Driver

Il driver o i driver di comunicazione installati verranno eseguiti all'avvio del progetto applicativo e rimarranno attivi per tutta la durata dell'elaborazione del progetto, secondo le modalità di esecuzione scelte o le logiche script VBA utilizzando l'interfaccia COM (Component Object Model) dei driver. Ad ogni attivazione della comunicazione con il campo, il sistema provvederà a registrare nel Log Storico un messaggio di notifica sullo stato della comunicazione.



La rappresentazione di un Led di colore verde sulla Barra di Stato nel bordo inferiore della finestra del Supervisore (se visualizzata) indica la corretta comunicazione con il campo del driver installato. Il colore rosso del led invece indica la presenza di un errore sulla comunicazione.

Eventuali problemi di comunicazione (cavo, connessioni, impostazioni, ecc.) provocheranno errori di comunicazione che saranno segnalati dal driver, evidenziati nella Barra di Stato e registrati nel Log Storico.



Nota: i driver sono indipendenti dal progetto, e la loro configurazione viene salvata su appositi file separati, individuabili dall'estensione ".drvsettings" e "dynsettings". Questa filosofia consente di mantenere intatto il progetto al cambiare del PLC o del dispositivo di comunicazione.

L'esecuzione di un driver nel progetto, durante il funzionamento Runtime del Supervisore, può essere subordinata alle condizioni stabilite dal programmatore.

1.9. Errori Hardware RS232

L'autodiagnosi dei Driver di Comunicazione del Supervisore prevede l'emissione di codici di errori hardware nella comunicazione, secondo quanto fornito dal registro di stato del chip UART della seriale installata sul P.C.

Al proposito, raccomandiamo di utilizzare seriali con chip UART del tipo 16550A, che utilizzano code FIFO di 16 byte dati. Il tipo di seriale installata sul P.C. è facilmente rilevabile mediante l'esecuzione del file Microsoft Diagnostic MSD.EXE.

Gli errori hardware di comunicazione generalmente sono dovuti alle possibili seguenti cause:

- Disturbi sulla linea seriale
- Differenze di potenziale tra le masse delle apparecchiature
- Schede seriali inadeguate alle prestazioni desiderate
- Cavi di comunicazione difettosi o inadeguati
- Baud rate troppo alto per l'hardware utilizzato
- Guasti ai dispositivi di comunicazione

Gli errori hardware forniti dal driver rispettano il codice emesso sul registro degli errori del chip UART della seriale.

In caso di errore hardware, il Supervisore segnalerà un errore generico riportante un numero. Tramite il numero indicato, tradotto in binario, è possibile rilevare l'indicazione dell'errore o degli errori secondo il significato dei singoli bit come da tabella:

VALORE	CODE	SIGNIFICATO
1	RX OVER	La seriale ha ricevuto più caratteri di quelli ammissibili nel proprio buffer
2	OVERRUN	La seriale ha ricevuto un carattere prima che il precedente potesse essere elaborato dal sistema
4	RX PARITY	Errore nella parità dei dati, incongruenza tra la parità ricevuta e quella impostata
8	FRAME	Errore nella cornice dei dati. I dati ricevuti non rispettano le caratteristiche impostate (lunghezza, bit di stop, ecc.)
16	BREAK	Stato di Break (Attesa) richiesto dal partecipante



Tutti gli altri errori di comunicazione dipendono dallo specifico protocollo: riferirsi quindi alle indicazioni del driver specifico. I messaggi di errore dei driver sono visualizzati sulla barra di stato e visibili nel Log Storico.

1.10. Impostazioni Generali

1.11. General (Driver)

In questa scheda di impostazione si possono configurare alcune proprietà comuni a tutti i driver di comunicazione.

In genere non è necessario modificare le impostazioni di default.

Wait Time

Tempo di pausa, espresso in millisecondi, tra l'esecuzione di due incarichi (blocchi dati) di comunicazione successivi. Di default è impostato il valore 0 (nessuna pausa). La modifica del valore di default (0) può essere necessaria per dispositivi che richiedono un tempo di pausa tra interrogazioni successive (ad esempio dispositivi con scarse prestazioni).

Timeout

Tempo di timeout per l'esecuzione di incarichi sincroni. Il valore è espresso in millisecondi.

Minimum Threshold

Questo parametro determina la soglia minima per la frammentazione dei pacchetti di dati scambiati con il dispositivo. Come indicato nell'introduzione, nella gestione dinamica della comunicazione, il Supervisore calcola automaticamente (all'avvio del progetto) la dimensione e la quantità degli incarichi dinamici da creare al driver per la comunicazione relativa alle variabili dinamiche ("Indirizzo Fisico I/O" dei tag).

Il Supervisore infatti cercherà di ottimizzare la comunicazione aggregando il maggior numero possibile di dati in un unico incarico. Quando i dati sono collegati su indirizzi distanti tra loro, questo valore

determina la distanza in byte che consente al Supervisore di decidere se creare un nuovo incarico per il blocco dati successivo.

Ad esempio:

VAR00001 è collegata all'indirizzo word 0 del dispositivo
VAR00002 è collegata all'indirizzo word 3 del dispositivo
VAR00003 è collegata all'indirizzo word 12 del dispositivo
VAR00004 è collegata all'indirizzo word 17 del dispositivo

Di default, il parametro "Minimum Threshold" del driver è impostato a 5. In questo esempio, all'avvio del progetto, il driver creerà dinamicamente il file XML ".dynsettings" nella cartella "Risorse" del progetto, nel quale provvederà a calcolare la generazione di nr. 2 incarichi dinamici al driver.

Nel primo incarico si leggeranno le word da 0 a 3, nel secondo incarico (necessario perchè la word successiva da leggere è ad un indirizzo superiore a 5 byte come indicato dal parametro "Minimum Threshold") si leggeranno le word da 12 a 17.

Ipotizziamo ora di impostare al valore 20 il parametro "Minimum Threshold". Al successivo avvio del progetto, il driver creerà dinamicamente un solo incarico, leggendo dalla word 0 alla word 18.



Il numero di incarichi generato automaticamente dipenderà quindi dal valore impostato come parametro "Minimum Threshold".

Aggregation Limit

Questo parametro consente di specificare il numero massimo di byte che potranno essere aggregati per un incarico dinamico per ogni task. Lasciando a zero questo parametro il driver utilizzerà come limite il valore massimo imposto dal protocollo selezionato. La modifica di questo valore si rende necessaria qualora si utilizzino dispositivi per i quali il numero massimo di byte che si possono scambiare con un task è inferiore al limite del protocollo.

Synch.Startup

Questa opzione determina la sincronizzazione tra logiche e comunicazione del driver all'avvio del progetto.

Se si imposta a "true" quest'opzione, il Supervisore attenderà che gli incarichi statici di input siano stati completamente eseguiti prima di elaborare le logiche e gli script del progetto.

Questa opzione pertanto comporterà un tempo più lungo di avvio del progetto ma le logiche saranno eseguite con la certezza di valori di input "aggiornati".

VBA Interface

Indica se il driver supporta l'interfaccia COM (Component Object Model). L'interfaccia COM, definita anche OLE2, permette l'utilizzo delle logiche script VBA per la manipolazione del driver, secondo i metodi, le proprietà e gli eventi descritti nella apposita documentazione.

Polling Time

Questo parametro, espresso in millisecondi, rappresenta il tempo minimo di polling per l'esecuzione degli incarichi per l'aggiornamento dei dati **quando le variabili sono in uso**.

Questo valore verrà acquisito da tutti gli incarichi Dinamici, che quindi avranno tutti lo stesso Polling Time (solo per alcuni driver è possibile specificare un Polling Time per ogni incarico Dinamico). Per quanto riguarda gli incarichi Statici, invece, questo valore viene inserito come valore di default nell'omonima proprietà dei Task ("Polling Time") quando questi vengono creati. Successivamente sarà quindi possibile modificare questo valore tramite la proprietà del task Statico.

Impostando il valore uguale a 0 significa che i dati sono aggiornati con la massima velocità possibile.

Può essere impostato un valore più alto, ad esempio, qualora i dati non richiedano tempi di aggiornamento rapidi.

Unused Polling Time

Questo parametro, espresso in millisecondi, permette di forzare un aggiornamento dei dati anche quando le variabili **non sono in uso, stabilendo comunque un tempo di polling**.

Questo valore verrà acquisito da tutti gli incarichi Dinamici, che quindi avranno tutti lo stesso Unused Polling Time. Per quanto riguarda gli incarichi Statici, invece, questo valore viene inserito come valore di default nell'omonima proprietà dei Task ("Unused Polling Time") quando questi vengono creati. Successivamente sarà quindi possibile modificare questo valore tramite la proprietà del task Statico.



Impostando questo parametro = 0, se le variabili non sono in uso gli incarichi non saranno mai eseguiti.

Error Polling Time

Questo parametro, espresso in millisecondi, rappresenta il tempo durante il quale non verrà eseguito nessun task per la stazione che è in errore. Quindi quando una stazione va in errore il driver attenderà questo tempo prima di eseguire un nuovo tentativo.

Def. Refresh Time

Attualmente non utilizzato. Riservato per implementazioni future.

Protocol Priority

Questa casella di riepilogo permette di impostare la priorità dei thread di comunicazione, ovvero la priorità che deve avere l'esecuzione del driver rispetto agli altri processi del Supervisore.

I valori possibili, dalla priorità inferiore alla maggiore, sono:

- **Normal**
- **High**
- **Very High**
- **Real Time**



Si consiglia di modificare il valore di default (Normal) con estrema cautela, per evitare il rischio che il carico di lavoro della CPU diventi eccessivo (CPU al 100%).

Può essere utile provare ad aumentare la priorità del processo del driver qualora vi sia una necessità dovuta a carichi elevati di comunicazione. In tal caso si consiglia, comunque, di impostare il parametro Wait Time ad un valore diverso da 0, in modo da evitare che il driver utilizzi troppe risorse di sistema.

Suspend tasks in case of error

Questa proprietà sarà attiva solo per i driver che consentono l'indirizzamento simbolico dei task, "Tag Name", e resterà disattivata, con valore "True", per i driver che consentono solo l'indirizzamento numerico, "Address Type".

Impostando questa proprietà a "True", quando si verifica un errore di comunicazione, tutti gli incarichi verranno disabilitati, salvo il job che ha generato l'errore. Questo comportamento può risultare inadeguato nel caso di driver che, come il Beckhoff TwinCAT o l'Allen-Bradley Ethernet/IP, consentono l'utilizzo dell'indirizzamento simbolico (nomi delle variabili del dispositivo). In questo caso sarà quindi possibile impostare l'opzione a "False", consentendo così di mantenere attivi tutti gli incarichi anche in caso di errore. Impostando l'opzione "Suspend tasks in case of error" al valore "False" si avrà il seguente funzionamento:

il driver proseguirà con le interrogazioni successive anche nel caso in cui una interrogazione non vada a buon fine

una interrogazione non andata a buon fine verrà ritentata dopo il periodo di tempo indicato nel parametro "Error polling Time"

un qualsiasi errore di comunicazione andrà comunque a riflettersi sullo stato del bit di comunicazione (_SysVar_:CommDriverStatus)

Management of "In Use" state for structures

Questo parametro, se impostato a True, consente di mettere "In Use" tutti i membri di una variabile struttura anche se una sola parte di essi è effettivamente in uso nel progetto. In questo modo si avrà la certezza che tutta la variabile struttura verrà scambiata dal driver anche se soltanto uno dei suoi membri è in uso nel progetto. Impostando invece questa proprietà a "False" la gestione dello stato "In Use" verrà fatta singolarmente per ogni membro della struttura, e quindi solo i membri effettivamente in uso nel progetto verranno scambiati dal driver.



Attenzione! Impostando a True questa proprietà le performance di comunicazione potrebbero peggiorare in modo significativo, soprattutto se nel progetto ci sono molte variabili struttura che contengono a loro volta molti membri.

Direct Output for Input/Output

Tramite questo parametro è possibile modificare il comportamento del driver nell'eseguire gli incarichi dinamici di tipo "input/output". Se il suo valore è "False", nel caso in cui il valore della variabile sia stato modificato sulla supervisione, l'operazione di output viene preceduta da una operazione di input e poi, se il valore nel dispositivo non è cambiato, completata con l'operazione di output. Impostando il parametro a "True" invece è possibile eseguire un incarico di input/output direttamente in output, cioè senza eseguire una riletture.

1.12. Debug

In questa scheda di impostazione si possono configurare alcune proprietà riguardanti il Debug, comuni a tutti i driver di comunicazione.

In genere non è necessario modificare le impostazioni di default.

Debug Window

Selezionando "true", il driver mostrerà nella apposita finestra di Debug (visualizzabile dalla barra di Output dell'area di lavoro del Supervisore) tutti i messaggi diagnostici e di debug eventualmente generati dal driver.
Selezionando "false", la finestra di Debug non sarà visualizzata.

Max Entries

In questa casella è possibile definire il valore relativo al numero massimo di stringhe di messaggi diagnostici visualizzati nella finestra di debug, prima di ricircolare. Lasciando il valore di default, la finestra manterrà visualizzate le ultime 10.000 stringhe di messaggi.

Enable Log to file

Selezionando "true", il driver registrerà su un apposito file di Log tutti i messaggi diagnostici di Debug generati eventualmente dal driver.

Il numero di dati registrato sarà quello impostato nella impostazione precedente, mentre il nome ed il percorso del file sono definiti nella impostazione successiva.

Log FileName

In questa casella occorre specificare il nome ed il percorso del file per il Log diagnostico del driver.

Il file generato sarà un normale file di testo.

1.13. Impostazioni Stazioni

1.14. Stations

In questa scheda di impostazione occorre inserire ed configurare le "**stazioni di comunicazione**". Ogni driver dovrà necessariamente avere almeno una stazione di comunicazione, inserita e configurata



Ricordiamo infatti che il Supervisore offre la possibilità di impostare la comunicazione del driver sia tramite il concetto di "incarichi dinamici" che tramite il concetto dei "**Task**" (descritti nell'apposita sezione). Gli incarichi dinamici sono creati automaticamente dal driver all'avvio del progetto, in funzione dei link agli indirizzi del dispositivo definiti nelle proprietà "Indirizzo Fisico I/O" di ogni singolo tag).

Le stazioni di comunicazione permettono di definire come il driver deve gestire la comunicazione, dove ogni "stazione" rappresenta un canale di comunicazione verso uno dei dispositivi previsti.

Add

Il pulsante "Add" permette di inserire una nuova "stazione" per il driver di comunicazione. Inserendo una nuova stazione, verrà automaticamente visualizzata la relativa finestra di impostazione dei parametri di comunicazione richiesti.

La stazione o le stazioni inserite potranno in seguito essere modificate o rimosse utilizzando i pulsanti sotto indicati.

Edit

Il pulsante "Edit" permette di modificare i parametri di una "stazione" precedentemente inserita. Occorrerà pertanto selezionare la stazione desiderata e quindi utilizzare il pulsante "Edit" oppure il doppio clic.

Remove

Il pulsante "Remove" permette di eliminare una "stazione" precedentemente inserita. Occorrerà pertanto selezionare la stazione desiderata e quindi utilizzare il pulsante "Remove".

Test Cable/Comm.

Questo pulsante permette di eseguire un test di comunicazione immediato con il dispositivo. Grazie a questa utile funzionalità, il driver proverà a stabilire la comunicazione verificando quindi il corretto collegamento dei cablaggi.



Il test di comunicazione viene eseguito mediante l'invio in lettura di un dato, secondo criteri specifici di test dipendenti da ogni singolo protocollo.

E' opportuno pertanto riferirsi alla specifica di ciascun singolo driver per verificare che tipo di test viene eseguito. L'esito del test potrebbe infatti dipendere dallo specifico dispositivo.

Ad esempio, con protocolli generici come Modbus, il test prevede l'invio di una richiesta dati in lettura secondo un Function Code che non potrebbe non essere implementato sul dispositivo collegato.

Inoltre si ricorda che in ogni caso il test verifica solo la corretta configurazione del cavo e dei parametri generali, mentre è sempre cura del programmatore verificare le corrette associazioni di dati tra Supervisore e dispositivo.

1.15. General (Station)

In questa scheda di impostazione occorre definire, per la stazione selezionata, le impostazioni inerenti il gruppo di proprietà "Generali".

Station Name

Nome che identifica la stazione corrispondente al dispositivo con il quale si intende comunicare. Si tratta di un nome interno al driver.

Se si definiscono più stazioni, il nome di ciascuna di esse deve essere univoco.

Error Threshold

Nel caso in cui intervengano errori di comunicazione, questo parametro definisce il numero di errori raggiunto il quale il Driver notificherà effettivamente l'errore di comunicazione. Il contatore interno permette infatti, in presenza di comunicazioni disturbate, di non segnalare errore immediatamente ma di tentare di riprendere la comunicazione. Raggiunto il numero di tentativi, la segnalazione di errore verrà emessa dal driver.

State/Command Variable (gestita solo dai driver con versione di classe basata sulla build 250 in avanti)

Assegnando il nome di una variabile numerica definita nella supervisione (tipo consigliato Byte) a questa proprietà, è possibile controllare lo stato della comunicazione con la Stazione selezionata, abilitare / disabilitare la comunicazione con la stazione, avviare/arrestare le connessioni TAPI (solo per driver seriali) o RAS (solo per driver Ethernet), e per passare tra diversi server TCP / IP (solo per driver Ethernet). Vedere la seguente tabella per il significato di ogni bit della variabile.

E' necessario tenere presente che alcuni Bit possono essere utilizzati solo per verificare lo stato della stazione, mentre altri possono essere usati per impartire comandi alla stazione.

- Il bit 1 della variabile può essere utilizzato per controllare e modificare lo stato di abilitazione della Stazione (Active / Inactive, Abilitata/Disabilitata)
- I Bits 4, 5, 6 possono essere utilizzati per gestire una connessione TAPI (driver seriale) o una connessione RAS (driver Ethernet)

E' possibile utilizzare come variabile di stato una variabile di tipo Bit se è necessario la sola informazione sullo stato della comunicazione con la stazione.

Bit 0 (Stato)	Stato della Comunicazione: 0 = OK 1 = Errore
Bit 1 (Stato/Commando)	Stato della Stazione: 0 = Attiva/Abilitata 1 = Disattiva/Disabilitata
Bit 2 (Commando)	Cambia il Server TCP attivo
Bit 3 (Stato)	TCP Server attivo: 0 = Server primario 1 = Server di Backup
Bit 4 (Stato)	Stato della connessione Modem: 0 = non connesso 1 = connesso

Bit 5 (Commando)	Apri la connessione Modem
Bit 6 (Commando)	Chiude la connessione Modem
Bit 7 (Non usato)	Sempre a 0

Keep Opened

Permette di definire se il driver (**solo per driver seriali**) deve tenere aperta (e quindi sempre impegnata) la porta di comunicazione. Se impostato il valore = True, il driver verrà caricato all'avvio dell'esecuzione del progetto e terrà sempre aperta (impegnata) la porta di comunicazione associata. Impostando "False", dopo ogni operazione di "Lettura" o "Scrittura", il driver chiuderà, lasciandola libera, la porta seriale di comunicazione.

1.16. Serial Port Settings

In questo gruppo di impostazioni occorre definire, per la stazione selezionata, le configurazioni inerenti il gruppo di proprietà di Settaggio della porta seriale.

Utilizzando le impostazioni TAPI della stazione per gestire la comunicazione via modem, la seriale viene gestita dal driver del modem. Le impostazioni con le quali viene aperta la seriale sono quindi quelle stabilite nelle proprietà avanzate del driver del modem, che possono essere diverse da quelle impostate nella stazione del driver Movicon ("Serial Port Settings").

Port

Definisce il numero della porta seriale COM da utilizzare per la comunicazione.



Nota: occorre accertarsi che non sussistano conflitti in Windows nell'utilizzo delle porte. Ad esempio, installando le porte Com3 e Com4, occorre accertarsi che l'indirizzo assegnato e l'IRQ siano compatibili con la configurazione del PC. A tal scopo si consigliano schede seriali indirizzabili.

Baudrate

Definisce la velocità della comunicazione seriale (Baud Rate). Occorre impostare il valore di velocità di comunicazione corrispondente a quella del dispositivo con il quale si intende comunicare.

Byte Size

Definisce la quantità di byte richiesta dal protocollo di comunicazione interessato.

Parity

Definisce il tipo di parità richiesta dal protocollo di comunicazione interessato.

Stop Bits

Definisce il numero di Bit di Stop richiesto dal protocollo di comunicazione interessato.

Flow Control

Definisce il tipo di Controllo di Flusso dei dati per il tipo di comunicazione utilizzato. In particolare, permette di adattare la comunicazione secondo le necessità richieste a basso livello dalla porta seriale del dispositivo collegato. Di default il driver imposta "None" ovvero nessun controllo di flusso, tuttavia, in funzione del dispositivo, potrebbe essere necessario selezionare un tipo di Controllo Flusso (ad esempio in caso di segnalazione di errore con codice "1").

Le possibili selezioni sono:

None: Nessun Controllo di Flusso. Il controllo non è richiesto dal protocollo.

Hardware: Il Controllo di Flusso è gestito dai segnali elettrici della linea seriale (es. RTS, CTS, ecc.).

Xon/Xoff: Il controllo di flusso dei dati è di tipo Xon/Xoff.

NONE (Signal disabled): imposta il controllo di flusso a NONE ed imposta la seriale per disabilitare la gestione dei segnali DTR e RTS

RTS Toggle: imposta la seriale per gestire il segnale di controllo RTS in Toggle, ovvero la seriale mantiene alto il segnale fintanto che ci sono caratteri da spedire.

1.17. Queue Size

In questo gruppo di impostazioni occorre definire, per la stazione selezionata, le configurazioni inerenti la dimensione della coda (buffer) della porta seriale.



Generalmente, queste impostazioni sono riservate agli utenti esperti, pertanto si consiglia di mantenere le impostazioni di default.

Rx Queue

Definisce la quantità di byte di dati da gestire per la porta seriale come buffer (coda) di ricezione dei valori. Se non diversamente specificato, viene assunto il valore di default.

Per gli utenti esperti, è possibile modificare questo valore adattandolo alle necessità del tipo di sistema utilizzato.

Tx Queue

Definisce la quantità di byte di dati da gestire per la porta seriale come buffer (coda) di trasmissione dei valori. Se non diversamente specificato, viene assunto il valore di default.

Per gli utenti esperti, è possibile modificare questo valore adattandolo alle necessità del tipo di sistema utilizzato.

1.18. Timeouts

In questo gruppo di impostazioni occorre definire, per la stazione selezionata, le configurazioni inerenti i tempi di timeouts della comunicazione seriale.



Generalmente, queste impostazioni sono riservate agli utenti esperti, pertanto si consiglia di mantenere le impostazioni di default.

Rx Timeout

Definisce il valore di tempo, espresso in millisecondi, oltre il quale il driver deve notificare il timeout di comunicazione in ricezione.

Il timeout è riferito alla ricezione dei dati richiesti.

Tx Timeout

Definisce il valore di tempo, espresso in millisecondi, oltre il quale il driver deve notificare il timeout di comunicazione in trasmissione.

Il timeout è riferito alla trasmissione dei dati.

I parametri di seguito riportati si riferiscono ai soli driver seriali:

CD Timeout

Attualmente non utilizzato. Riservato per implementazioni future.

CTS Timeout

Definisce il valore di tempo, espresso in millisecondi, oltre il quale il driver deve notificare il timeout di comunicazione per il parametro seriale CTS. Definisce il tempo entro il quale deve essere completata la singola operazione di scrittura, a basso livello (API di Windows) sulla porta seriale.

DSR Timeout

Definisce il valore di tempo, espresso in millisecondi, oltre il quale il driver deve notificare il timeout di comunicazione per il parametro seriale DSR. Definisce l'intervallo di tempo massimo tra la ricezione di un carattere ed il successivo, inoltre concorre a definire il tempo massimo entro il quale deve essere completata la singola operazione di lettura, a basso livello (API di Windows) sulla porta seriale.

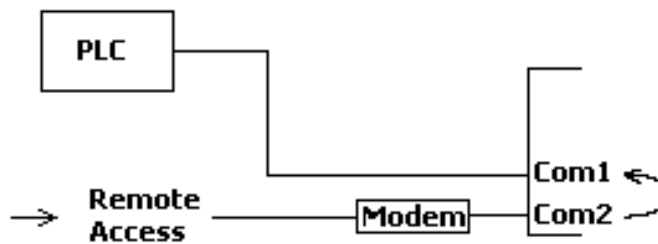
1.19. Bridging Service Settings

In questo gruppo di impostazioni occorre definire, per la stazione selezionata, le configurazioni inerenti la funzione di Bridging.

Le funzioni di Bridging offrono l'opportunità di utilizzare il Supervisore quale "ponte" per il servizio di teleservice, consentendo di rendere trasparente da remoto una eventuale porta modem sul PC al PLC,

ovvero consentire l'accesso ad un PC remoto direttamente sul PLC utilizzando la porta di comunicazione del driver.

L'illustrazione seguente mostra il tipo di collegamento:



Naturalmente il driver, durante l'accesso in "bridging esterno", interromperà la propria comunicazione.

Enable

Abilita la possibilità di utilizzare la funzione di Bridging. Se posto a "true", il driver si metterà in ascolto sulla porta seriale specificata, quindi in caso di chiamata da remoto (es. per teleservice), il driver disconnetterà la comunicazione verso il PLC e metterà in connessione la porta modem con la porta collegata al PLC, in modo trasparente, fino al comando di chiusura (disconnessione), dove verrà ripristinata la comunicazione del Supervisore.

Per default, il servizio è disabilitato.

Port

Definisce il numero della porta seriale COM da utilizzare per il servizio di Bridging, sulla quale è collegato il modem.



Nota: occorre accertarsi che non sussistano conflitti in Windows nell'utilizzo delle porte. Ad esempio, installando le porte Com3 e Com4, occorre accertarsi che l'indirizzo assegnato e l'IRQ siano compatibili con la configurazione del PC. A tal scopo si consigliano schede seriali indirizzabili.

Baudrate

Definisce la velocità della comunicazione seriale (Baud Rate). Occorre impostare il valore di velocità di comunicazione corrispondente a quella del dispositivo con il quale si intende comunicare.

Byte Size

Definisce la quantità di byte richiesta dal protocollo di comunicazione interessato.

Parity

Definisce il tipo di parità richiesta dal protocollo di comunicazione interessato.

Stop Bits

Definisce il numero di Bit di Stop richiesto dal protocollo di comunicazione interessato.

Flow Control

Definisce il tipo di Controllo di Flusso dei dati per il tipo di comunicazione utilizzato. In particolare, permette di adattare la comunicazione secondo le necessità richieste a basso livello dalla porta seriale del dispositivo collegato. Di default il driver imposta "None" ovvero nessun controllo di flusso, tuttavia, in funzione del dispositivo, potrebbe essere necessario selezionare un tipo di Controllo Flusso (ad esempio in caso di segnalazione di errore con codice "1").

Le possibili selezioni sono:

None: Nessun Controllo di Flusso. Il controllo non è richiesto dal protocollo.

Hardware: Il Controllo di Flusso è gestito dai segnali elettrici della linea seriale (es. RTS, CTS, ecc.).

Xon/Xoff: Il controllo di flusso dei dati è di tipo Xon/Xoff.

Rx Queue

Definisce la quantità di byte di dati da gestire per la porta seriale come buffer (coda) di ricezione dei valori. Se non diversamente specificato, viene assunto il valore di default.

Per gli utenti esperti, è possibile modificare questo valore adattandolo alle necessità del tipo di sistema utilizzato.

Tx Queue

Definisce la quantità di byte di dati da gestire per la porta seriale come buffer (coda) di trasmissione dei valori. Se non diversamente specificato, viene assunto il valore di default.

Per gli utenti esperti, è possibile modificare questo valore adattandolo alle necessità del tipo di sistema utilizzato.

CD Timeout

Attualmente non utilizzato. Riservato per implementazioni future.

CTS Timeout

Definisce il valore di tempo, espresso in millisecondi, oltre il quale il driver deve notificare il timeout di comunicazione per il parametro seriale CTS. Definisce il tempo entro il quale deve essere completata la singola operazione di scrittura, a basso livello (API di Windows) sulla porta seriale.

DSR Timeout

Definisce il valore di tempo, espresso in millisecondi, oltre il quale il driver deve notificare il timeout di comunicazione per il parametro seriale DSR. Definisce l'intervallo di tempo massimo tra la ricezione di un carattere ed il successivo, inoltre concorre a definire il tempo massimo entro il quale deve essere completata la singola operazione di lettura, a basso livello (API di Windows) sulla porta seriale.

Display Dialog

Se abilitato (= true), consente al Supervisore di visualizzare una finestra di dialogo al momento della connessione in Bridging, consentendo all'utente locale di annullare la connessione da remoto.

Disconn.Delay

Definisce il tempo di ritardo, in millisecondi, dalla ricezione del segnale di disconnessione, per attivare effettivamente la chiusura della connessione di bridging.

Connection String

Definisce la stringa in ricezione dal modem che determina la richiesta di connessione da remoto. Al ricevimento di questa stringa, il driver attiva la richiesta di servizio di bridging.

Disconnection String

Definisce la stringa in ricezione dal modem che determina la richiesta di termine della connessione da remoto. Al ricevimento di questa stringa, il driver disattiva il servizio di bridging e ripristina la normale comunicazione del driver.

Init String

Definisce la stringa di inizializzazione del modem (es. AT&FS0=1).

OK String

Definisce la stringa di OK del modem (es. OK).

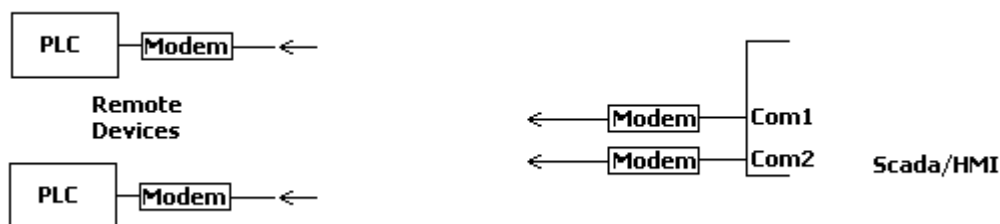
1.20. TAPI Settings

In questo gruppo di impostazioni occorre definire, per la stazione selezionata, le configurazioni inerenti la funzione TAPI del driver.

Le funzioni TAPI consentono al driver di connettersi alla stazione remota via modem, in modo automatico e trasparente.

Qualora si disponga di stazioni PLC o dispositivi remoti, connessi al supervisore tramite modem, la funzione TAPI consente di gestire il protocollo di comunicazione dopo avere stabilito la connessione con i modem.

La figura seguente mostra il tipo di collegamenti:



Il modem del PLC deve essere configurato in ricezione e pronto a ricevere le chiamate del Supervisore. Il Supervisore effettuerà la chiamata al modem interessato, per le stazioni del driver interessate, automaticamente nel momento in cui le variabili associate andranno "in uso".

Utilizzando le impostazioni TAPI della stazione per gestire la comunicazione via modem, la seriale viene gestita dal driver del modem. Le impostazioni con le quali viene aperta la seriale sono quindi quelle stabilite nelle proprietà avanzate del driver del modem, che possono essere diverse da quelle impostate nella stazione del driver Movicon ("Serial Port Settings").



Ricordiamo che le risorse "sempre in esecuzione" (quali DataLoggers, Allarmi, Schedulatori, Logica Generale) manterranno sempre "in uso" le variabili.



Le funzioni RAS sono disponibili solamente se la proprietà 'Enable' è abilitata.

Utilizzando la connessione TAPI si possono avere diversi comportamenti in caso di chiamata fallita che dipendono dalle impostazioni fatte sulla stazione. Le combinazioni sono le seguenti:

1. Utilizzo della variabile di stato della stazione ("State/Command Variable") e proprietà "Dial Only on Command" abilitata.
In caso di chiamata fallita il comportamento sarà il seguente:
 - Il "Bit 1" della variabile di stato viene impostato a "True" dopo che sono stati eseguiti i tentativi programmati di chiamata senza successo
 - Appare un errore nel log storico con la descrizione del motivo per cui non è stato possibile portare a termine la chiamata alla stazione
 - Il "Bit 1" della variabile di stato viene impostato automaticamente a "False" dal driver quando viene comandata un'altra connessione ("Bit 5")
2. Utilizzo della variabile di stato della stazione ("State/Command Variable") e proprietà "Dial Only on Command" disabilitata.
In caso di chiamata fallita il comportamento sarà il seguente:
 - Il modem fa partire la chiamata quando le variabili della stazione vanno in uso (ad esempio al caricamento di una pagina)
 - Il "Bit 1" della variabile di stato viene impostato a "True" dopo che sono stati eseguiti i tentativi programmati di chiamata senza successo
 - Appare un errore nel log storico con la descrizione del motivo per cui non è stato possibile portare a termine la chiamata alla stazione
 - Se il "Bit 1" viene impostato a "False" (la stazione viene riattivata), allora il driver riprova a chiamare la stazione se ci sono ancora incarichi attivi verso quella stazione
3. Non viene utilizzata la variabile di stato della stazione ("State/Command Variable")
In caso di chiamata fallita il comportamento sarà il seguente:
 - Il driver tenta di connettersi alla stazione fin tanto che ci sono incarichi attivi verso quella stazione. I tentativi successivi immediati sono quelli specificati dal parametro "Retries", poi c'è un'attesa pari al valore inserito nel parametro "Retry Hold Time (sec)" fra un gruppo di tentativi e il successivo

Phone Number

Definisce il numero telefonico da chiamare, corrispondente al numero telefonico al quale è collegato il dispositivo remoto.

Dalla casella base 250 in poi, è possibile anche inserire in questo campo una variabile di tipo stringa del RealTimeDB del Supervisore. In questo modo sarà possibile cambiare il numero di telefono da chiamare durante la fase di runtime impostandolo semplicemente sulla variabile.

Retries

Definisce il numero di tentativi di chiamata consecutivi (senza attesa) in caso di fallimento nello stabilire la connessione con la stazione remota. Se si imposta il valore "0" viene fatto un solo tentativo iniziale di chiamata, dopodiché si attende il tempo impostato nel parametro "Retry Hold Time (sec)" e di nuovo viene fatto un singolo tentativo di chiamata.

Disconnect After

Definisce il tempo (espresso in secondi) di inattività, prima di eseguire la chiusura della connessione. La connessione avviene non appena le variabili interessate vanno "in uso" nel progetto. Se le variabili interessate non sono più in uso, dopo il tempo impostato il Supervisore chiude la comunicazione.

Retry Hold Time (sec)

Definisce il tempo di attesa (espresso in secondi) tra un gruppo di tentativi di chiamata e il successivo gruppo. Il numero di chiamate di ogni gruppo è il valore del parametro "Retries" + 1. Non vi è un limite sul numero massimo di tentativi di chiamata da fare. Il driver continuerà a provare fino a che la stazione continua a rimanere attiva e le variabili ad essa associate continuano a rimanere in uso.

Enable

Abilita, se posto a "true", la funzione di telecontrollo TAPI e le chiamate via modem al dispositivo remoto.

Prompt before connect

Se abilitato (true), il sistema visualizzerà una finestra di dialogo chiedendo conferma per effettuare la chiamata ed attivare la connessione remota, ogni volta che il sistema dovrà eseguire le connessioni via modem.

Show Dlg

Se abilitato (true), il sistema visualizzerà una finestra di dialogo informativa per informare l'utente della connessione in corso e del relativo stato.

Dial only on command (gestita solamente per i driver che utilizzano una classe base dalla build 250 in avanti)

Se si abilita questa proprietà, la connessione modem verrà attivata solo su comando utilizzando l'apposito Bit della variabile di Stato/Comando (bit 5) e verrà terminata utilizzando l'apposito Bit della stessa variabile (bit 6). Per maggiori dettagli vedere la proprietà "State/Command Variable". Questa opzione è utile quando si deve comunicare con più stazioni remote utilizzando un unico modem per le chiamate. Il progettista può configurare un driver per ogni stazione remota, ognuno con un diverso numero di telefono e una diversa variabile di Stato/Comando. A questo punto l'operatore potrà collegarsi ad una stazione remota per volta utilizzando la specifica variabile di Stato/Comando. Una volta terminato lo scambio di dati con l'attuale stazione l'operatore potrà interrompere la comunicazione e connettersi ad un'altra stazione.

1.21. TCP/IP Settings

In questo gruppo di impostazioni occorre definire, per la stazione selezionata, le configurazioni inerenti i parametri di accesso Ethernet TCP-IP.

Server Address

Permette di specificare l'indirizzo IP del Server o del dispositivo di rete al quale ci si deve connettere. Esempio: 192.168.0.1; localhost; Server1

Server Port

Permette di specificare la porta TCP del Server o del dispositivo al quale ci si deve connettere. Questo valore completa l'indirizzo IP del dispositivo. Ad esempio, per Modbus TCP-IP la porta è sempre la 502 (stabilita dal protocollo), mentre per altri dispositivi occorre riferirsi alle specifiche del dispositivo.

Backup Server Address

Si tratta dell'indirizzo di un Server di Backup. Se viene impostato questo indirizzo, quando il driver non riesce a comunicare con il Server "primario", prova a connettersi al Server di Backup. In sostanza si tratta di una ridondanza gestita a livello di driver. Se si interrompe anche la comunicazione con il Server di Backup, il driver tenta nuovamente di connettersi al Server primario, e così via.

Il campo può contenere il nome di un Server o una lista di Server separati dal carattere ";". La lista di Server consente di avere più di un Backup Server che il driver gestirà in modo ciclico, cioè quando un Server va in errore, viene usato quello immediatamente successivo alla lista di Backup fino a quando la lista non è esaurita e si riparte dal Server principale.

Switch Server Timeout

Si tratta del tempo, in millisecondi, che passa tra il verificarsi di un errore di comunicazione con un Server e il tentativo di connettersi all'altro Server.

Local Bound Address

Permette di specificare l'indirizzo IP locale, ovvero quello della scheda ethernet del PC che si intende utilizzare per comunicare. Normalmente non occorre specificare nulla, ma potrebbe essere necessario in caso di presenza di più di una scheda ethernet installata sul PC. Lasciando il campo vuoto, il sistema assume l'indirizzo quello di default del sistema operativo.

Local Bound Port

Permette di specificare l'indirizzo della porta TCP locale, riferito alla scheda ethernet del PC che si intende utilizzare per comunicare. Normalmente non occorre specificare nulla, ma potrebbe essere un parametro richiesto dal protocollo o dal dispositivo utilizzato.
Lasciando il campo = 0 la scelta della porta viene affidata al sistema operativo.

Use Ping

Abilitando questa opzione il driver utilizzerà un messaggio di "ping" del protocollo ICMP per testare l'esistenza di un Server di Backup prima di tentare la connessione verso quel Server. Questo consente di velocizzare il collegamento al Server di Backup in caso di una lista di Server molto lunga. L'attesa per la risposta sul messaggio di "ping" corrisponde al valore impostato nel campo "Timeouts\Rx Timeout" della stazione.

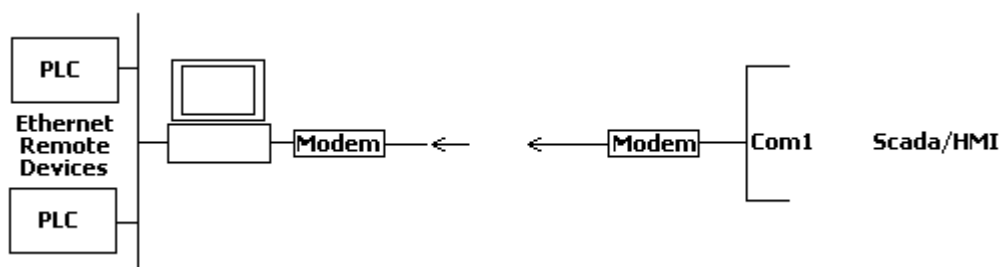
1.22. RAS Settings

In questo gruppo di impostazioni occorre definire, per la stazione selezionata, le configurazioni inerenti i parametri di accesso a dispositivi remoti tramite le funzioni RAS (Remote Access Service) del sistema operativo.

Le funzioni RAS del driver consentono di connettersi ad un network remoto via modem, in modo automatico e trasparente.

Qualora si disponga di dispositivi di rete connessi ad un PC remoto, la funzione RAS consente di gestire il protocollo di comunicazione di rete dopo avere stabilito la connessione con i modem tramite l'accesso ad un Server RAS, ovvero un PC di accesso alla rete raggiungibile via modem. Dopo la connessione RAS, il driver potrà accedere alla rete.

La figura seguente mostra il tipo di collegamenti:



Il modem del Server RAS deve essere configurato in ricezione e pronto a ricevere le chiamate del Supervisore. Il Supervisore effettuerà la chiamata al modem interessato, per le stazioni del driver interessate, automaticamente nel momento in cui le variabili associate andranno "in uso".



Ricordiamo che le risorse "sempre in esecuzione" (quali DataLoggers, Allarmi, Schedulatori, Logica Generale) manterranno sempre "in uso" le variabili.



Le funzioni RAS sono disponibili solamente se la proprietà 'Enable' è abilitata.

Dial-up

Permette di specificare una tra le Connessioni RAS definite nel sistema operativo. Se lasciato vuoto, il driver richiederà di specificare manualmente, al momento della connessione, il numero di telefono, username e password della stazione alla quale connettersi.

La connessione verrà richiesta non appena una tra le variabili associate andrà "in uso" nel progetto in esecuzione.

Phone Number

Definisce il numero di telefono della stazione remota alla quale connettersi.

Dalla casella base 250 in poi, è possibile anche inserire in questo campo una variabile di tipo stringa del RealTimeDB del Supervisore. In questo modo sarà possibile cambiare il numero di telefono da chiamare durante la fase di runtime impostandolo semplicemente sulla variabile.

User Name

Specifica il nome utente (username) per l'accesso alla stazione remota.

Password

Specifica la password per l'accesso alla stazione remota.

Retries

Definisce il numero massimo di tentativi di connessione nel caso in cui questa fallisca. Al termine dei tentativi, se la comunicazione non fosse stabilita, verrà segnalato l'errore.

Disconnect After

Definisce il tempo (espresso in secondi) di inattività, prima di eseguire la chiusura della connessione. La connessione avviene non appena le variabili interessate vanno "in uso" nel progetto. Se le variabili interessate non sono più in uso, dopo il tempo impostato il Supervisore chiude la comunicazione.

Retry Hold Time (sec)

Definisce il tempo (espresso in secondi) prima di ritentare la connessione.

Enable

Abilita, se posto a "true", la funzione di telecontrollo TAPI e le chiamate via modem al dispositivo remoto.

Prompt before connect

Se abilitato (true), il sistema visualizzerà una finestra di dialogo chiedendo conferma per effettuare la chiamata ed attivare la connessione remota, ogni volta che il sistema dovrà eseguire le connessioni via modem.

Show Dlg

Se abilitato (true), il sistema visualizzerà una finestra di dialogo informativa per informare l'utente della connessione in corso e del relativo stato.

Dial only on command (gestita solamente per i driver che utilizzano una classe base dalla build 250 in avanti)

Se si abilita questa proprietà, la connessione modem verrà attivata solo su comando utilizzando l'apposito Bit della variabile di Stato/Comando (bit 5) e verrà terminata utilizzando l'apposito Bit della stessa variabile (bit 6). Per maggiori dettagli vedi variabile di Stato/Comando. Questa opzione è utile quando si deve comunicare con più stazioni remote utilizzando un unico modem per le chiamate. Il progettista può configurare un driver per ogni stazione remota, ognuno con un diverso numero di telefono e una diversa variabile di Stato/Comando. A questo punto l'operatore potrà collegarsi ad una stazione remota per volta utilizzando la specifica variabile di Stato/Comando. Una volta terminato lo scambio di dati con l'attuale stazione l'operatore potrà interrompere la comunicazione e connettersi ad un'altra stazione.

1.23. Configurazioni speciali (TAPI e RAS)

Le tecniche RAS e TAPI sono utilizzate dal driver per gestire la supervisione di progetti che necessitano di un controllo remoto. Ci sono alcuni aspetti che vanno valutati con cura.

Le più comuni architetture di controllo remoto sono:

1. Una stazione remota e un progetto di supervisione che dialoga con la stazione mediante un modem
2. Varie stazioni remote, collocate in zone diverse, e un progetto di supervisione che dialogano con le stazioni mediante un solo modem
3. Varie stazioni remote, che si trovano sulla stessa rete, e un progetto di supervisione che dialoga con le stazioni mediante un solo modem

Le possibili soluzioni per una comunicazione ottimale sono:

1. E' necessario definire nel driver una stazione e in questo caso la gestione delle chiamate potrà essere eseguita sia in automatico (il driver avvia una chiamata quando il tag va in uso) che in manuale, utilizzando la variabile di Stato/Comando
2. E' necessario definire due o più stazioni, ognuna con la sua variabile di Stato/Comando. In questo caso le chiamate dovranno essere gestite mediante la variabile di Stato/Comando per evitare chiamate simultanee
3. E' necessario definire due o più stazioni, ognuna con la sua variabile di Stato/Comando, ma solo una stazione può essere utilizzata per la chiamata RAS o TAPI. Una volta istituita la connessione per la stazione "principale", la comunicazione con le restanti stazioni saranno attivate utilizzando il bit 1 delle variabili di Stato/Comando associate

Per ulteriori dettagli sulle impostazioni delle variabili di Stato/Comando vedere Generale (Station)

1.24. Half Duplex

In questo gruppo di impostazioni occorre definire, per la stazione selezionata, le configurazioni inerenti alla commutazione di direzione della comunicazione nel caso di una linea seriale half duplex.



Generalmente, queste impostazioni sono riservate agli utenti esperti, pertanto si consiglia di mantenere le impostazioni di default (impostazioni per linee seriali full-duplex).

Signal

Seleziona il segnale della RS232 utilizzato per controllare la direzione della trasmissione. Possibili valori: **None** (valore di default), **RTS** e **DTR**.

Delay (ms.) Before First Char

Definisce il tempo atteso dopo il set a ON del segnale di controllo prima di dare inizio alla trasmissione. Valore di default = 0.

Delay (ms.) After Last Char

Definisce il tempo atteso dopo la trasmissione dell'ultimo carattere prima di mettere a OFF il segnale di controllo. Valore di default = 0.

I tempi sono espressi in millisecondi.

1.25. Impostazioni Task

1.26. Tasks

In questa sezione occorre inserire ed impostare gli "**incarichi di comunicazione (Task)**", qualora si intenda utilizzare questa tecnica di comunicazione.



Ricordiamo infatti che il Supervisore offre la possibilità di impostare la comunicazione del driver sia tramite il concetto di "task" che tramite il concetto degli **incarichi dinamici**. Gli incarichi dinamici sono creati automaticamente dal driver all'avvio del progetto, in funzione dei link agli indirizzi del dispositivo definiti nelle proprietà "Indirizzo Fisico I/O" di ogni singolo tag).

Gli incarichi di comunicazione permettono di assegnare al driver degli incarichi "**statici**", che saranno eseguiti in polling verso i dispositivi previsti.



Utilizzando la tecnica di comunicazione ad "incarichi", occorrerà definire in modo statico le associazioni di dati tra gli indirizzi del dispositivo e le variabili del Supervisore.

Add

Il pulsante "Add" permette di inserire un nuovo "Incarico" (Task) per il driver di comunicazione. Inserendo un nuovo incarico, verrà automaticamente visualizzata la relativa finestra di impostazione dei parametri di comunicazione richiesti.

Gli incarichi inseriti potranno in seguito essere modificati o rimossi utilizzando i pulsanti sotto indicati.

Edit

Il pulsante "Edit" permette di modificare i parametri di un "Incarico" precedentemente inserito. Occorrerà pertanto selezionare l'incarico desiderato e quindi utilizzare il pulsante "Edit" oppure il doppio clic.

Remove

Il pulsante "Remove" permette di eliminare un "Incarico" precedentemente inserito. Occorrerà pertanto selezionare l'incarico desiderato e quindi utilizzare il pulsante "Remove".

1.27. Proprietà Generali Task Statici

In questa scheda di impostazione occorre definire, per l'incarico selezionato, le impostazioni inerenti il gruppo di proprietà "Generali".

Station

Nome che identifica la stazione corrispondente al dispositivo con il quale si intende comunicare. Le stazioni devono essere definite (almeno una) nella scheda di impostazioni relativa.
Se si definiscono più stazioni, occorre selezionare in questa casella la stazione desiderata per la quale l'incarico sarà eseguito.

Task Name

Nome che identifica il nome dell'incarico in fase di impostazione. E' possibile definire una qualsiasi stringa identificativa dell'incarico (Task).
Ogni Task deve avere un nome univoco nella lista dei task.

Variables

Permette di associare la variabile (o le variabili) del Supervisore sulle quali l'incarico di comunicazione andrà ad agire. Dal pulsante di selezione posto sul bordo destro, è possibile selezionare una qualsiasi variabile (Tag) precedentemente inserita nella risorsa RealTime Database del progetto o crearne di nuove. La variabile inserita verrà automaticamente aggiunta al Real Time DB. Le variabili nel campo variabili **devono essere separate dal carattere ';' e dovrebbero risultare consecutive sul dispositivo a partire dal valore inserito nella proprietà 'Address'**.



Nota importante: è fortemente raccomandato l'utilizzo di una lista di variabili dello stesso tipo per evitare errori non prevedibili. Per strutture di dati complesse sui dispositivi, i cui membri possono essere di diverso tipo, è possibile utilizzare strutture dati definite nel realtime db e linkare direttamente la variabile di tipo struttura al dispositivo con un task statico.

Il driver calcola automaticamente il numero di byte da leggere/scrivere a partire dall'indirizzo specificato per il dispositivo aggiungendo la dimensione di ogni variabile presente nella lista, in base alla definizione delle stesse nel RealTimeDB del progetto.

Il driver attua poi controlli aggiuntivi in funzione dell'indirizzo specificato.

Indirizzamento a Bit

Quando è necessario accedere a indirizzi a Bit sul dispositivo, tutte le variabili definite nella proprietà 'Variables' devono essere dichiarate nel progetto di supervisione di tipo bit, altrimenti sarà visualizzato un messaggio di errore. Il numero di Bit che deve essere scambiato con il dispositivo è uguale al numero di variabili dichiarate nella lista 'Variables'.
L'indirizzo di partenza è quello specificato nella proprietà 'Start Address'.

Indirizzamento a Byte, word o double word

Nessuna dei tag inseriti nella proprietà 'Variables' deve essere dichiarata di tipo Bit nel RealTimeDB del progetto di supervisione, altrimenti sarà visualizzato un errore.

Conditional Variable

Permette di associare una variabile del progetto il cui stato determinerà l'esecuzione condizionata dell'incarico di comunicazione.
Dal pulsante di selezione posto sul bordo destro, è possibile selezionare una qualsiasi variabile (Tag) precedentemente inserita nella risorsa RealTime Database del progetto.
La variabile (di qualunque tipo) pertanto condizionerà l'esecuzione dell'incarico: se posta a valore "diverso da zero" (> 0) l'incarico di comunicazione verrà eseguito dal driver.



Al termine "certo" dell'esecuzione dell'incarico, il driver metterà automaticamente a zero il valore della variabile associata. Pertanto, occorrerà considerare questo nella gestione della variabile da parte di eventuali logiche.

Polling Time

Questo parametro, espresso in millisecondi, determina il tempo minimo di polling dell'esecuzione del singolo incarico, per l'aggiornamento dei dati **quando le variabili sono in uso**.
Il valore di default del Polling Time, inserito alla creazione del task, è determinato dal valore stabilito nelle proprietà Generali del driver ("Polling Time"), ma può essere modificato a piacimento per ogni singolo incarico. Il valore uguale a zero significa che i dati verranno aggiornati con la massima velocità possibile.

Può essere impostato un valore più alto, ad esempio, qualora i dati non richiedano tempi di aggiornamento rapidi.

Unused Polling Time

Questo parametro, espresso in millisecondi, permette di forzare un aggiornamento dei dati del singolo incarico anche quando le variabili **non sono in uso, stabilendo comunque un tempo di polling**. Il valore di default dell'Unused Polling Time, inserito alla creazione del task, è determinato dal valore stabilito nelle proprietà Generali del driver ("Unused Polling Time"), ma può essere modificato a piacimento per ogni singolo incarico.

Impostando questo valore a 10000 (ovvero 10 secondi) significa che il task verrà eseguito con un tempo minimo di 10 secondi anche quando le sue variabili non sono in uso.



Impostando questo parametro = 0, se le variabili non sono in uso gli incarichi non saranno mai eseguiti. Quando questo valore è diverso da zero tutte le variabili non 'in uso' il cui incarico è di tipo input o input/output saranno aggiornate in lettura con questa frequenza. Questo meccanismo potrebbe rallentare la comunicazione in caso di un numero elevato di tag.

Address offset variable (disponibile solo per i driver che utilizzano le Classi Base versione 250 o successive)

In questo campo è possibile specificare il nome di una variabile di tipo intero definita nel RealTimeDB. Il valore della variabile potrà essere modificato a runtime e sarà utilizzato come offset positivo o negativo, in funzione del valore assunto, rispetto all'indirizzo iniziale impostato nel task. In questo modo sarà possibile leggere in diversi punti la memoria del dispositivo semplicemente modificando tale valore a runtime.

L'unità di offset dipende dal driver e dal tipo di indirizzamento:

- Per tipi di indirizzamento a Bit un valore di offset pari a 1 corrisponde ad uno scostamento a Bit.
- Per tipi di indirizzamento a Byte, un valore di offset pari a 1 corrisponde a un Byte
- Per tipi di indirizzamento a Word, un valore di offset pari a 1 corrisponde ad una Word

Il valore inserito nella proprietà Address Offset può avere significati diversi a seconda del Driver che si sta utilizzando. Ad esempio utilizzando un driver Siemens se si deve leggere una variabile di tipo word, si dovrà utilizzare un offset +2 o multiplo di 2 in quanto nel Siemens l'indirizzamento è gestito a byte. Nel caso in cui invece si utilizzi un driver Modbus se si deve leggere una variabile di tipo word si dovrà inserire un offset +1 o multiplo di 1 in quanto nel Modbus l'indirizzamento è gestito a word.

Nota bene: anche modificando il valore di offset i task non verranno eseguiti immediatamente ma solo secondo le tempistiche opportune. Per avere la certezza che un incarico è stato eseguito è possibile utilizzare la variabile condizionale. Quindi ad esempio un incarico di Exeption Output verrà eseguito con il nuovo offset solamente quando il valore della variabile associata sarà modificato.

Un esempio per scrivere in diverse zone della memoria del dispositivo, potrebbe essere quello di utilizzare un incarico di Unconditional output con variabile condizionale e variabile di offset. Una volta modificato il valore di offset e il valore da scrivere nella variabile sarà sufficiente impostare a 1 la variabile condizionale. Quando la variabile condizionale tornerà a zero, si avrà la certezza che l'incarico è stato eseguito e si potrà passare all'indirizzamento successivo.

Swap Byte

Questa selezione permette di invertire, per i dati di tipo "word", i byte tra loro. Si otterrà che i dati collegati tra Supervisore e dispositivo vedranno il byte "alto" invertito con il byte "basso" e viceversa, per ogni dato di tipo word dell'incarico.

Swap Word

Questa selezione permette di invertire, per i dati di tipo "double word", le word tra loro. Si otterrà che i dati collegati tra Supervisore e dispositivo vedranno la word "alta" invertita con la word "bassa" e viceversa, per ogni dato di tipo double dell'incarico.

Type

In questa selezione occorre impostare il tipo di esecuzione che si intende assegnare all'incarico. Le possibilità offerte sono:

Input

Questa selezione definisce che il tipo di incarico sarà di "Sola Lettura". In tal modo, se le variabili sono in uso, il driver eseguirà il polling in lettura sugli indirizzi associati per il dispositivo, e li trasferirà alle variabili del progetto desiderate.

Input/Output

Questa selezione definisce che il tipo di incarico sarà di "Lettura-Scrittura". In tal modo, se le variabili sono in uso, il driver eseguirà il polling in lettura sugli indirizzi

associati per il dispositivo, e li trasferirà alle variabili del progetto desiderate. Nel caso in cui una variabile cambi il suo valore dal Supervisore, il driver eseguirà la scrittura dei dati verso il dispositivo, quindi ritornerà in lettura.

Exception Output

Questa selezione definisce che il tipo di incarico sarà di "Sola Scrittura", gestendo la scrittura su eccezione, ovvero solo se un dato cambia.

Unconditional Output

Questa selezione definisce che il tipo di incarico sarà di "Sola Scrittura", gestendo la scrittura continuamente, indipendentemente dal fatto che i dati cambino.

Write Outputs at Startup

Questa proprietà ha senso solo per un incarico di Input/Output o di "Exception Output". Se impostata a "True" l'incarico viene eseguito in output all'avvio del progetto.

Device Data

I parametri di questo gruppo determinano l'indirizzamento alle aree di memoria dello specifico dispositivo.

Riferirsi pertanto alla documentazione specifica del protocollo.

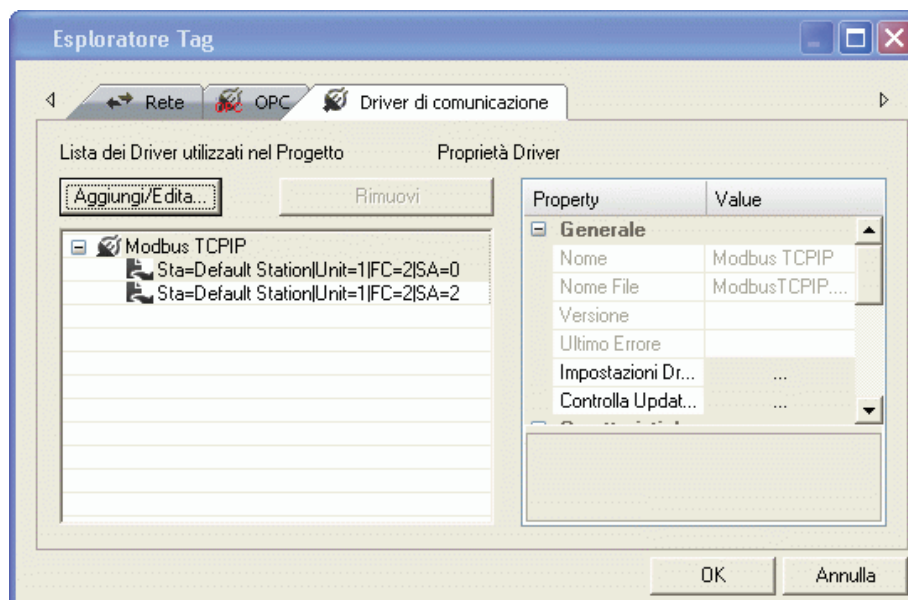
1.28. Task Dinamici

I task dinamici vengono creati automaticamente dal driver allo startup del progetto di supervisione, basandosi sull'indirizzo dinamico inserito nell'apposita proprietà "Indirizzo Fisico I/O" di ogni singola variabile.

Le impostazioni per i task dinamici non si configurano mediante il comando "Impostazioni Driver", ma vengono specificate per ogni singolo tag Tramite la finestra "Esploratore Tag".

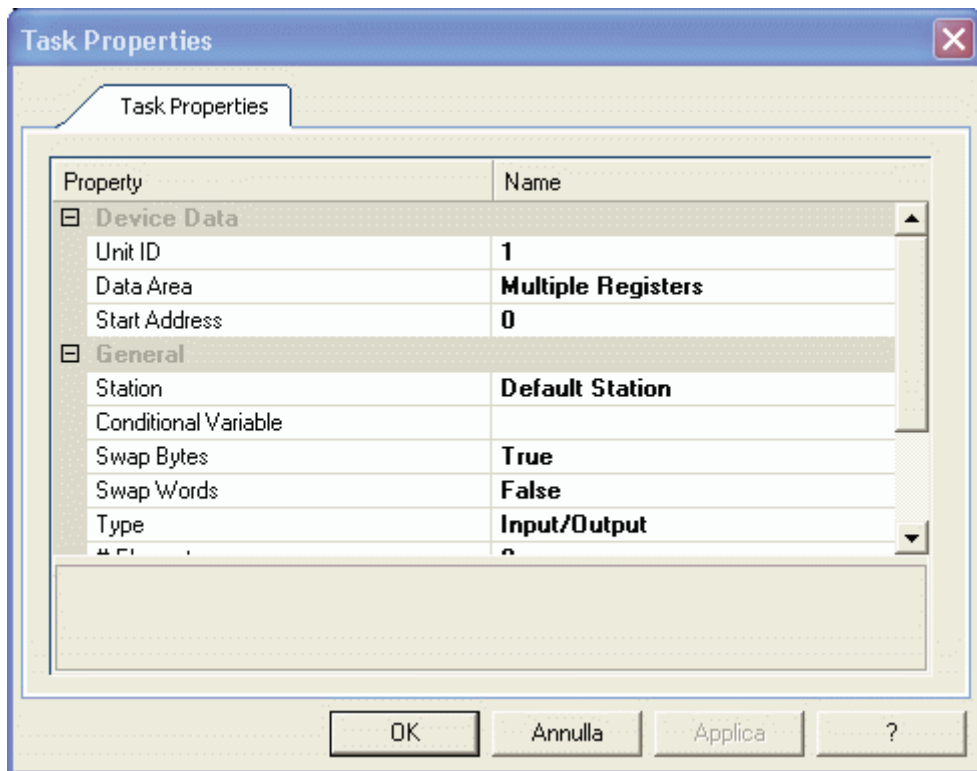
Inserimento "Indirizzo Fisico I/O" di un Tag

Per impostare le proprietà dinamiche delle variabili selezionare la lista delle variabili dal RealTimeDB nella finestra Esploratore Progetto, selezionare successivamente la singola variabile di interesse e visualizzare la Finestra delle Proprietà. Tramite il pulsante di selezione posto sul bordo destro della proprietà "Indirizzo Fisico I/O" aprire la finestra "Esploratore Tag".



Finestra Esploratore Tag.

Nella finestra Esploratore Tag selezionare la Tab "Driver di comunicazione". In questa finestra viene riportata la lista dei driver inseriti nel progetto e per ogni driver vengono elencati i link dinamici già definiti. Inoltre è presente un pulsante "Aggiungi/Edita..." tramite il quale è possibile aprire la finestra "Task Properties" per editare i link dinamici, e un pulsante "Rimuovi" che ha lo scopo di eliminare il link dinamico selezionato.



Finestra Task Properties.

Una volta aperta la finestra "Esploratore Tag" l'inserimento/modifica di un link dinamico può essere fatta nei seguenti modi:

Inserimento di un link dinamico nuovo

1. dalla proprietà "Indirizzo Fisico I/O" della variabile aprire la finestra "Esploratore Tag"
2. selezionare il nome del driver che si vuole utilizzare
3. cliccare sul pulsante "Aggiungi/Edita..." per aprire la finestra "Task Properties"
4. se la variabile non aveva nessun link dinamico assegnato la finestra "Task Properties" verrà aperta con i valori di default. Se invece la variabile aveva già un suo link dinamico associato la finestra "Task Properties" verrà aperta con i valori di quel link. Impostare quindi i parametri desiderati
5. chiudere la finestra "Task Properties" con "OK". In questo modo verrà composto l'indirizzo dinamico con i valori impostati, verrà aggiunto il nuovo link nella lista degli indirizzi dinamici e gli verrà dato il focus
6. chiudere la finestra "Esploratore Tag" con OK per inserire il link dinamico nella proprietà "Indirizzo Fisico I/O" della variabile

Modifica di un link dinamico esistente

1. dalla proprietà "Indirizzo Fisico I/O" della variabile aprire la finestra "Esploratore Tag"
2. selezionare un link dinamico dalla lista del driver desiderato e cliccare sul pulsante "Aggiungi/Edita..." per aprire la finestra "Task Properties"
3. se la variabile non aveva nessun link dinamico assegnato la finestra "Task Properties" verrà aperta con i valori del link dinamico selezionato. Se invece la variabile aveva già un suo link dinamico associato la finestra "Task Properties" verrà aperta con i valori del link della variabile. Impostare quindi i parametri desiderati
4. chiudere la finestra "Task Properties" con "OK". In questo modo verrà composto l'indirizzo dinamico con i valori impostati, verrà aggiornato il link nella lista degli indirizzi dinamici e gli verrà dato il focus
5. chiudere la finestra "Esploratore Tag" con OK per inserire il link dinamico nella proprietà "Indirizzo Fisico I/O" della variabile

Selezione di un link dinamico esistente

1. dalla proprietà "Indirizzo Fisico I/O" della variabile aprire la finestra "Esploratore Tag"
2. selezionare un link dinamico dalla lista del driver desiderato
3. chiudere la finestra "Esploratore Tag" con OK per inserire il link dinamico nella proprietà "Indirizzo Fisico I/O" della variabile

Si tenga conto che la lista dei link dinamici di un driver visualizzati nella finestra Esploratore Tag ha soltanto una funzione di promemoria o di selezione per il programmatore. Infatti una volta che il link

è stato assegnato alla variabile questo potrebbe anche essere rimosso dalla lista senza provocare malfunzionamenti al progetto. La lista dei link dinamici viene salvata nel file ".dyndrv" del driver presente nella cartella "RESOURCES" del progetto. Eliminando ad esempio questo file non si avrà più a disposizione la lista dei link dinamici inseriti fino a quel momento nella finestra "Esploratore Tag", ma nella proprietà "Indirizzo Fisico I/O" delle variabili rimarrà salvato l'indirizzo dinamico che era stato assegnato in precedenza.

Proprietà dei Task Dinamici

Tramite la finestra "Task Properties" è possibile definire le impostazioni del link dinamico della variabile. Tramite questa finestra è possibile assegnare le proprietà generali del task, gruppo "General" e le proprietà specifiche per l'indirizzamento del dispositivo le quali cambiano a seconda del dispositivo. Di seguito viene riportata la descrizione delle proprietà generali del task, per quelle specifiche sull'indirizzamento del dispositivo si rimanda all'help specifico del driver.

Station

Nome che identifica la stazione corrispondente al dispositivo con il quale si intende comunicare. Le stazioni devono essere definite (almeno una) nella scheda di impostazioni relativa. Se si definiscono più stazioni, occorre selezionare in questa casella la stazione desiderata per la quale l'incarico sarà eseguito.

Conditional Variable

Permette di associare una variabile del progetto il cui stato determinerà l'esecuzione condizionata dell'incarico di comunicazione. Dal pulsante di selezione posto sul bordo destro, è possibile selezionare una qualsiasi variabile (Tag) precedentemente inserita nella risorsa RealTime Database del progetto o crearne di nuove. La variabile (di qualunque tipo) pertanto condiziona l'esecuzione dell'incarico: se posta a valore "diverso da zero" (> 0) l'incarico di comunicazione verrà eseguito dal driver. Una volta riportata al valore zero si avrà la certezza dell'esecuzione del task.

Swap Byte

Questa selezione permette di invertire, per i dati di tipo "word", i byte tra loro. Si otterrà che i dati collegati tra Supervisore e dispositivo vedranno il byte "alto" invertito con il byte "basso" e viceversa, per ogni dato di tipo word dell'incarico. Questa proprietà è presente solo per alcuni driver.

Swap Word

Questa selezione permette di invertire, per i dati di tipo "double word", le word tra loro. Si otterrà che i dati collegati tra Supervisore e dispositivo vedranno la word "alta" invertita con la word "bassa" e viceversa, per ogni dato di tipo double dell'incarico. Questa proprietà è presente solo per alcuni driver.

Type

In questa selezione occorre impostare il tipo di esecuzione che si intende assegnare all'incarico. Le possibilità offerte sono:

Input	Questa selezione definisce che il tipo di incarico sarà di "Sola Lettura". In tal modo, se le variabili sono in uso, il driver eseguirà il polling in lettura sugli indirizzi associati per il dispositivo, e li trasferirà alle variabili del progetto desiderate.
Input/Output	Questa selezione definisce che il tipo di incarico sarà di "Lettura-Scrittura". In tal modo, se le variabili sono in uso, il driver eseguirà il polling in lettura sugli indirizzi associati per il dispositivo, e li trasferirà alle variabili del progetto desiderate. Nel caso in cui una variabile cambi il suo valore dal Supervisore, il driver eseguirà la scrittura dei dati verso il dispositivo, quindi ritornerà in lettura.
Exception Output	Questa selezione definisce che il tipo di incarico sarà di "Sola Scrittura", gestendo la scrittura su eccezione, ovvero solo se un dato cambia.
Unconditional Output	Questa selezione definisce che il tipo di incarico sarà di "Sola Scrittura", gestendo la scrittura continuamente, indipendentemente dal fatto che i dati cambino.

#Elements

Specifica il numero di elementi da leggere/scrivere. I valori possibili sono 0 o 1. Se lasciato a zero, il driver calcolerà automaticamente il numero di byte in funzione del tipo di variabile definita nel RealTimeDB del progetto di supervisione. Ad esempio se la variabile associata è di tipo Word verranno presi in considerazione per l'incarico 2 byte a partire dall'indirizzo definito.

Se viceversa il numero di elementi è pari a 1, indipendentemente dal tipo definito per la variabile associata, il driver calcolerà automaticamente il numero di byte in funzione del tipo di indirizzamento. Quindi, per esempio, se l'area indirizzata sul dispositivo è a Word, allora verranno letti/scritti 2 byte anche se la variabile di appoggio è di tipo word.

Write Outputs at Startup

Questa proprietà ha senso solo per un incarico di Input/Output o di "Exception Output". Se impostata a "True" l'incarico viene eseguito in output all'avvio del progetto.

1.29. Import Database Dispositivo

1.30. Importazione Database Dispositivo

Grazie a questa importante funzionalità, è possibile accedere direttamente al database del PLC o del dispositivo (o alla equivalente fonte di dati) per ottenere l'importazione nel progetto Supervisore delle variabili (Tag) desiderate.

Al termine dell'operazione, verrà riempito automaticamente il RealTime Database del progetto, inserendo tutte le variabili importate, le quali verranno definite come Tag Dinamici in area "Non Condivisa", con il tipo di variabile corrispondente e con l'indirizzo già assegnato per il dispositivo.

Attivando il comando verrà visualizzata la finestra per la selezione della "fonte dei dati", ovvero il database del PLC, il file simbolico oppure il file .CSV ottenuto dall'esportazione dei dati del PLC o del dispositivo.

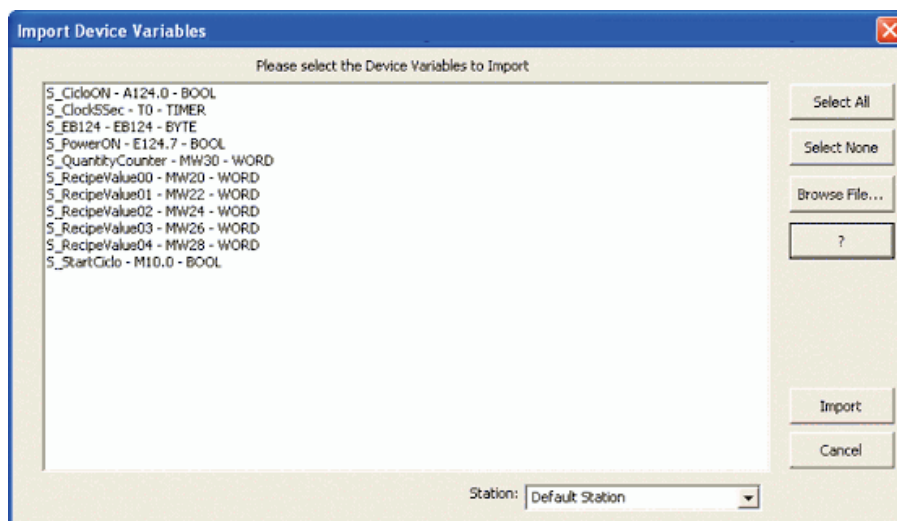
Occorrerà pertanto disporre del file, e quindi selezionarlo. A questo punto verrà quindi visualizzata la finestra illustrata sotto, rappresentante le variabili in esso contenute.



Attenzione: l'importazione dei dati dal PLC è supportata in tutti i driver per i dispositivi più noti. Verificare le modalità di accesso o i requisiti della "fonte dei dati", che possono variare a seconda del dispositivo.



Se durante l'importazione dei dati vengono rilevate delle variabili già presenti nel RealTimeDB del progetto, queste verranno automaticamente sovrascritte senza richiedere alcuna conferma. In particolare verranno sovrascritte le proprietà "Tipo di Dato" e "Indirizzo Fisico I/O", mentre la proprietà "Descrizione Tag" verrà lasciata invariata. La proprietà "Descrizione Tag" verrà quindi importata solo se la variabile non esiste nel RealTimeDB del progetto.



Quando si seleziona un numero elevato di variabili da importare è possibile che l'operazione di importazione impieghi del tempo ad essere completata. E' comunque possibile abortire l'operazione premendo il tasto "ESC".

L'importazione delle Strutture e degli Array tramite il Tool di importazione del Data Base del dispositivo può avvenire in due modi differenti:

Se dalla finestra di dialogo del tool viene selezionata la radice della Struttura ed eseguita l'importazione col tasto "Import" allora verrà creato nel progetto un prototipo definito coi membri della struttura e una variabile dello stesso tipo di quel prototipo. E' quindi possibile eseguire l'importazione di una selezione multipla di strutture. Inoltre la stessa funzione è realizzata anche tramite la sequenza di selezione del pulsante "Select All" e successivamente del pulsante "Import".

Se vengono selezionati invece uno o più membri di una o più Strutture e poi eseguita l'importazione, verranno create le variabili del tipo corrispondente a quello dei membri selezionati ma non i prototipi struttura. Lo stesso risultato lo si può ottenere cliccando sul pulsante "Expand All" per esplorare i membri delle Strutture, poi cliccando sul pulsante "Select All" per selezionare tutti i membri ed infine su quello "Import" per importare i singoli membri come variabili distinte.

Ovviamente se nell'elenco delle variabili del dispositivo sono presenti sia variabili singole oltre che a variabili Struttura, l'effetto del "Select All" è quello di selezionarle tutte.

Per quanto riguarda l'importazione di Array se viene selezionata e importata la radice dell'Array verrà creata nel progetto una variabile di tipo Array con proprietà "Tipo Elementi" impostata sul tipo dell'Array. La stessa funzione la si realizza anche tramite la sequenza di selezione del pulsante "Select All" e successivamente del pulsante "Import".

Se vengono selezionati invece uno o più elementi di una o più Array e poi eseguita l'importazione, verranno create le variabili del tipo corrispondente a quello degli elementi selezionati ma non le variabili di tipo Array. Lo stesso risultato lo si può ottenere cliccando sul pulsante "Expand All" per esplorare gli elementi delle Array, poi cliccando il pulsante "Select All" per selezionarli tutti ed infine su quello "Import" per importare i singoli elementi come variabili distinte.

Ovviamente se nell'elenco delle variabili del dispositivo sono presenti sia variabili singole oltre che variabili Array l'effetto del "Select All" è quello di selezionarle tutte.

Select All

Permette di selezionare tutte le variabili del file di importazione.

In caso di selezione parziale, occorre utilizzare la combinazione di tasti CTRL+Clic, oppure SHIFT+Clic.

Select None

Permette di deselectionare tutte le variabili del file di importazione.

In caso di deselection parziale, occorre utilizzare la combinazione di tasti CTRL+Clic, oppure SHIFT+Clic.

Browse File...

Permette di cambiare il file di origine, attivando la finestra standard per la selezione dei file.

? (help)

Attiva la visualizzazione della guida contenente le informazioni relative ai requisiti del formato di origine dei dati.

Import

Attiva l'importazione delle variabili dal file di origine (fonte dei dati del dispositivo) al progetto del Supervisore. Al termine dell'importazione, la risorsa Realtime Database del progetto risulterà popolata da tutte le variabili importate.



Poichè le "fonti dati" dipendono dal dispositivo e potrebbero cambiare, si consiglia sempre di verificare le proprietà delle variabili importate, verificando la corretta esecuzione del parsing automatico, l'assegnazione del tipo e dell'indirizzo del dispositivo al termine dell'importazione.

Cancel

questo pulsante annulla l'operazione di importazione.

Station:

Questa casella permette di selezionare la stazione del driver da assegnare alle variabili importate, nel caso in cui al driver siano state definite più stazioni.

1.31. API Interfacce Basic Script

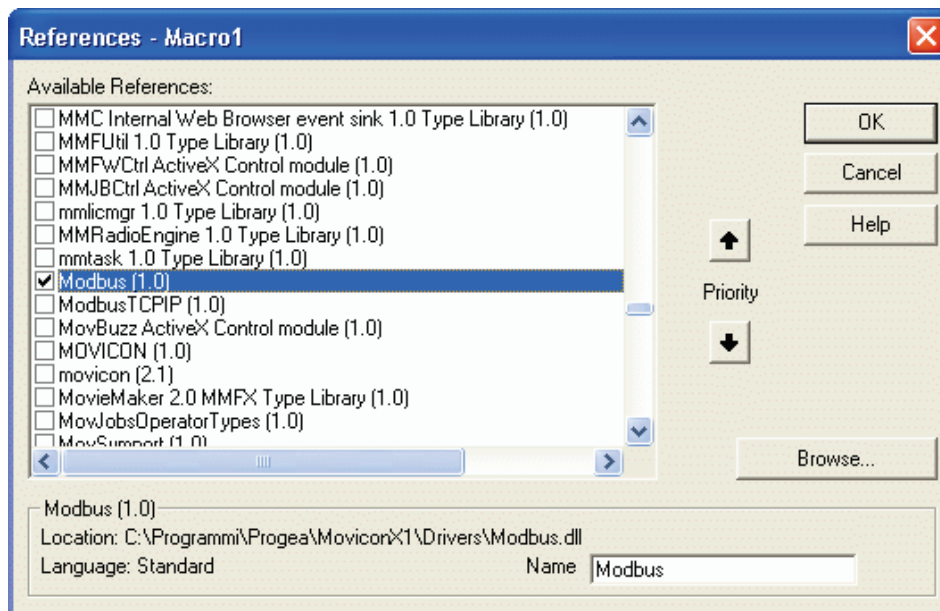
1.32. Utilizzo delle interfacce Basic Script

Il Supervisore consente di utilizzare una serie di funzioni per la gestione dei driver di comunicazione anche tramite le routine Basic Script. In questo modo è possibile ad esempio creare e distruggere dei task di comunicazione durante la fase di runtime, oppure recuperare informazioni sullo stato dei task delle stazioni, ecc.



Per poter utilizzare l'interfaccia COM (Component Object Model) dei driver di comunicazione è necessario avere abilitata l'opzione "VBA Driver" nella licenza. In caso contrario, in fase di runtime, le funzioni basic script dei driver non verranno eseguite.

Per maggiore comodità di programmazione è possibile abilitare il Reference del driver interessato all'interno della routine basic script al fine di poter dichiarare gli oggetti del tipo necessario (es. DriverInterface, TaskInterface, ecc.) e non di tipo generico (Object). In questo modo sarà possibile avere a disposizione la lista dei metodi e delle proprietà di ogni oggetto.



Si tenga però conto che l'abilitazione dei Reference non è supportata su piattaforma WinCE. Se pertanto si dovrà sviluppare un progetto che dovrà poi essere eseguito su WinCE non si dovranno abilitare i Reference e si dovranno dichiarare tutti gli oggetti di tipo Object.

Gli esempi riportati nella documentazione delle funzioni sono stati tutti realizzati abilitando il Reference del driver.

Il seguente esempio può funzionare soltanto su piattaforma Windows 32/64 bit:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim bError As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")

    bError = station.IsInError
    MsgBox "IsInError = " & CStr(bError), vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
End Sub
```

per renderlo compatibile con la piattaforma WinCE i Reference dei driver non dovranno essere abilitati e la dichiarazione delle variabili dovrà diventare la seguente:

```

Sub Main
    Dim drv As Object
    Dim station As Object
    Dim bError As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")

    bError = station.IsInError
    MsgBox "IsInError = " & CStr(bError),vbInformation,"ERROR"

    Set drv = Nothing
    Set station = Nothing
End Sub

```

Task Sincroni e Task Asincroni

Quando si mette in esecuzione un task tramite la funzione "Execute()" si può decidere se il task dovrà essere eseguito in modo Sincrono o Asincrono:

Task Sincroni

Per rendere un task sincrono è sufficiente assegnare alla funzione "Execute()" un tempo di timeout maggiore di zero. In questo modo quando la funzione viene richiamata l'esecuzione dello script attende che la funzione abbia terminato, dopodichè prosegue. A questo punto se la funzione restituisce il valore "True" significa che è stata eseguita, mentre se il valore di ritorno è "False" significa che non è stata eseguita, ad esempio potrebbe essere scaduto il timeout.



La funzione "Execute()" restituisce il valore "True" quando è stata eseguita, il che significa che il task è stato eseguito, ma potrebbe essere stato eseguito con errore. In questo caso quindi è necessario controllare lo stato della Stazione per verificare se questa è in errore oppure no.

I task sincroni vengono normalmente utilizzati in quei casi in cui è necessario avere un controllo sulle avvenute letture/scritture dei dati. Ad esempio se è necessario avere la certezza di avere trasferito una ricetta di produzione prima di dare il comando di l'inizio produzione.



Per un corretto funzionamento del task sincrono è necessario impostare come parametro di timeout della funzione "Execute()" un valore superiore al valore di timeout impostato nella Stazione. Ad esempio se il timeout della stazione è 2000 msec, si può inserire un timeout per la funzione "Execute()" di 5000 msec. Questo evita che possa scattare prima il timeout della funzione rispetto a quello della Stazione.

Task Asincroni

Per rendere un task asincrono è sufficiente assegnare alla funzione "Execute()" un tempo di timeout uguale a zero. In questo modo quando la funzione viene richiamata l'esecuzione dello script continua senza attendere che la funzione abbia terminato. In questo caso quindi il task verrà eseguito quando possibile ma non sarà possibile sapere quando esattamente questo è stato eseguito. In questo caso la funzione restituisce sempre il valore "True" a meno che non ci sia un errore sulla creazione del task, come ad esempio un parametro errato. Il valore "True" per un task asincrono non significa quindi che il task è stato eseguito, ma semplicemente che è stato creato ed è in attesa di essere eseguito.

1.32.1. DriverInterface

DriverInterface Events

1.33. FireOnDataSent, DriverInterface Event

Description	Questo evento viene richiamato quando alcuni dati sono stati inviati da un Task. Può essere usato per ricevere nuove informazioni senza dover eseguire dei polling.
--------------------	---

Remarks Questo Evento è disponibile solo se usato da applicativi esterni. Non è disponibile nei Basic Script del Supervisore.

Parameter	Description
(ByVal IpszTaskName As String)	Nome del Task che ha inviato i dati.

Example:

1.34. FireOnNewData, DriverInterface Event

Description Questo evento viene richiamato quando cambiano i dati di un Task. Può essere usato per ricevere i dati senza dover eseguire dei polling.

Remarks Questo Evento è disponibile solo se usato da applicativi esterni. Non è disponibile nei Basic Script del Supervisore.

Parameter	Description
(ByVal IpszTaskName As String)	Nome del Task nel quale sono cambiati i dati.

Example:

```
Dim WithEvents Drv As ModbusTCP/IP.DriverInterface
Dim task As ModbusTCP/IP.TaskInterface
Dim task2 As ModbusTCP/IP.TaskInterface
Dim var As Variant

Private Sub Drv_FireOnNewData(ByVal IpszTaskName As String)
    If IpszTaskName = "Dyn1" Then
        var = task.GetReadByteBuffer
        Label1.Caption = CStr(var)
    ElseIf IpszTaskName = "Dyn2" Then
        var = task2.GetReadByteBuffer
        Text2.Text = CStr(var)
    End If
End Sub
```

1.35. FireOnQualityChanged, DriverInterface Event

Description Questo evento viene richiamato quando cambia la qualità dei dati di un Task. Può essere usato per ricevere nuove informazioni senza dover eseguire dei polling.

Remarks Questo Evento è disponibile solo se usato da applicativi esterni. Non è disponibile nei Basic Script del Supervisore.

Parameter	Description
(ByVal IpszTaskName As String)	Nome del Task nel quale è cambiata la qualità dei dati.

Example:

```
Dim WithEvents Drv As ModbusTCP/IP.DriverInterface
Dim task As ModbusTCP/IP.TaskInterface
Dim var As Variant

Private Sub Drv_FireOnQualityChanged(ByVal IpszTaskName As String)
    Dim Station As ModbusTCP/IP.StationInterface
```

```

Set Station = task.GetStationObject
lblQuality = CStr(Station.Quality)
If Station.IsInError Then
    lblError = Station.LastErrorString
Else
    lblError = "No Error"
End If
End Sub

```

DriverInterface Functions

1.36. AddDriverTask, DriverInterface Function

Syntax AddDriverTask(lpszName, lpszSettings)

Description Questa funzione inserisce un Task dinamicamente al Driver. Quando un Task dinamico viene creato le sue risorse vengono allocate, ma per metterlo in esecuzione è necessario utilizzare la funzione "Execute". Il Task resterà allocato nel Driver fino a quando non verrà rimosso tramite la funzione "RemoveDriverTask".

Remarks

Parameter	Description
lpszName As String	Nome del Task da creare. Deve essere un nome univoco.
lpszSettings As String	<p>Stringa contenente le impostazioni per la creazione del Task dinamico. La costruzione della stringa dipende dal driver che si sta utilizzando, quindi è necessario far riferimento alle impostazioni dei singoli driver. In linea di massima comunque la stringa è la stessa che viene inserita nella proprietà "Indirizzo Fisico I/O" di una variabile del Supervisore.</p> <p>La stringa è comunque composta da una serie di parole chiave seguite dal relativo valore e separate tra loro dal carattere " " (pipe). Ad esempio per il driver Modbus si potrebbe avere:</p> <p>[DRV]Modbus TCPIP.Sta=ModbusTCP Typ=10 Size=10 TaskType=0 Unit=1 FC=2 SA=20</p> <p>dove:</p> <p>[DRV]: è il parametro che indica l'utilizzo di un driver e dovrà essere seguito dal nome del driver, nell'esempio "Modbus TCPIP"</p> <p>Sta: è il parametro che indica il nome della stazione di riferimento su cui creare l'incarico. Nell'esempio la stazione si chiamerà "ModbusTCP"</p> <p>Typ: indica il tipo di dato che verrà scambiato. Questo parametro potrà assumere i seguenti valori:</p> <p>0 = Bit 1 = Sign Byte 2 = Byte 3 = Sign Word 4 = Word 5 = Sign DWord 6 = DWord 7 = Float 8 = Double 9 = String 10 = Array 11 = Structure</p>

	<p>Size: è il parametro che indica la dimensione in byte di dati che il task dovrà scambiare</p> <p>TaskType: è il parametro che indica di che tipo il task dovrà essere creato. Questo parametro potrà assumere i seguenti valori:</p> <p>0 = Task Input 1 = Task Input/Output 2 = Task Exception Output 3 = Task Unconditional Output</p> <p>Unit: è il parametro che indica il numero dell'unità. Questo parametro è presente solo per il driver Modbus</p> <p>FC: è il parametro che indica il function code da utilizzare. Questo parametro è presente solo per il driver Modbus</p> <p>SA: è il parametro, start address, che indica l'indirizzo di partenza da cui leggere/scrivere i dati sul dispositivo</p>
--	---

Result TaskInterface Object

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim Station As StationInterface
    Dim TaskReset As TaskInterface
    Dim sDynString As String
    Dim vBuffer As Variant

    'Create driver object
    Set drv = GetDriverInterface("Modbus TCP/IP")
    If drv Is Nothing Then
        MsgBox "Modbus TCP/IP driver not installed in the project!",
            vbExclamation, GetProjectTitle
        Exit Sub
    End If

    'Create station object
    Set Station = drv.GetDriverStation("Station1")
    If Station Is Nothing Then
        MsgBox "Station1 didn't found in the driver stations list!", vbExclamation,
            GetProjectTitle
        Exit Sub
    End If

    If Station.IsInError Then
        MsgBox "Station in Error!", vbExclamation, GetProjectTitle
        Exit Sub
    End If

    sDynString = "[DRV]Modbus"
    TCP/IP.Sta=Station1|Typ=4|Size=2|TaskType=2|Unit=1|FC=2|SA=100"

    'Create task object
    Set TaskReset = drv.AddDriverTask("Dyn1", sDynString)
    If TaskReset Is Nothing Then
        MsgBox "the dynamic string '" & sDynString & "' is wrong!",
            vbExclamation, GetProjectTitle
        Exit Sub
    End If

    If Not TaskReset.IsValid Then
        MsgBox "AddDriverTask Failed!", vbExclamation, GetProjectTitle
        Exit Sub
    End If

    TaskReset.PollingTime = 0

    Dim ret As Boolean

    'Reset buffer value
    vBuffer = 0
```

```

ret = TaskReset.SetWriteByteBuffer(vBuffer)

If ret = False Then
    MsgBox "SetWriteByteBuffer Failed!", vbExclamation, GetProjectTitle
Exit Sub
End If

'Execute a sync write command
ret = TaskReset.Execute(5000)

If ret = False Then
    MsgBox "TaskReset.Execute Failed!", vbExclamation, GetProjectTitle
Exit Sub
End If

'Remove task
ret = drv.RemoveDriverTask("Dyn1")

If ret = False Then
    MsgBox "RemoveDriverTask Failed!", vbExclamation, GetProjectTitle
Exit Sub
End If

Set Station = Nothing
Set drv = Nothing
Set TaskReset = Nothing
End Sub

```

1.37. AddStation, DriverInterface Function

Syntax AddStation(lpszName)

Description Crea dinamicamente una nuova Stazione e la inserisce nella lista stazioni del Driver. Quando una Stazione viene creata le sue risorse vengono allocate, ma per metterla in esecuzione è necessario utilizzare la funzione "StartStation". La Stazione resterà allocata fino a quando non è rimossa tramite la funzione "RemoveStation".

Remarks Una stazione può avere qualsiasi nome, purché univoco e non nullo.

Parameter	Description
lpszName As String	Nome della Stazione da creare. Deve essere un nome univoco.

Result Boolean
Se la funzione è eseguita con successo restituisce il valore True . Se la stazione aggiunto ha un nome nullo o la stazione esiste nella lista stazioni è restituito False.

NB: La nuova stazione aggiunta viene parametrizzata con valori delle proprietà predefiniti .

NB: Per memorizzare in modo permanente la nuova stazione (nel file .DrvSettings) richiamare la funzione della SaveConfigDriver.

Example:

```

Option Explicit
Const DRIVER_NAME = "Modbus Serial"

Sub Main
Dim objDrvInt As DriverInterface
Dim sStationName as String
Dim bRet As Boolean

```

```

sStationName = This.GetParameter(0)

If sStationName <> "" Then

    Set objDrvInt = GetDriverInterface(DRIVER_NAME)

    'ADD STATION
    bRet = objDrvInt.AddStation(sStationName)
End If

If Not bRet Then Debug.Print "VBA Driver - AddStation Error! Station
name: " & sStationName & ""

Set objDrvInt = Nothing

End Sub

```

1.38. CloseDriver, DriverInterface Function

Syntax CloseDriver ()

Description Ferma il driver dopo che è stato avviato.

Remarks Questa funzione, al contrario di TerminateDriver, è disponibile nei Basic Script del Supervisore.

Parameter	Description
None	None

Result Boolean
Se la funzione è eseguita con successo, restituisce il valore True .

Example:

```

Option Explicit
Const DRIVER_NAME = "Modbus Serial"

Sub Main
Dim objDrvInt As Object
Dim bRet As Boolean

Set objDrvInt = GetDriverInterface(DRIVER_NAME)

'CLOSE DRIVER
bRet = objDrvInt.CloseDriver()
If bRet = False Then
    Debug.Print "CloseDriver Failed!"
Else
    Debug.Print "CloseDriver Succeeded!"
End If
Set objDrvInt = Nothing

End Sub

```

1.39. GetDriverStation, DriverInterface Function

Syntax GetDriverStation(IpszStation)

Description Questa funzione restituisce un oggetto di tipo Stazione.

Remarks

Parameter	Description
IpszStation As String	Nome della Stazione.

Result Object
Viene restituito un oggetto di tipo StationInterface se la funzione è stata eseguita con successo, altrimenti l'oggetto sarà Nothing.

Example:
Vedere l'esempio della funzione "AddDriverTask".

1.40. GetDriverTask, DriverInterface Function

Syntax GetDriverTask(IpszName)

Description Questa funzione restituisce un oggetto di tipo Task. Il Task richiesto può essere di tipo statico o dinamico.

Remarks

Parameter	Description
IpszName As String	Nome del Task.

Result Object
Viene restituito un oggetto di tipo TaskInterface se la funzione è stata eseguita con successo, altrimenti l'oggetto sarà Nothing.

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim Station As StationInterface
    Dim task As TaskInterface
    Dim var As Variant
    Dim nVal As Integer
    Dim nVal2 As Integer
    Dim AppByte As Byte

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Async")

    If task Is Nothing Then
        MsgBox "Async task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    While Not This.IsStopping
        var = task.GetReadByteBuffer
        If (var(1) And &H80) > 0 Then
```

```

        AppByte = Not(var(0))
        nVal = AppByte + 1
        AppByte = Not(var(1))
        If nVal > 255 Then
            nVal2 = (((AppByte + 1) * 256) + (nVal And &HFF)) * -1
        Else
            nVal2 = ((AppByte * 256) + (nVal And &HFF)) * -1
        End If
        SetVariableValue("AsyncRead", nVal2)
    Else
        SetVariableValue("AsyncRead", var(0) + (var(1)*256))
    End If
    DoEvents
Wend

Set Station = Nothing
Set drv = Nothing
Set task = Nothing
End Sub

```

1.41. InitDriver, DriverInterface Function

Syntax	InitDriver(lpszSerialNumber, lpszFileSettings)
Description	Questa funzione inizializza la comunicazione. Dopo che la funzione è stata eseguita con successo il driver sarà pronto ad operare.
Remarks	Questa funzione è disponibile solo se usata da applicativi esterni. Non è disponibile nei Basic Script del Supervisore.

Parameter	Description
lpszSerialNumber As String	Numero di serie per l'esecuzione del Driver. Contattare l'ufficio commerciale per ottenere una licenza valida.
LpszFileSettings As String	<p>Percorso della cartella contenente il file di impostazioni del Driver. Il file deve avere il seguente nome:</p> <p><DriverDLLName>.drvsettings</p> <p>dove "DriverDLLName" è il nome della libreria del Driver in questione.</p> <p>Il file deve essere in formato XML e può essere editato in tre modi:</p> <ol style="list-style-type: none"> 1. Manualmente tramite un editor di testo 2. Eseguendo preventivamente una chiamata della funzione "SetDriver" 3. Utilizzando l'interfaccia utente di impostazione del Driver messa a disposizione dal Supervisore

Result	<p>Boolean</p> <p>True = il driver è stato inizializzato con successo</p> <p>False = inizializzazione del driver fallita</p>
---------------	--

Example:

```

Private Sub Form_Load()
    Dim bInitOk As Boolean
    Set Drv = CreateObject("SupervisorDriver.ModbusTCP/IP.1")
    'Drv.SetDriver "C:\Projects\TestPrj\RESOURCES\TestPrj\"
    bInitOk = Drv.InitDriver("XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "C:\Projects\TestPrj\RESOURCES\TestPrj\")
    If Not bInitOk Then
        MsgBox "InitDriver failed!", vbExclamation, "ERROR"
    End If
End Sub

```

1.42. OpenDriver, DriverInterface Function

Syntax OpenDriver ()

Description Riavvia il driver dopo che è stato fermato con la funzione CloseDriver.

Remarks I parametri di configurazione del driver vengono ricaricati.
Questa funzione, al contrario di InitDriver, è disponibile nei Basic Script del Supervisore.

Parameter	Description
None	None

Result Boolean
Se la funzione è eseguita con successo, restituisce il valore True .

Example:

```
Option Explicit
Const DRIVER_NAME = "Modbus Serial"

Sub Main
Dim objDrvInt As Object
Dim bRet As Boolean

Set objDrvInt = GetDriverInterface(DRIVER_NAME)

'CLOSE DRIVER
bRet = objDrvInt.OpenDriver()
If bRet = False Then
    Debug.Print "OpenDriver Failed!"
Else
    Debug.Print "OpenDriver Succeeded!"
End If
Set objDrvInt = Nothing

End Sub
```

1.43. RemoveDriverTask, DriverInterface Function

Syntax RemoveDriverTask(IpszName)

Description Questa funzione rimuove un Task dal Driver precedentemente creato liberando così le risorse occupate.

Remarks

Parameter	Description
IpszName As String	Nome del Task.

Result Boolean
True = la funzione è stata eseguita con successo
False = esecuzione della funzione fallita

Example:

Vedere l'esempio della funzione "AddDriverTask".

1.44. RemoveStation, DriverInterface Function

Syntax RemoveStation(lpszName)

Description Distrugge e rimuove una stazione dall'elenco delle stazioni del driver. Questa funzione rimuove sia le stazioni aggiunte dinamicamente con AddStation () che quelle create tramite l'interfaccia utente del driver (file. DrvSettings).

Remarks La funzione ferma l'esecuzione dei task eventualmente definiti per la stazione, inoltre se eseguita su stazioni che già esistono all'avvio del runtime del progetto rende inutilizzabile la Stazione a runtime ma non cancella i suoi dati XML dal file .drvsettings.

Parameter	Description
lpszName As String	Nome della Stazione da rimuovere.

Result Boolean
Se la funzione è eseguita con successo restituisce il valore True .

NB: Per rendere permanente la rimozione (nel file DrvSettings) richiamare al funzione SaveConfigDriver .

Example:

```
Option Explicit
Const DRIVER_NAME = "Modbus Serial"

Sub Main
Dim objDrvInt As DriverInterface
Dim sStationNameas String
Dim bRet As Boolean

sStationName = This.GetParameter(0)

If sStationName <> "" Then

Set objDrvInt = GetDriverInterface(DRIVER_NAME)

'REMOVE STATION
bRet = objDrvInt.RemoveStation(sStationName)
End If

If Not bRet Then Debug.Print "VBA Driver - RemoveStation Error! Station
name: " & sStationName & ""

Set objDrvInt = Nothing

End Sub
```

1.45. SaveConfigDriver, DriverInterface Function

Syntax SaveConfigDriver ()

Description Salva in modo permanente (nel file .DrvSettings) i parametri di configurazione del driver. Può essere utilizzata, per esempio, per salvare una stazione aggiunta dinamicamente con la funzione AddStation.

Remarks

Parameter	Description
None	None

Result Boolean
Se la funzione è eseguita con successo restituisce il valore True .

Example:

```
Option Explicit
Const DRIVER_NAME = "Modbus Serial"

Sub Main
Dim objDrvInt As DriverInterface
Dim bRet As Boolean

Set objDrvInt = GetDriverInterface(DRIVER_NAME)

'SAVE ALL STATIONS
bRet = objDrvInt.SaveConfigDriver

If Not bRet Then Debug.Print "VBA Driver - SaveConfigDriver Error!"

Set objDrvInt = Nothing

End Sub
```

1.46. SetDriver, DriverInterface Function

Syntax SetDriver(IpszFileSettings)

Description Questa funzione apre una finestra di dialogo tramite la quale è possibile editare le impostazioni del Driver di Comunicazione.

Remarks Questa funzione è disponibile solo se usata da applicativi esterni. Non è disponibile nei Basic Script del Supervisore.
La funzione va richiamata prima di avere inizializzato il driver, altrimenti la chiamata fallirà.

Parameter	Description
IpszFileSettings As String	Percorso della cartella contenente il file di impostazioni del Driver. Il file deve avere il seguente nome: <DriverDLLName>.drvsettings dove "DriverDLLName" è il nome della libreria del Driver in questione.

Result Boolean
True = la funzione è stata eseguita con successo
False = esecuzione della funzione fallita

Example:

```
Private Sub Form_Load()
Dim bInitOk As Boolean
Set Drv = CreateObject("SupervisorDriver.ModbusTCPIP.1")

Dim bRet As Boolean
bRet = Drv.SetDriver("C:\Projects\TestPrj\RESOURCES\TestPrj\")
If Not bRet Then
MsgBox " SetDriver failed!",vbExclamation,"ERROR"
End If
End Sub
```


1.47. StartStation, DriverInterface Function

Syntax StartStation(lpszName)

Description Carica e avvia il thread di una nuova Stazione aggiunta dinamicamente dalla funzione AddStation.

Remarks Per avviare la comunicazione di una stazione aggiunta da AddStation è necessario chiamare la funzione startStation.

NB: . Le stazioni create dal file .DrvSettings non hanno bisogno di essere avviate da startStation.

Parameter	Description
lpszName As String	Nome della Stazione .

Result Boolean
Se la funzione è eseguita con successo restituisce il valore True .

Example:

```
Option Explicit
Const DRIVER_NAME = "Modbus Serial"

Sub Main
Dim objDrvInt As DriverInterface
Dim sStationNameas String
Dim bRet As Boolean

sStationName = This.GetParameter(0)

If sStationName <> "" Then

Set objDrvInt = GetDriverInterface(DRIVER_NAME)

'START STATION
bRet = objDrvInt.StartStation(sStationName)
End If

If Not bRet Then Debug.Print "VBA Driver - StartStation Error! Station
name: " & sStationName & ""

Set objDrvInt = Nothing

End Sub
```

1.48. TerminateDriver, DriverInterface Function

Syntax TerminateDriver()

Description Questa funzione termina la comunicazione del Driver precedentemente inizializzata tramite la funzione "InitDriver".

Remarks Questa funzione è disponibile solo se usata da applicativi esterni. Non è disponibile nei Basic Script del Supervisore.

Parameter	Description
-----------	-------------

None	None
------	------

Result Boolean
 True = il driver è stato terminato con successo
 False = terminazione del driver fallita

Example:

```
Private Sub Form_Load()
  Dim bInitOk As Boolean
  Set Drv = CreateObject("SupervisorDriver.ModbusTCP/IP.1")
  'Drv.SetDriver "C:\Projects\TestPrj\RESOURCES\TestPrj\"
  bInitOk = Drv.InitDriver("XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "C:\Projects\TestPrj\RESOURCES\TestPrj\")
  If Not bInitOk Then
    MsgBox "InitDriver failed"
  End If

  If not Drv.TerminateDriver()
    MsgBox "TerminateDriver failed!", vbExclamation, "ERROR"
  End If
End Sub
```

DriverInterface Properties

1.49. DelayEvents, DriverInterface Property

Syntax DelayEvents = Long

Description Questa proprietà imposta o restituisce il tempo di ritardo per l'esecuzione degli eventi del Driver.

Remarks Questa funzione è disponibile solo se usata da applicativi esterni. Non è disponibile nei Basic Script del Supervisore.

Parameter	Description
None	None

Result Long
 Tempo di ritardo interno esecuzione Eventi espresso in Millisecondi.

Example:

```
Private Sub Form_Load()
  Dim INewVal As Long
  Dim bRet As Boolean
  Dim drv As Object

  Set drv = GetDriverInterface("ModbusTCP/IP")
  INewVal = drv.DelayEvents
  INewVal = INewVal + 100
  drv.DelayEvents = INewVal
End Sub
```

1.49.1. TaskInterface

TaskInterface Functions

1.50. Execute, TaskInterface Function

Syntax Execute(dwTimeout)

Description Questa funzione mette in esecuzione un Task. Il task in questione può essere stato preventivamente creato con la funzione **"AddDriverTask"**, oppure può essere un task già esistente nella lista dei task statici del driver e dovrà essere referenziato tramite la funzione **"GetDriverTask"**.

ATTENZIONE! La funzione restituisce il valore True nel caso in cui il task sia stato eseguito, ma per sapere se il task è stato eseguito con successo, quindi senza errori, sarà necessario testare ad esempio lo stato della stazione.

Remarks

Parameter	Description
DwTimeout As Long	Tempo espresso in Millisecondi di Timeout per l'esecuzione del Task. Impostando questo parametro ad un valore maggiore di zero (si consiglia un valore superiore al tempo di timeout impostato nella stazione) il task verrà eseguito in modo Sincrono. Impostando questo parametro a Zero il Task verrà eseguito in modo Asincrono. In ogni caso il Task verrà eseguito solo una volta, quindi se si utilizzano Task Asincroni si dovrà realizzare il codice in modo tale che richiami ciclicamente la funzione.

Result Boolean
True = la funzione è stata eseguita con successo
False = esecuzione della funzione fallita

Example:
Vedere l'esempio della funzione "AddDriverTask" e **"GetDriverTask"**.

1.51. GetFlatVariableList, TaskInterface Functiony

Syntax GetFlatVariableList

Description Questa funzione restituisce la lista delle variabili associate ad un task statico.

Remarks

Parameter	Description
None	None

Result String
Viene restituita una stringa contenente la lista delle variabili separate dal separatore predefinito. Il separatore può essere recuperato tramite la funzione "GetVariableListSeparator".

Example:
Sub Main
 Dim drv As DriverInterface

```

Dim task As TaskInterface
Dim strVarlist As String

Set drv = GetDriverInterface("Modbus TCP/IP")
Set task = drv.GetDriverTask("Task")
If task Is Nothing Then
    MsgBox "Task not found!", vbExclamation, "ERROR"
    Exit Sub
End If

strVarlist = task.GetFlatVariableList
MsgBox "GetFlatVariableList = " & strVarlist, vbInformation, "ERROR"

Set drv = Nothing
Set task = Nothing
End Sub

```

1.52. GetMaxByteSize, TaskInterface Function

Syntax GetMaxByteSize

Description Questa funzione restituisce il numero massimo di byte che possono essere allocati per il buffer di lettura/scrittura di un task.

Remarks

Parameter	Description
None	None

Result Long

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim nSizeBytes As Long

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    nSizeBytes = task.GetMaxByteSize
    MsgBox "GetMaxByteSize = " & CStr(nSizeBytes), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub

```

1.53. GetReadByteBuffer, TaskInterface Function

Syntax GetReadByteBuffer

Description Restituisce l'ultimo valore letto da un Task. Il task in questione dovrà essere stato preventivamente creato con la funzione **"AddDriverTask"** e messo in esecuzione con la funzione "Execute". Se il task è già presente nella lista dei task statici del progetto non sarà necessario creare ed eseguire il task ma basterà referenziarlo tramite la funzione "GetDriverTask".

Remarks

Parameter	Description
None	None

Result Variant
In questa variabile verrà messo il valore letto. Normalmente il tipo di variabile viene dimensionata dal driver in base all'area dati letta. Se però si stanno leggendo i dati di un task statico la funzione ritorna un'array di Byte contenente i dati.

Example1:
Vedere l'esempio della funzione "GetDriverTask".

Example2:
Sub Main
 Dim drv As DriverInterface
 Dim Station As StationInterface
 Dim task As TaskInterface
 Dim ret As Boolean
 Dim var As Variant

 Set drv = GetDriverInterface("Modbus TCP/IP")
 Set task = drv.AddDriverTask("Async", "Sta=Station1|Unit=1|FC=2|SA=2")
 If Not task.IsValid Then
 MsgBox "Task Async is not valid!", vbExclamation, "ERROR"
 Exit Sub
 End If

 task.PollingTime = 500

 While Not This.IsStopping
 ret = task.Execute(5000)
 If ret = False Then
 MsgBox "task.Execute failed!", vbExclamation, "ERROR"
 Exit While
 End If

 var = task.**GetReadByteBuffer**
 SetVariableValue("AsyncRead", var)
 DoEvents
 Wend

 Set Station = Nothing
 Set drv = Nothing
 Set task = Nothing
End Sub

1.54. GetStationObject, TaskInterface Function

Syntax GetStationObject

Description Restituisce l'oggetto Stazione di appartenenza del Task.

Remarks

Parameter	Description
None	None

Result Object
Viene restituito un oggetto di tipo StationInterface se la funzione è stata eseguita con successo, altrimenti l'oggetto sarà Nothing.

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim Station As StationInterface
    Dim task As TaskInterface

    'Create driver object
    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Async")
    If task Is Nothing Then
        MsgBox "Async task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    'Create station object
    Set Station = task.GetStationObject
    If Station.IsInError Then
        MsgBox "Station in Error!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set Station = Nothing
    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.55. GetTotalSizeBytes, TaskInterface Function

Syntax GetTotalSizeBytes

Description Questa funzione restituisce il numero di byte che verranno letti/scritti dal task.

Remarks Se nell'incarico sono utilizzate variabili di tipo bit (si intendono le variabili del RealTimeDB del Progetto), il valore ritornato è il numero di bit (il driver, internamente, riserva un byte per ciascun bit).

Parameter	Description
None	None

Result Long

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim nSizeBytes As Long

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End Sub
```

```

End If

nSizeBytes = task.GetTotalSizeBytes
MsgBox "GetTotalSizeBytes = " & CStr(nSizeBytes),vbInformation,"ERROR"

Set drv = Nothing
Set task = Nothing
End Sub

```

1.56. GetVariableListSeparator, TaskInterface Function

Syntax GetVariableListSeparator

Description Questa funzione restituisce il separatore usato nella lista delle variabili associate ad un task statico.

Remarks

Parameter	Description
None	None

Result String
Viene restituita una stringa contenente il carattere separatore della lista delle variabili (default = ";").

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strListSep As String

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!",vbExclamation,"ERROR"
        Exit Sub
    End If

    strListSep = task.GetVariableListSeparator
    MsgBox "GetVariableListSeparator = " & strListSep,vbInformation,"ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub

```

1.57. GetXMLSettings, TaskInterface Function

Syntax GetXMLSettings

Description Questa funzione restituisce le impostazioni complete del task in formato XML.

Remarks

Parameter	Description
-----------	-------------

None	None
------	------

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strGetXMLSettings As String

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
    Exit Sub
    End If

    strGetXMLSettings = task.GetXMLSettings
    MsgBox "GetXMLSettings = " & strGetXMLSettings, vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.58. HasBeenExecuted, TaskInterface Function

Syntax HasBeenExecuted

Description Questa funzione restituisce l'informazione sul fatto che il task sia stato eseguito in lettura, con successo, almeno una volta. Questa funzione va quindi utilizzata per gli incarichi di "Input" o "Input/Output".

Remarks

Parameter	Description
None	None

Result Boolean
 True = il task è stato eseguito in lettura, con successo, almeno una volta
 False = il task non è ancora stato eseguito in lettura o se lo è stato ha restituito un errore

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bHasBeenExecuted As Boolean

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
    Exit Sub
    End If

    bHasBeenExecuted = task.HasBeenExecuted
    MsgBox "HasBeenExecuted = " & CStr(bHasBeenExecuted), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
```


End Sub

1.59. IsValid, TaskInterface Function

Syntax IsValid

Description Questa funzione restituisce l'informazione sul fatto che il task sia valido oppure no.

Remarks

Parameter	Description
None	None

Result Boolean
True = il task è valido
False = il task non è valido

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bIsValid As Boolean

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
    End If

    bIsValid = task.IsValid
    MsgBox "IsValid = " & CStr(bIsValid), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.60. SetWriteByteBuffer, TaskInterface Function

Syntax SetWriteByteBuffer(vBuffer)

Description Scrive un valore nel buffer di uscita di un Task. Il task in questione dovrà essere stato preventivamente creato con la funzione **"AddDriverTask"** e messo in esecuzione con la funzione "Execute". Se il task è già presente nella lista dei task statici del progetto non sarà necessario creare ed eseguire il task ma basterà referenziarlo tramite la funzione "GetDriverTask".

Remarks

Parameter	Description
VBuffer As Variant	Valore da scrivere sul buffer di uscita del task.

Result Boolean
True = la funzione è stata eseguita con successo
False = esecuzione della funzione fallita

Example:

Vedere l'esempio della funzione "AddDriverTask".

1.61. ToString, TaskInterface Function

Syntax ToString

Description Questa funzione restituisce la stringa di impostazioni dinamiche di un task.

Remarks

Parameter	Description
None	None

Result String
il contenuto della stringa di impostazione dipenderà dal driver utilizzato.

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strToString As String

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    strToString = task.ToString
    MsgBox "ToString = " & strToString, vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

TaskInterface Properties

1.62. AutoDelete, TaskInterface Property

Syntax Boolean = task.AutoDelete

Description Questa proprietà restituisce lo stato dell'impostazione di AutoDelete del task.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Boolean
True = AutoDelete abilitato
False = AutoDelete disabilitato

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bAutoDelete As Boolean

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    bAutoDelete = task.AutoDelete
    MsgBox "AutoDelete = " & CStr(bAutoDelete), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.63. ConditionalVariable, TaskInterface Property

Syntax String = task.ConditionalVariable

Description Questa proprietà restituisce il nome della variabile condizionale associata al task.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strCondVar As String

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    strCondVar = task.ConditionalVariable
    MsgBox "ConditionalVariable = " & strCondVar, vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.64. InUse, TaskInterface Property

Syntax Boolean = task.InUse

Description Questa proprietà restituisce l'informazione sul fatto che il task è in uso oppure no.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Boolean
True = task in uso
False = task non in uso

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bInUse As Boolean

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
    End If

    bInUse = task.InUse
    MsgBox "InUse = " & CStr(bInUse), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.65. LastExecutionTime, TaskInterface Property

Syntax Date = task.LastExecutionTime

Description Questa proprietà restituisce la data dell'ultima esecuzione del task.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Date

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim dLastExec As Date
```

```

Set drv = GetDriverInterface("Modbus TCP/IP")
Set task = drv.GetDriverTask("Task")
If task Is Nothing Then
    MsgBox "Task not found!", vbExclamation, "ERROR"
Exit Sub
End If

dLastExec = task.LastExecutionTime
MsgBox "LastExecutionTime = " & CStr(dLastExec), vbInformation, "ERROR"

Set drv = Nothing
Set task = Nothing
End Sub

```

1.66. Name, TaskInterface Property

Syntax	String = task.Name
Description	Questa proprietà restituisce il nome del task.
Remarks	Proprietà di sola lettura.

Parameter	Description
None	None

Result String

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strName As String

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
Exit Sub
End If

strName = task.Name
MsgBox "Task Name = " & strName, vbInformation, "ERROR"

Set drv = Nothing
Set task = Nothing
End Sub

```

1.67. PollingTimeNoInUse, TaskInterface Property

Syntax	Long = task.PollingTimeNoInUse
Description	Questa proprietà imposta o restituisce il tempo di polling per i task non in uso. Il tempo è espresso in Millisecondi.
Remarks	

Parameter	Description
None	None

Result Long

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim IPollTime As Long

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    IPollTime = task.PollingTimeNoInUse
    MsgBox "PollingTimeNoInUse = " & CStr(IPollTime), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.68. SwapWords, TaskInterface Property

Syntax Boolean = task.SwapWords

Description Questa proprietà restituisce lo stato dell'impostazione di swap delle word del task.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Boolean
True = Swap Word abilitato
False = Swap Word disabilitato

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bSwapWords As Boolean

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    bSwapBytes = task.SwapWords
    MsgBox "SwapWords = " & CStr(bSwapWords), vbInformation, "ERROR"

    Set drv = Nothing
```

```
Set task = Nothing
End Sub
```

1.69. Station, TaskInterface Property

Syntax String = task.Station

Description Questa proprietà restituisce il nome della stazione di appartenenza del task.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strStation As String

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    strStation = task.Station
    MsgBox "Task Station = " & strStation, vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.70. SwapBytes, TaskInterface Property

Syntax Boolean = task.SwapBytes

Description Questa proprietà restituisce lo stato dell'impostazione di swap dei byte del task.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Boolean
True = Swap Byte abilitato
False = Swap Byte disabilitato

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim bSwapBytes As Boolean

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    bSwapBytes = task.SwapBytes
    MsgBox "SwapBytes = " & CStr(bSwapBytes), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub

```

1.71. PollingTime, TaskInterface Property

Syntax Long = task.PollingTime

Description Questa proprietà imposta o restituisce il tempo di polling per i task in uso. Il tempo è espresso in Millisecondi.

Remarks

Parameter	Description
None	None

Result Long

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim IPollTime As Long

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    IPollTime = task.PollingTime
    MsgBox "PollingTime = " & CStr(IPollTime), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub

```

1.72. Type, TaskInterface Property

Syntax Byte = task.Type

Description Questa proprietà restituisce il tipo di task.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Byte
0 = Input Type
1 = Input/Output Type
2 = Exception Output Type
3 = Unconditional Output Type
-2147220992 = non definito

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim nTaskType As Byte

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    nTaskType = task.Type
    MsgBox "Task Type = " & CStr(nTaskType), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.73. WriteOutputsAtStartup, TaskInterface Property

Syntax Boolean = task.WriteOutputsAtStartup

Description Questa proprietà restituisce l'impostazione di scrittura del task all'avvio del progetto. Indica il valore della proprietà "Write Outputs at Startup". Ha significato solo per gli incarichi di "Input/Output" o di "Exception Output".

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Boolean

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set task = drv.GetDriverTask("Task")
    If Not task.IsValid Then
        MsgBox "Task is not valid!", vbExclamation, "ERROR"
    End If
End Sub
```

```

Exit Sub
End If
MsgBox "WriteOutputsAtStartup = " & task.WriteOutputsAtStartup,vbExclamation,"ERROR"
Set drv = Nothing
Set task = Nothing
End Sub

```

1.73.1. StationInterface

StationInterface Functions

1.74. GetBaseStationInterface, StationInterface Function

Syntax GetBaseStationInterface

Description Questa funzione utilizzata a questo livello non esegue alcuna azione e restituisce un oggetto Nothing.

Remarks

Parameter	Description
None	None

Result Nothing

Example:

1.75. GetSubStationInterface, StationInterface Function

Syntax GetSubStationInterface

Description Questa funzione restituisce un oggetto di tipo Sub Stazione. A seconda del tipo di driver l'oggetto restituito potrà essere: "Nothing", "SerialStationInterface", "SerialHalfDuplexBaseStationInterface", "SerialTAPIHalfDuplexBaseStationInterface", "TAPIStationInterface", "TCPIPStationInterface", "RASStationInterface".

Remarks

Parameter	Description
None	None

Result Object:
Nothing
SerialStationInterface
SerialHalfDuplexBaseStationInterface
SerialTAPIHalfDuplexBaseStationInterface
TCPIPStationInterface
TAPIStationInterface se il driver è seriale
RASStationInterface se il driver è TCPIP

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim drv1 As DriverInterface
    Dim station1 As StationInterface
    Dim MODBUSStation As ModbusStationInterface
    Dim SerialTAPIHalfDuplexBaseStation As SerialTAPIHalfDuplexBaseStationInterface
    Dim SerialStation As SerialStationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim TCPIPStation As TCPIPStationInterface
    Dim RASStation As RASStationInterface

    Set drv = GetDriverInterface("Modbus Serial")
    Set drv1 = GetDriverInterface("Modbus TCP/IP")

    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set MODBUSStation = station.GetSubStationInterface
    If MODBUSStation Is Nothing Then
        MsgBox "MODBUSStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set SerialTAPIHalfDuplexBaseStation = MODBUSStation.GetBaseStationInterface
    If SerialTAPIHalfDuplexBaseStation Is Nothing Then
        MsgBox "SerialTAPIHalfDuplexBaseStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPIStation = SerialTAPIHalfDuplexBaseStation.GetBaseStationInterface
    If TAPIStation Is Nothing Then
        MsgBox "TAPIStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set SerialStation = TAPIStation.GetBaseStationInterface
    If SerialStation Is Nothing Then
        MsgBox "SerialStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set station1 = drv1.GetDriverStation("Station2")
    If station1 Is Nothing Then
        MsgBox "Station1 not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set RASStation = station1.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TCPIPStation = RASStation.GetBaseStationInterface
    If TCPIPStation Is Nothing Then
        MsgBox "TCPIPStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set drv = Nothing
    Set station = Nothing
    Set drv1 = Nothing
    Set station1 = Nothing
    Set TAPIStation = Nothing
    Set SerialStation = Nothing
    Set RASStation = Nothing
End Sub
```

```
Set TCPIPStation = Nothing
End Sub
```

1.76. GetTask, StationInterface Function

Syntax GetTask(IpszTaskName)

Description Questa funzione restituisce un oggetto di tipo Task. Il Task richiesto può essere di tipo statico o dinamico.

Remarks

Parameter	Description
IpszTaskName As String	Nome del Task.

Result Object
Viene restituito un oggetto di tipo TaskInterface se la funzione è stata eseguita con successo, altrimenti l'oggetto sarà Nothing.

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim task As TaskInterface
    Dim bInUse As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set task = station.GetTask("Task")
    If task Is Nothing Then
        MsgBox "Task not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    bInUse = task.InUse
    MsgBox "InUse = " & CStr(bInUse), vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
End Sub
```

1.77. GetXMLSettings, StationInterface Function

Syntax GetXMLSettings

Description Questa funzione restituisce le impostazioni complete della stazione in formato XML.

Remarks

Parameter	Description
-----------	-------------

None	None
------	------

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim strGetXMLSettings As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    strGetXMLSettings = station.GetXMLSettings
    MsgBox "GetXMLSettings = " & strGetXMLSettings, vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
End Sub
```

1.78. IsInError, StationInterface Function

Syntax IsInError

Description Questa funzione restituisce lo stato di errore della Stazione.

Remarks

Parameter	Description
None	None

Result Boolean
True = Stazione in errore
False = Stazione ok

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim bError As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    bError = station.IsInError
    MsgBox "IsInError = " & CStr(bError), vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
End Sub
```

StationInterface Properties

1.79. ErrorThreshold, StationInterface Property

Syntax Long = station.ErrorThreshold

Description Questa proprietà imposta o restituisce il parametro della stazione: Error Threshold.

Remarks

Parameter	Description
None	None

Result Long

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim IErrorThreshold As Long

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
    End If

    Exit Sub

    station.ErrorThreshold = 2

    IErrorThreshold = station.ErrorThreshold
    MsgBox "Error Threshold e= " & IErrorThreshold, vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.80. LastErrorCode, StationInterface Property

Syntax Long = station.LastErrorCode

Description Questa proprietà restituisce il codice dell'ultimo errore avvenuto sulla stazione.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Long

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim ILastError As Long

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    ILastError = station.LastErrorCode
    MsgBox "LastErrorCode = " & CStr(ILastError), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.81. LastErrorString, StationInterface Property

Syntax String = station.LastErrorString

Description Questa proprietà restituisce la stringa descrittiva dell'ultimo errore avvenuto sulla stazione.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim ILastError As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    ILastError = station.LastErrorString
    MsgBox "LastErrorString = " & CStr(ILastError), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.82. Name, StationInterface Property

Syntax	String = station.Name
Description	Questa proprietà restituisce il nome della stazione.
Remarks	Proprietà di sola lettura.

Parameter	Description
None	None

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strName As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    strName = station.Name
    MsgBox "Name = " & strName, vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub
```

1.83. Quality, StationInterface Property

Syntax	Integer = station.Quality
Description	Questa proprietà restituisce la qualità della stazione.
Remarks	Proprietà di sola lettura.

Parameter	Description
None	None

Result Integer
I valori restituiti sono i valori di qualità come da specifiche OPC:

```
252 = OPC_STATUS_MASK
3 = OPC_LIMIT_MASK
0 = OPC_QUALITY_BAD
64 = OPC_QUALITY_UNCERTAIN
192 = OPC_QUALITY_GOOD
4 = OPC_QUALITY_CONFIG_ERROR
8 = OPC_QUALITY_NOT_CONNECTED
12 = OPC_QUALITY_DEVICE_FAILURE
16 = OPC_QUALITY_SENSOR_FAILURE
```



```

20 = OPC_QUALITY_LAST_KNOWN
24 = OPC_QUALITY_COMM_FAILURE
28 = OPC_QUALITY_OUT_OF_SERVICE
68 = OPC_QUALITY_LAST_USABLE
80 = OPC_QUALITY_SENSOR_CAL
84 = OPC_QUALITY_EGU_EXCEEDED
88 = OPC_QUALITY_SUB_NORMAL
216 = OPC_QUALITY_LOCAL_OVERRIDE

```

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim nQuality As Integer

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    nQuality = station.Quality
    MsgBox "Quality = " & CStr(nQuality), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub

```

1.84. StateCommandVariable, StationInterface Property

Syntax String = station.StateCommandVariable

Description Questa proprietà imposta o restituisce il parametro della stazione: State/Command Variable.

Remarks

Parameter	Description
None	None

Result String

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim strStateCommandVariable As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    station.StateCommandVariable = "StateCommand"

    strStateCommandVariable = station.StateCommandVariable
    MsgBox "State/Command Variable= " & strStateCommandVariable, vbInformation, "ERROR"

```

```

Set drv = Nothing
Set task = Nothing
End Sub

```

1.85. StationType, StationInterface Property

Syntax Byte = station.StationType

Description Questa proprietà restituisce il tipo della stazione in questione.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Byte
0 = normal. Questo valore viene restituito quando si utilizzano quei driver che si interfacciano con librerie esterne, come ad esempio Modbus TCP/IP, INTERBUS, ecc.
1 = SerialType
2 = TAPIType
3 = TCPIPType
4 = RASTCPIPType
-2147220992 = stazione non definita

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim task As TaskInterface
    Dim nStationType As Byte

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
    End If

    nStationType = station.StationType
    MsgBox "StationType = " & CStr(nStationType), vbInformation, "ERROR"

    Set drv = Nothing
    Set task = Nothing
End Sub

```

1.85.1. SerialStationInterface

SerialStationInterface Functions

1.86. GetSubStationInterface, SerialStationInterface Function

Syntax GetSubStationInterface

Description Questa funzione, a seconda del tipo di driver, utilizzata a questo livello può restituire un oggetto di tipo **"SerialStationInterface"** o **"ModbusStationInterface"**.

Remarks

Parameter	Description
None	None

Result Object:
SerialStationInterface
ModbusStationInterface

Example:
Vedere l'esempio della funzione "GetBaseStationInterface".

1.87. GetXMLSettings, SerialStationInterface Function

Syntax GetXMLSettings

Description Questa funzione restituisce le impostazioni complete della stazione in formato XML.

Remarks

Parameter	Description
None	None

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim SerialStation As SerialStationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim strGetXMLSettings As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPIStation = station.GetSubStationInterface
    If TAPIStation Is Nothing Then
        MsgBox "TAPIStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set SerialStation = TAPIStation.GetBaseStationInterface
    If SerialStation Is Nothing Then
        MsgBox "SerialStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    strGetXMLSettings = SerialStation.GetXMLSettings
```

```

MsgBox "GetXMLSettings = " & CStr(strGetXMLSettings),vbInformation,"ERROR"

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
End Sub

```

SerialStationInterface Properties

1.88. BaudRate, SerialStationInterface Property

Syntax	Long = station.BaudRate
Description	Questa proprietà imposta o restituisce il valore di BaudRate per la COM della stazione in questione.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Long

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim MODBUSStation As ModbusStationInterface
    Dim TAPIHalfDupStation As SerialTAPIHalfDuplexBaseStationInterface
    Dim SerialStation As SerialStationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim lBaudRate As Long
    Dim nParity As Byte
    Dim nPortID As Byte
    Dim nStopBits As Byte
    Dim nByteSize As Byte
    Dim nFlowControl As Byte

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR"
        Exit Sub
    End If

    Set MODBUSStation = station.GetSubStationInterface
    If MODBUSStation Is Nothing Then
        MsgBox "MODBUSStation not found!",vbExclamation,"ERROR"
        Exit Sub
    End If

    Set SerialTAPIHalfDuplexBaseStation = MODBUSStation.GetBaseStationInterface
    If SerialTAPIHalfDuplexBaseStation Is Nothing Then
        MsgBox "SerialTAPIHalfDuplexBaseStation not found!",vbExclamation,"ERROR"
        Exit Sub
    End If

    Set TAPIStation = SerialTAPIHalfDuplexBaseStation.GetBaseStationInterface
    If TAPIStation Is Nothing Then

```

```

        MsgBox "TAPIStation not found!",vbExclamation,"ERROR"
    Exit Sub
End If

Set SerialStation = TAPIStation.GetBaseStationInterface
If SerialStation Is Nothing Then
    MsgBox "SerialStation not found!",vbExclamation,"ERROR"
    Exit Sub
End If

nPortID = SerialStation.PortID
lBaudRate = SerialStation.BaudRate
nByteSize = SerialStation.ByteSize
nParity = SerialStation.Parity
nStopBits = SerialStation.StopBits
nFlowControl = SerialStation.FlowControl

MsgBox "PortID = " & CStr(nPortID) & vbLf & _
    "BaudRate = " & CStr(lBaudRate) & vbLf & _
    "ByteSize = " & CStr(nByteSize) & vbLf & _
    "Parity = " & CStr(nParity) & vbLf & _
    "StopBits = " & CStr(nStopBits) & vbLf & _
    "FlowControl = " & CStr(nFlowControl),vbInformation,"ERROR"

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
End Sub

```

1.89. ByteSize, SerialStationInterface Property

Syntax	Byte = station.ByteSize
Description	Questa proprietà imposta o restituisce valore di ByteSize (Bit di Dati) per la COM della stazione in questione.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Byte

Example:
Vedere l'esempio della proprietà "BaudRate".

1.90. CdTimeout, SerialStationInterface Property

Syntax	Long = station.CdTimeout
Description	Questa proprietà imposta o restituisce il valore di Timeout per il segnale Cd della seriale selezionata per la stazione. Il tempo è espresso in millisecondi.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "RxTimeout".

1.91. CtsTimeout, SerialStationInterface Property

Syntax Long = station.CtsTimeout

Description Questa proprietà imposta o restituisce il valore di Timeout per il segnale Cts della seriale selezionata per la stazione. Il tempo è espresso in millisecondi.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "RxTimeout".

1.92. DsrTimeout, SerialStationInterface Property

Syntax Long = station.DsrTimeout

Description Questa proprietà imposta o restituisce il valore di Timeout per il segnale Dsr della seriale selezionata per la stazione. Il tempo è espresso in millisecondi.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "RxTimeout".

1.93. FlowControl, SerialStationInterface Property

Syntax Byte = station.FlowControl

Description Questa proprietà imposta o restituisce il valore di Controlli di Flusso per la COM della stazione in questione.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Byte
0 = Nessuno
1 = Hardware
2 = Xon / Xoff

Example:
Vedere l'esempio della proprietà "BaudRate".

1.94. KeepPortOpened, SerialStationInterface Property

Syntax Boolean = station.KeepPortOpened

Description Questa proprietà consente di mantenere sempre aperta la seriale durante il runtime del progetto oppure di fare in modo che venga aperta solo quando devono essere scambiati i dati e poi venga chiusa.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Boolean
True = la seriale viene mantenuta aperta
False = la seriale viene aperta e chiusa ogni volta che devono essere scambiati dei dati

Example:
Sub Main
Dim drv As DriverInterface
Dim station As StationInterface
Dim SerialStation As SerialStationInterface
Dim MODBUSStation As ModbusStationInterface
Dim TAPIHalfDupStation As SerialTAPIHalfDuplexBaseStationInterface
Dim TAPIStation As TAPIStationInterface
Dim bKeepPortOpened As Boolean

Set drv = GetDriverInterface("Modbus Serial")
Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
MsgBox "Station not found!", vbExclamation, "ERROR"
Exit Sub
End If

Set MODBUSStation = station.**GetSubStationInterface**

```

If MODBUSStation Is Nothing Then
    MsgBox "MODBUSStation not found!",vbExclamation,"ERROR"
Exit Sub
End If

Set SerialTAPIHalfDuplexBaseStation = MODBUSStation .GetBaseStationInterface
If SerialTAPIHalfDuplexBaseStation Is Nothing Then
    MsgBox "SerialTAPIHalfDuplexBaseStation not found!",vbExclamation,"ERROR"
Exit Sub
End If

Set TAPIStation = SerialTAPIHalfDuplexBaseStation .GetBaseStationInterface
If TAPIStation Is Nothing Then
    MsgBox "TAPIStation not found!",vbExclamation,"ERROR"
Exit Sub
End If

Set SerialStation = TAPIStation.GetBaseStationInterface
If SerialStation Is Nothing Then
    MsgBox "SerialStation not found!",vbExclamation,"ERROR"
Exit Sub
End If

bKeepPortOpened = SerialStation.KeepPortOpened
MsgBox "KeepPortOpened = " & CStr(bKeepPortOpened),vbInformation,"ERROR"

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
End Sub

```

1.95. Parity, SerialStationInterface Property

Syntax	Byte = station.Parity
Description	Questa proprietà imposta o restituisce valore di Parità per la COM della stazione in questione.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result

Byte

0 = Nessuna

1 = Dispari

2 = Pari

3 = Mark

4 = Spazio

Example:
Vedere l'esempio della proprietà "BaudRate".

1.96. PortID, SerialStationInterface Property

Syntax Byte = station.PortID

Description	Questa proprietà imposta o restituisce il valore identificativo della porta COM usata della stazione in questione.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Byte
0 = COM1
1 = COM2
2 = COM3
3 = COM4
.....

Example:
Vedere l'esempio della proprietà "BaudRate".

1.97. RxTimeout, SerialStationInterface Property

Syntax	Long = station.RxTimeout
Description	Questa proprietà imposta o restituisce il valore di Timeout per il segnale Rx della seriale selezionata per la stazione. Il tempo è espresso in millisecondi.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Long

Example:
Sub Main
 Dim drv As DriverInterface
 Dim station As StationInterface
 Dim SerialStation As SerialStationInterface
 Dim MODBUSStation As ModbusStationInterface
 Dim TAPIHalfDupStation As SerialTAPIHalfDuplexBaseStationInterface
 Dim TAPIStation As TAPIStationInterface
 Dim IRxTimeout As Long
 Dim ITxTimeout As Long
 Dim ICdTimeout As Long
 Dim ICtsTimeout As Long
 Dim IDsrTimeout As Long

 Set drv = GetDriverInterface("Modbus Serial")
 Set station = drv.GetDriverStation("Station1")
 If station Is Nothing Then
 MsgBox "Station not found!", vbExclamation, "ERROR"
 Exit Sub
 End If

 Set MODBUSStation = station.**GetSubStationInterface**
 If MODBUSStation Is Nothing Then
 MsgBox "MODBUSStation not found!", vbExclamation, "ERROR"
 Exit Sub
 End If

```

End If

Set SerialTAPIHalfDuplexBaseStation = MODBUSStation..GetBaseStationInterface
If SerialTAPIHalfDuplexBaseStation Is Nothing Then
    MsgBox "SerialTAPIHalfDuplexBaseStation not found!",vbExclamation,"ERROR"
Exit Sub
End If

Set TAPIStation = SerialTAPIHalfDuplexBaseStation..GetBaseStationInterface
If TAPIStation Is Nothing Then
    MsgBox "TAPIStation not found!",vbExclamation,"ERROR"
Exit Sub
End If

Set SerialStation = TAPIStation.GetBaseStationInterface
If SerialStation Is Nothing Then
    MsgBox "SerialStation not found!",vbExclamation,"ERROR"
Exit Sub
End If

IRxTimeout = SerialStation.RxTimeout
ITxTimeout = SerialStation.TxTimeout
ICdTimeout = SerialStation.CdTimeout
ICtsTimeout = SerialStation.CtsTimeout
IDsrTimeout = SerialStation.DsrTimeout

MsgBox "RxTimeout = " & CStr(IRxTimeout) & vbLf & _
    "TxTimeout = " & CStr(ITxTimeout) & vbLf & _
    "CdTimeout = " & CStr(ICdTimeout) & vbLf & _
    "CtsTimeout = " & CStr(ICtsTimeout) & vbLf & _
    "DsrTimeout = " & CStr(IDsrTimeout),vbInformation,"ERROR"

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
End Sub

```

1.98. SizeReceivingQueue, SerialStationInterface Property

Syntax Long = station.SizeReceivingQueue

Description Questa proprietà imposta o restituisce la dimensione del buffer di ricezione per la COM della stazione in questione. Il valore è espresso in Byte.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "BaudRate" .

1.99. SizeTransmissionQueue, SerialStationInterface Property

Syntax Long = station.SizeTransmissionQueue

Description Questa proprietà imposta o restituisce la dimensione del buffer di trasmissione per la COM della stazione in questione. Il valore è espresso in Byte.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Long

Example:

See the "BaudRate" property example.

1.100. StopBits, SerialStationInterface Property

Syntax Byte = station.StopBits

Description Questa proprietà imposta o restituisce valore di Bit di Stop per la COM della stazione in questione.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
None	None

Result Byte
0 = 1 bit di stop
1 = 1.5 bit di stop
2 = 2 bit di stop

Example:

Vedere l'esempio della proprietà "BaudRate".

1.101. TxTimeout, SerialStationInterface Property

Syntax Long = station.TxTimeout

Description Questa proprietà imposta o restituisce il valore di Timeout per il segnale Tx della seriale selezionata per la stazione. Il tempo è espresso in millisecondi.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva apertura della porta seriale.

Parameter	Description
-----------	-------------

None	None
------	------

Result Long

Example:

Vedere l'esempio della proprietà "RxTimeout".

1.101.1. TAPISessionInterface

TAPISessionInterface Functions

1.102. GetBaseStationInterface, TAPISessionInterface Function

Syntax GetBaseStationInterface

Description Questa funzione utilizzata a questo livello restituisce un oggetto di tipo **"SerialStationInterface"**.

Remarks

Parameter	Description
None	None

Result Object
Viene restituito un oggetto di tipo SerialStationInterface se la funzione è stata eseguita con successo, altrimenti l'oggetto sarà Nothing.

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim MODBUSStation As ModbusStationInterface
    Dim TAPIHalfDupStation As SerialTAPIHalfDuplexBaseStationInterface
    Dim SerialStation As SerialStationInterface
    Dim TAPISession As TAPISessionInterface
    Dim SubTAPISession As TAPISessionInterface

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set MODBUSStation = station.GetSubStationInterface
    If MODBUSStation Is Nothing Then
        MsgBox "MODBUSStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPIHalfDupStation = MODBUSStation .GetBaseStationInterface
    If TAPIHalfDupStation Is Nothing Then
        MsgBox "TAPIHalfDupStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPISession = TAPIHalfDupStation .GetBaseStationInterface
```

```

If TAPIStationIs Nothing Then
    MsgBox "TAPIStationnot found!",vbExclamation,"ERROR"
Exit Sub
End If

Set SerialStation = TAPIStation.GetBaseStationInterface
If SerialStation Is Nothing Then
    MsgBox "SerialStation not found!",vbExclamation,"ERROR"
Exit Sub
End If

Set SubTAPIStation = TAPIStation.GetSubStationInterface
If SubTAPIStation Is Nothing Then
    MsgBox "SubTAPIStation not found!",vbExclamation,"ERROR"
Exit Sub
End If

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
Set SubTAPIStation = Nothing
End Sub

```

1.103. GetSubStationInterface, TAPIStationInterface Function

Syntax GetSubStationInterface

Description Questa funzione utilizzata a questo livello restituisce un oggetto uguale a se stesso di tipo "**TAPIStationInterface**".

Remarks

Parameter	Description
None	None

Result Object
Viene restituito un oggetto di tipo TAPIStationInterface se la funzione è stata eseguita con successo, altrimenti l'oggetto sarà Nothing.

Example:
Vedere l'esempio della funzione "GetBaseStationInterface".

1.104. GetXMLSettings, TAPIStationInterface Function

Syntax GetXMLSettings

Description Questa funzione restituisce le impostazioni complete della stazione in formato XML.

Remarks

Parameter	Description
None	None

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim MODBUSStation As ModbusStationInterface
    Dim TAPIHalfDupStation As SerialTAPIHalfDuplexBaseStationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim strGetXMLSettings As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set MODBUSStation = station.GetSubStationInterface
    If MODBUSStation Is Nothing Then
        MsgBox "MODBUSStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPIHalfDupStation = MODBUSStation .GetBaseStationInterface
    If TAPIHalfDupStation Is Nothing Then
        MsgBox "TAPIHalfDupStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPIStation = TAPIHalfDupStation .GetBaseStationInterface
    If TAPIStation Is Nothing Then
        MsgBox "TAPIStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    strGetXMLSettings = TAPIStation.GetXMLSettings
    MsgBox "GetXMLSettings = " & CStr(strGetXMLSettings), vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
    Set TAPIStation = Nothing
End Sub
```

1.105. IsConnected, TAPIStationInterface Function

Syntax IsConnected

Description Questa funzione restituisce lo stato di connessione della Stazione TAPI. Il valore "True" indica che la stazione è in fase di connessione oppure è già stata connessa.

Remarks

Parameter	Description
None	None

Result Boolean
True = Stazione connessa o in fase di connessione
False = Stazione non connessa

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim MODBUSStation As ModbusStationInterface
    Dim TAPIHalfDupStation As SerialTAPIHalfDuplexBaseStationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim bIsConnected As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set MODBUSStation = station.GetSubStationInterface
    If MODBUSStation Is Nothing Then
        MsgBox "MODBUSStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPIHalfDupStation = MODBUSStation .GetBaseStationInterface
    If TAPIHalfDupStation Is Nothing Then
        MsgBox "TAPIHalfDupStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPIStation = TAPIHalfDupStation .GetBaseStationInterface
    If TAPIStation Is Nothing Then
        MsgBox "TAPIStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    bIsConnected = TAPIStation.IsConnected
    MsgBox "IsConnected = " & CStr(bIsConnected), vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
    Set TAPIStation = Nothing
End Sub

```

TAPIStationInterface Properties

1.106. DisconnectAfterSecs, TAPIStationInterface Property

Syntax	Long = station.DisconnectAfterSecs
Description	Questa proprietà imposta o restituisce il tempo espresso in Millisecondi dopo il quale la connessione modem della Stazione TAPI verrà chiusa. Il conteggio del tempo parte dal momento in cui tutti gli incarichi di comunicazione non sono più in uso, e naturalmente viene riavanzato se almeno un incarico torna in uso prima della scadenza del tempo.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result Long

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim MODBUSStation As ModbusStationInterface
    Dim TAPIHalfDupStation As SerialTAPIHalfDuplexBaseStationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim strPhoneNumber As String
    Dim nRetries As Byte
    Dim IDisconAfter As Long
    Dim IRetryAfter As Long
    Dim bEnableTAPI As Boolean
    Dim bPrompt As Boolean
    Dim bShowCon As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set MODBUSStation = station.GetSubStationInterface
    If MODBUSStation Is Nothing Then
        MsgBox "MODBUSStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPIHalfDupStation = MODBUSStation .GetBaseStationInterface
    If TAPIHalfDupStation Is Nothing Then
        MsgBox "TAPIHalfDupStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPIStation = TAPIHalfDupStation .GetBaseStationInterface
    If TAPIStation Is Nothing Then
        MsgBox "TAPIStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    strPhoneNumber = TAPIStation.PhoneNumber
    nRetries = TAPIStation.Retries
    IDisconAfter = TAPIStation.DisconnectAfterSecs
    IRetryAfter = TAPIStation.RetryAfterSecs
    bEnableTAPI = TAPIStation.EnableTAPICallOnThisStation
    bPrompt = TAPIStation.PromptForConnection
    bShowCon = TAPIStation.ShowConnectionDlg

    MsgBox "PhoneNumber = " & CStr(strPhoneNumber) & vbCrLf & _
        "Retries = " & CStr(nRetries) & vbCrLf & _
        "DisconnectAfterSecs = " & CStr(IDisconAfter) & vbCrLf & _
        "RetryAfterSecs = " & CStr(IRetryAfter) & vbCrLf & _
        "EnableTAPICallOnThisStation = " & CStr(bEnableTAPI) & vbCrLf & _
        "PromptForConnection = " & CStr(bPrompt) & vbCrLf & _
        "DsrTimShowConnectionDlgout = " & CStr(bShowCon), vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
    Set TAPIStation = Nothing
End Sub

```

1.107. EnableTAPICallOnThisStation, TAPIStationInterface Property

Syntax

Boolean = station.DisconnectAfterSecs

Description	Questa proprietà imposta o restituisce l'abilitazione ad eseguire la chiamata via modem della Stazione.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result	Boolean True = chiamata TAPI abilitata False = chiamata TAPI disabilitata
---------------	---

Example:
Vedere l'esempio della proprietà "**DisconnectAfterSecs**".

1.108. EndConnectionTime, TAPIStationInterface Property

Syntax	Date = station.EndConnectionTime
Description	Questa proprietà riporta la data e l'ora di fine dell'ultima chiamata via modem della Stazione.
Remarks	Proprietà di sola lettura.

Parameter	Description
None	None

Result	Date
---------------	------

Example:
Vedere l'esempio della proprietà "**StartConnectionTime**".

1.109. LastConnectionTime, TAPIStationInterface Property

Syntax	Date = station.LastConnectionTime
Description	Questa proprietà riporta il tempo totale di connessione via modem della Stazione, per quanto riguarda l'ultima chiamata eseguita.
Remarks	Proprietà di sola lettura.

Parameter	Description
None	None

Result	Date
---------------	------

Example:

Vedere l'esempio della proprietà "**StartConnectionTime**".

1.110. LastTAPIError, TAPIStationInterface Property

Syntax	Long = station.LastTAPIError
Description	Questa proprietà riporta il codice dell'ultimo errore intervenuto durante una connessione via modem della Stazione.
Remarks	Proprietà di sola lettura.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "**StartConnectionTime**".

1.111. LastTAPIErrorString, TAPIStationInterface Property

Syntax	String = station.LastTAPIErrorString
Description	Questa proprietà riporta un testo descrittivo sullo stato della connessione via modem della Stazione. Può essere un messaggio di errore o il semplice stato di connessione.
Remarks	Proprietà di sola lettura.

Parameter	Description
None	None

Result String

Example:

Vedere l'esempio della proprietà "**StartConnectionTime**".

1.112. PhoneNumber, TAPIStationInterface Property

Syntax	String = station.PhoneNumber
Description	Questa proprietà imposta o restituisce il numero di telefono da comporre per la chiamata via modem della Stazione.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result String

Example:

Vedere l'esempio della proprietà "**DisconnectAfterSecs**".

1.113. PromptForConnection, TAPIStationInterface Property

Syntax Boolean = station.PromptForConnection

Description Questa proprietà consente di attivare o disattivare la finestra di conferma per l'esecuzione della chiamata via modem della Stazione. Quando abilitata, prima di eseguire la chiamata il Supervisore mostrerà una finestra di dialogo tramite la quale l'utente dovrà dare conferma. Fintanto che la conferma non verrà inoltrata la chiamata non verrà eseguita.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result Boolean
True = mostra la finestra di conferma per la chiamata
False = esegue la chiamata senza chiedere alcuna conferma

Example:

Vedere l'esempio della proprietà "**DisconnectAfterSecs**".

1.114. Retries, TAPIStationInterface Property

Syntax Byte = station.Retries

Description Questa proprietà imposta o restituisce il numero di tentativi da eseguire per la chiamata via modem della Stazione, nel caso la connessione non vada a buon fine.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result Byte

Example:

Vedere l'esempio della proprietà "**DisconnectAfterSecs**".

1.115. RetryAfterSecs, TAPIStationInterface Property

Syntax	Long = station.RetryAfterSecs
Description	Questa proprietà imposta o restituisce il tempo espresso in Millisecondi dopo il quale verrà eseguita di nuovo la chiamata in caso il tentativo precedente sia fallito.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "**DisconnectAfterSecs**".

1.116. ShowConnectionDlg, TAPIStationInterface Property

Syntax	Boolean = station.ShowConnectionDlg
Description	Questa proprietà consente di attivare o disattivare la visualizzazione della finestra di stato di connessione della chiamata via modem della Stazione.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result Boolean
True = mostra la finestra di connessione in corso
False = non mostra la finestra di connessione in corso

Example:

Vedere l'esempio della proprietà "**DisconnectAfterSecs**".

1.117. StartConnectionTime, TAPIStationInterface Property

Syntax	Date = station.StartConnectionTime
Description	Questa proprietà riporta la data e l'ora di inizio dell'ultima chiamata via modem della Stazione.
Remarks	Proprietà di sola lettura.

Parameter	Description
-----------	-------------

None	None
------	------

Result Date

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim MODBUSStation As ModbusStationInterface
    Dim TAPIHalfDupStation As SerialTAPIHalfDuplexBaseStationInterface
    Dim TAPIStation As TAPIStationInterface
    Dim dStartConn As Date
    Dim dEndConn As Date
    Dim dTotalConn As Date
    Dim dLastConn As Date
    Dim lLastError As Long
    Dim strLastErrorString As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set MODBUSStation = station.GetSubStationInterface
    If MODBUSStation Is Nothing Then
        MsgBox "MODBUSStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPIHalfDupStation = MODBUSStation..GetBaseStationInterface
    If TAPIHalfDupStation Is Nothing Then
        MsgBox "TAPIHalfDupStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TAPIStation = TAPIHalfDupStation..GetBaseStationInterface
    If TAPIStation Is Nothing Then
        MsgBox "TAPIStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    dStartConn = TAPIStation.StartConnectionTime
    dEndConn = TAPIStation.EndConnectionTime
    dTotalConn = TAPIStation.TotalConnectionTime
    dLastConn = TAPIStation.LastConnectionTime
    lLastError = TAPIStation.LastTAPIError
    strLastErrorString = TAPIStation.LastTAPIErrorString

    MsgBox "StartConnectionTime = " & CStr(dStartConn) & vbCrLf & _
        "EndConnectionTime = " & CStr(dEndConn) & vbCrLf & _
        "TotalConnectionTime = " & CStr(dTotalConn) & vbCrLf & _
        "LastConnectionTime = " & CStr(dLastConn) & vbCrLf & _
        "LastTAPIError = " & CStr(lLastError) & vbCrLf & _
        "LastTAPIErrorString = " & CStr(strLastErrorString), vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
    Set TAPIStation = Nothing
End Sub

```

1.118. TotalConnectionTime, TAPIStationInterface Property

Syntax	Date = station.TotalConnectionTime
Description	Questa proprietà riporta il tempo totale di connessione via modem della Stazione. Il tempo viene calcolato su tutte le chiamate eseguite a partire dall'avvio del progetto.
Remarks	Proprietà di sola lettura.

Parameter	Description
None	None

Result Date

Example:
Vedere l'esempio della proprietà "**StartConnectionTime**".

1.118.1. SerialTAPI

SerialTAPIFunctions

1.119. GetBaseStationInterface, SerialTAPIHalfDuplexBaseStationInterface Function

Syntax	GetBaseStationInterface
Description	Questa funzione utilizzata a questo livello restituisce un oggetto di tipo " TAPIStationInterface ".
Remarks	

Parameter	Description
None	None

Result Object
Viene restituito un oggetto di tipo **TAPIStationInterface** se la funzione è stata eseguita con successo, altrimenti l'oggetto sarà Nothing.

Example:
Sub Main
 Dim drv As DriverInterface
 Dim station As StationInterface
 Dim MODBUSStation As ModbusStationInterface
 Dim TAPIHalfDupStation As SerialTAPIHalfDuplexBaseStationInterface
 Dim SerialStation As SerialStationInterface
 Dim TAPIStation As TAPIStationInterface
 Dim SubTAPIStation As TAPIStationInterface

 Set drv = GetDriverInterface("Modbus Serial")
 Set station = drv.GetDriverStation("Station1")
 If station Is Nothing Then

```

        MsgBox "Station not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set MODBUSStation = station.GetSubStationInterface
If MODBUSStation Is Nothing Then
    MsgBox "MODBUSStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set TAPIHalfDupStation = MODBUSStation .GetBaseStationInterface
If TAPIHalfDupStation Is Nothing Then
    MsgBox "TAPIHalfDupStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set TAPIStation= TAPIHalfDupStation .GetBaseStationInterface
If TAPIStation Is Nothing Then
    MsgBox "TAPIStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set SerialStation = TAPIStation.GetBaseStationInterface
If SerialStation Is Nothing Then
    MsgBox "SerialStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set SubTAPIStation = TAPIStation.GetSubStationInterface
If SubTAPIStation Is Nothing Then
    MsgBox "SubTAPIStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set drv = Nothing
Set station = Nothing
Set MODBUSStation = Nothing
Set TAPIStation = Nothing
Set SerialStation = Nothing
Set SubSerialStation = Nothing
Set SubTAPIStation = Nothing
End Sub

```

1.120. GetSubStationInterface, SerialTAPIHalfDuplexBaseStationInterface Function

Syntax GetSubStationInterface

Description Questa funzione utilizzata a questo livello restituisce un oggetto uguale a se stesso di tipo "**SerialTAPIHalfDuplexBaseStationInterface**".

Remarks

Parameter	Description
None	None

Result Object
Viene restituito un oggetto di tipo **SerialTAPIHalfDuplexBaseStationInterface** se la funzione è stata eseguita con successo, altrimenti l'oggetto sarà Nothing.

Example:

```
Sub Main

    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim MODBUSStation As ModbusStationInterface
    Dim SerialTAPIHalfDuplexBaseStation As SerialTAPIHalfDuplexBaseStationInterface

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set MODBUSStation = station.GetSubStationInterface
    If MODBUSStation Is Nothing Then
        MsgBox "MODBUSStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set SerialTAPIHalfDuplexBaseStation = MODBUSStation .GetBaseStationInterface
    If SerialTAPIHalfDuplexBaseStation Is Nothing Then
        MsgBox "SerialTAPIHalfDuplexBaseStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set SerialTAPIHalfDuplexBaseStation = SerialTAPIHalfDuplexBaseStation .GetSubStationInterface
    If SerialTAPIHalfDuplexBaseStation Is Nothing Then
        MsgBox "SerialTAPIHalfDuplexBaseStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set drv = Nothing
    Set station = Nothing
    Set MODBUSStation = Nothing

End Sub
```

1.121. GetXMLSettings, SerialTAPIHalfDuplexBaseStationInterfaceFunction

Syntax GetXMLSettings

Description Questa funzione restituisce le impostazioni complete della stazione in formato XML.

Remarks

Parameter	Description
None	None

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
```



```

Dim MODBUSStation As ModbusStationInterface
Dim TAPIHalfDupStation As SerialTAPIHalfDuplexBaseStationInterface
Dim strGetXMLSettings As String

Set drv = GetDriverInterface("Modbus Serial")
Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
    MsgBox "Station not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set MODBUSStation = station.GetSubStationInterface
If MODBUSStation Is Nothing Then
    MsgBox "MODBUSStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set TAPIHalfDupStation = MODBUSStation .GetBaseStationInterface
If TAPIHalfDupStation Is Nothing Then
    MsgBox "TAPIHalfDupStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

strGetXMLSettings = TAPIHalfDupStation .GetXMLSettings
MsgBox "GetXMLSettings = " & CStr(strGetXMLSettings), vbInformation, "ERROR"

Set drv = Nothing
Set station = Nothing
Set TAPIStation = Nothing
End Sub

```

SerialTAPIProperties

1.122. DelayBeforeFirstChar, SerialTAPIHalfDuplexBaseStationInterface Property

Syntax	Long = objTAPIHalfDpStation.DelayBeforeFirstChar
Description	Questa proprietà imposta o restituisce il tempo che intercorre tra il set a ON del segnale di controllo e la trasmissione di un messaggio. Il tempo è espresso in millisecondi.
Remarks	Questo tempo va considerato non deterministico, perché Windows non è un sistema operativo real-time.

Parameter	Description
None	None

Result Long

Example:

```

Option Explicit
Const DRIVER_NAME = "Modbus Serial"

Dim objDrvInt As DriverInterface
Dim objStation As StationInterface
Dim objModbusStation As ModbusStationInterface
Dim objTAPIHalfDpStation As SerialTAPIHalfDuplexBaseStationInterface

```

```

Enum eHafDupSignal
    None
    RTS
    DTS
End Enum

Sub Main
Dim objDrvInt As DriverInterface
Dim sStationNameas String
    sStationName = This.GetParameter(0)
    If sStationName <> "" Then
        Set objDrvInt = GetDriverInterface(DRIVER_NAME)
        Set objStation = objDrvInt.GetDriverStation(sStationName)
        If Not objStation Is Nothing Then
            Set objModbusStation = objStation.GetSubStationInterface
            'Parametrize Modbus Station properties
            If Not objModbusStation Is Nothing Then
                'Retreive the TAPI Half-Duplex Base Objects
                Set objTAPIHalfDpStation = objModbusStation.GetBaseStationInterface
                If Not objTAPIHalfDpStation Is Nothing Then
                    objTAPIHalfDpStation.DelayBeforeFirstChar = 0
                Else
                    Debug.Print "VBA Driver - objTAPIHalfDpStation Error! Station: " &
objStation.Name & ""
                End If
            Else
                Debug.Print "VBA Driver - objModbusStation is Nothing for the Station" &
sStationName & ""
            End If
        Else
            Debug.Print "VBA Driver - objStation is Nothing for the Station" & sStationName & ""
        End If
        Set objDrvInt = Nothing
        Set objStation= Nothing
        Set objModbusStation = Nothing
        Set objTAPIHalfDpStation = Nothing
    End Sub

```

1.123. DelayAfterLastChar, SerialTAPIHalfDuplexBaseStationInterface Property

Syntax objTAPIHalfDpStation.**DelayAfterLastChar**

Description Questa proprietà imposta o restituisce il tempo che intercorre fra la trasmissione dell'ultimo carattere e l'impostazione ad OFF del segnale di controllo. Il tempo è espresso in millisecondi.

Remarks Questo tempo va considerato non deterministico, perché Windows non è un sistema operativo real-time.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "DelayBeforeFirstChar ".

1.124. HalfDuplexSignal, SerialTAPIOHalfDuplexBaseStationInterface Property

Syntax Long = station.**HalfDuplexSignal**

Description Questa proprietà seleziona o restituisce il segnale di controllo utilizzato per controllare la direzione della trasmissione.
Possibili valori: None, RTS e DTR.

Remarks

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "DelayBeforeFirstChar".

1.124.1. TCPIPStationInterface

TCPIPStationInterface Functions

1.125. GetBaseStationInterface, TCPIPStationInterface Function

Syntax GetBaseStationInterface

Description Questa funzione utilizzata a questo livello restituisce un oggetto di tipo "**StationInterface**".

Remarks

Parameter	Description
None	None

Result Object
Viene restituito un oggetto di tipo StationInterface se la funzione è stata eseguita con successo, altrimenti l'oggetto sarà Nothing.

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim SubRASStation As RASStationInterface
    Dim TCPIPStation As TCPIPStationInterface
    Dim SubTCPIPStation As StationInterface
```

```

Set drv = GetDriverInterface("Modbus TCP/IP")
Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
    MsgBox "Station not found!", vbExclamation, "ERROR"
Exit Sub
End If

Set RASStation = station.GetSubStationInterface
If RASStation Is Nothing Then
    MsgBox "RASStation not found!", vbExclamation, "ERROR"
Exit Sub
End If

Set TCPIPStation = RASStation.GetBaseStationInterface
If TCPIPStation Is Nothing Then
    MsgBox "TCPIPStation not found!", vbExclamation, "ERROR"
Exit Sub
End If

Set SubTCPIPStation = TCPIPStation.GetBaseStationInterface
If SubTCPIPStation Is Nothing Then
    MsgBox "SubTCPIPStation not found!", vbExclamation, "ERROR"
Exit Sub
End If

Set SubRASStation = TCPIPStation.GetSubStationInterface
If SubRASStation Is Nothing Then
    MsgBox "SubRASStation not found!", vbExclamation, "ERROR"
Exit Sub
End If

Set drv = Nothing
Set station = Nothing
Set RASStation = Nothing
Set TCPIPStation = Nothing
Set SubTCPIPStation = Nothing
Set SubRASStation = Nothing
End Sub

```

1.126. GetSubStationInterface, TCPIPStationInterface Function

Syntax GetSubStationInterface

Description Questa funzione, a seconda del tipo di driver, utilizzata a questo livello può restituire un oggetto di tipo "**TCPIPStationInterface**" o "**RASStationInterface**".

Remarks

Parameter	Description
None	None

Result Object:
TCPIPStationInterface
RASStationInterface

Example:
Vedere l'esempio della funzione "GetBaseStationInterface".

1.127. GetXMLSettings, TCPIPStationInterface Function

Syntax GetXMLSettings

Description Questa funzione restituisce le impostazioni complete della stazione in formato XML.

Remarks

Parameter	Description
None	None

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim TCPIPStation As TCPIPStationInterface
    Dim strGetXMLSettings As String

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TCPIPStation = RASStation.GetBaseStationInterface
    If TCPIPStation Is Nothing Then
        MsgBox "TCPIPStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    strGetXMLSettings = TCPIPStation.GetXMLSettings
    MsgBox "GetXMLSettings = " & CStr(strGetXMLSettings), vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
    Set TAPIStation = Nothing
    Set SerialStation = Nothing
End Sub
```

TCPIPStationInterface Properties

1.128. BackupServerAddress, TCPIPStationInterface Property

Syntax String = station.BackupServerAddress

Description Questa proprietà imposta o restituisce l'indirizzo IP del Server di Backup per la stazione. Indica il valore della proprietà "Backup Server Address".

Remarks

Parameter	Description
None	None

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim TCPIPStation As TCPIPStationInterface

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TCPIPStation = RASStation.GetBaseStationInterface
    If TCPIPStation Is Nothing Then
        MsgBox "TCPIPStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    TCPIPStation.BackupServerAddress = "192.168.0.10"
    TCPIPStation.SwitchServerTimeout = 20000

    MsgBox "BackupServerAddress = " & TCPIPStation.BackupServerAddress & vbCrLf & _
        "SwitchServerTimeout = " & _
        TCPIPStation.SwitchServerTimeout, vbExclamation, "ERROR"
    Set drv = Nothing
    Set station = Nothing
    Set RASStation = Nothing
    Set TCPIPStation = Nothing
End Sub
```

1.129. LocalBoundAddress, TCPIPStationInterface Property

Syntax String = station.LocalBoundAddress

Description Questa proprietà imposta o restituisce l'indirizzo IP locale che la Stazione utilizzerà per la connessione. Normalmente va specificato nel caso sul PC siano presenti due o più schede di rete. Se il campo viene lasciato vuoto verrà utilizzato l'indirizzo IP della scheda di default.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo all'apertura del socket. Ad esempio all'avvio del progetto prima che gli incarichi vengano eseguiti almeno una volta.

Parameter	Description
None	None

Result String

Example:

Vedere l'esempio della proprietà "ServerAddress".

1.130. LocalBoundPort, TCPIPStationInterface Property

Syntax long = station.LocalBoundPort

Description Questa proprietà imposta o restituisce il numero di Porta locale che la Stazione utilizzerà per la connessione. Normalmente va specificato nel caso sul PC siano presenti due o più schede di rete. Se il campo viene lasciato vuoto verrà utilizzata la porta di default.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo all'apertura del socket. Ad esempio all'avvio del progetto prima che gli incarichi vengano eseguiti almeno una volta.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "ServerAddress".

1.131. MaxReceive, TCPIPStationInterface Property

Syntax Long = station.MaxReceive

Description Questa proprietà imposta o restituisce la dimensione del buffer (byte) di ricezione dei dati per la stazione.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo all'apertura del socket. Ad esempio all'avvio del progetto prima che gli incarichi vengano eseguiti almeno una volta.

Parameter	Description
None	None

Result Long

Example:

Sub Main

```
Dim drv As DriverInterface
Dim station As StationInterface
Dim RASStation As RASStationInterface
Dim TCPIPStation As TCPIPStationInterface
```

```

Dim IMaxReceive As Long
Dim IMaxSend As Long

Set drv = GetDriverInterface("Modbus TCPIP")
Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
    MsgBox "Station not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set RASStation = station.GetSubStationInterface
If RASStation Is Nothing Then
    MsgBox "RASStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set TCPIPStation = RASStation.GetBaseStationInterface
If TCPIPStation Is Nothing Then
    MsgBox "TCPIPStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

IMaxReceive = TCPIPStation.MaxReceive
IMaxSend = TCPIPStation.MaxSend

MsgBox "MaxReceive = " & CStr(IMaxReceive) & vbCrLf & _
    "MaxSend = " & CStr(IMaxSend), vbInformation, "ERROR"

Set drv = Nothing
Set station = Nothing
Set RASStation = Nothing
Set TCPIPStation = Nothing
End Sub

```

1.132. MaxSend, TCPIPStationInterface Property

Syntax Long = station.MaxSend

Description Questa proprietà imposta o restituisce la dimensione del buffer (byte) di trasmissione dei dati per la stazione.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo all'apertura del socket. Ad esempio all'avvio del progetto prima che gli incarichi vengano eseguiti almeno una volta.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "MaxReceive"

1.133. RxTimeout, TCPIPStationInterface Property

Syntax Long = station.RxTimeout

Description	Questa proprietà imposta o restituisce il tempo di Timeout espresso in millisecondi sulla ricezione dei dati per la stazione.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo all'apertura del socket. Ad esempio all'avvio del progetto prima che gli incarichi vengano eseguiti almeno una volta.

Parameter	Description
None	None

Result Long

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim TCPIPStation As TCPIPStationInterface
    Dim IRxTimeout As Long
    Dim ITxTimeout As Long

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TCPIPStation = RASStation.GetBaseStationInterface
    If TCPIPStation Is Nothing Then
        MsgBox "TCPIPStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    IRxTimeout = TCPIPStation.RxTimeout
    ITxTimeout = TCPIPStation.TxTimeout

    MsgBox "RxTimeout = " & CStr(IRxTimeout) & vbCrLf & _
        "TxTimeout = " & CStr(ITxTimeout), vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
    Set RASStation = Nothing
    Set TCPIPStation = Nothing
End Sub

```

1.134. ServerAddress, TCPIPStationInterface Property

Syntax	String = station.ServerAddress
Description	Questa proprietà imposta o restituisce l'indirizzo IP o il nome del Server a cui la stazione si dovrà connettere.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo all'apertura del socket. Ad esempio all'avvio del progetto prima che gli incarichi vengano eseguiti almeno una volta.

Parameter	Description
None	None

Result String

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim TCPIPStation As TCPIPStationInterface
    Dim strServerAddress As String
    Dim lServerPort As Long
    Dim strLocalBoundAddress As String
    Dim lLocalBoundPort As Long

    Set drv = GetDriverInterface("Modbus TCPIP")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set TCPIPStation = RASStation.GetBaseStationInterface
    If TCPIPStation Is Nothing Then
        MsgBox "TCPIPStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    strServerAddress = TCPIPStation.ServerAddress
    lServerPort = TCPIPStation.ServerPort
    strLocalBoundAddress = TCPIPStation.LocalBoundAddress
    lLocalBoundPort = TCPIPStation.LocalBoundPort

    MsgBox "ServerAddress = " & CStr(strServerAddress) & vbCrLf & _
        "ServerPort = " & CStr(lServerPort) & vbCrLf & _
        "LocalBoundAddress = " & CStr(strLocalBoundAddress) & vbCrLf & _
        "LocalBoundPort = " & CStr(lLocalBoundPort), vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
    Set RASStation = Nothing
    Set TCPIPStation = Nothing
End Sub

```

1.135. ServerPort, TCPIPStationInterface Property

Syntax Long = station.ServerPort

Description Questa proprietà imposta o restituisce il numero di Porta del Server alla quale la stazione si dovrà connettere.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo all'apertura del socket. Ad esempio all'avvio del progetto prima che gli incarichi vengano eseguiti almeno una volta.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "ServerAddress".

1.136. SwitchServerTimeout, TCIPStationInterface Property

Syntax Long = station.SwitchServerTimeout

Description Questa proprietà imposta o restituisce il tempo, in millisecondi, che passa tra il verificarsi di un errore di comunicazione con un server e il tentativo di connettersi all'altro server. Indica il valore della proprietà "Switch Server Timeout".

Remarks

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "BackupServerAddress".

1.137. TxTimeout, TCIPStationInterface Property

Syntax Long = station.TxTimeout

Description Questa proprietà imposta o restituisce il tempo di Timeout espresso in millisecondi sulla trasmissione dei dati per la stazione.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo all'apertura del socket. Ad esempio all'avvio del progetto prima che gli incarichi vengano eseguiti almeno una volta.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "RxTimeout".

RASStationInterface Functions

1.138. GetBaseStationInterface, RASStationInterface Function

Syntax GetBaseStationInterface

Description Questa funzione utilizzata a questo livello restituisce un oggetto di tipo "TCPIPStationInterface".

Remarks

Parameter	Description
None	None

Result Object
Viene restituito un oggetto di tipo TCPIPStationInterface se la funzione è stata eseguita con successo, altrimenti l'oggetto sarà Nothing.

Example:

Sub Main

```
Dim drv As DriverInterface
Dim station As StationInterface
Dim RASStation As RASStationInterface
Dim SubRASStation As RASStationInterface
Dim TCPIPStation As TCPIPStationInterface

Set drv = GetDriverInterface("Modbus Serial")
Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
    MsgBox "Station not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set RASStation = station.GetSubStationInterface
If RASStation Is Nothing Then
    MsgBox "RASStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set TCPIPStation = RASStation.GetBaseStationInterface
If TCPIPStation Is Nothing Then
    MsgBox "TCPIPStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set SubRASStation = RASStation.GetSubStationInterface
If SubRASStation Is Nothing Then
    MsgBox "SubRASStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If

Set drv = Nothing
Set station = Nothing
Set RASStation = Nothing
Set SubRASStation = Nothing
Set TCPIPStation = Nothing
```

End Sub

1.139. GetSubStationInterface, RASStationInterface Function

Syntax GetSubStationInterface

Description Questa funzione utilizzata a questo livello restituisce un oggetto uguale a se stesso di tipo "**RASStationInterface**".

Remarks

Parameter	Description
None	None

Result Object
Viene restituito un oggetto di tipo RASStationInterface se la funzione è stata eseguita con successo, altrimenti l'oggetto sarà Nothing.

Example:
Vedere l'esempio della funzione "GetBaseStationInterface".

1.140. GetXMLSettings, RASStationInterface Function

Syntax GetXMLSettings

Description Questa funzione restituisce le impostazioni complete della stazione in formato XML.

Remarks

Parameter	Description
None	None

Result String

Example:

```
Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim strGetXMLSettings As String

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    strGetXMLSettings = RASStation.GetXMLSettings
```

```

        MsgBox "GetXMLSettings = " & CStr(strGetXMLSettings),vbInformation,"ERROR"

        Set drv = Nothing
        Set station = Nothing
        Set RASStation = Nothing
    End Sub

```

1.141. IsConnected, RASStationInterface Function

Syntax IsConnected

Description Questa funzione restituisce lo stato di connessione della Stazione RAS. Il valore "True" indica che la stazione è in fase di connessione oppure è già stata connessa.

Remarks

Parameter	Description
None	None

Result Boolean
 True = Stazione connessa o in fase di connessione
 False = Stazione non connessa

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim bIsConnected As Boolean

    Set drv = GetDriverInterface("Modbus Serial")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!",vbExclamation,"ERROR"
        Exit Sub
    End If

    Set TAPISStation = station.GetSubStationInterface
    If TAPISStation Is Nothing Then
        MsgBox "TAPISStation not found!",vbExclamation,"ERROR"
        Exit Sub
    End If

    bIsConnected = TAPISStation.IsConnected
    MsgBox "IsConnected = " & CStr(bIsConnected),vbInformation,"ERROR"

    Set drv = Nothing
    Set station = Nothing
    Set TAPISStation = Nothing
End Sub

```

RASStationInterface Properties

1.142. DisconnectAfterSecs, RASStationInterface Property

Syntax	Long = station.DisconnectAfterSecs
Description	Questa proprietà imposta o restituisce il tempo espresso in Millisecondi dopo il quale la connessione modem della Stazione RAS verrà chiusa. Il conteggio del tempo parte dal momento in cui tutti gli incarichi di comunicazione non sono più in uso, e naturalmente viene riazzerato se almeno un incarico torna in uso prima della scadenza del tempo.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result Long

Example:
Vedere l'esempio della proprietà "PhoneBookEntry".

1.143. EnableRASCallOnThisStation, RASStationInterface Property

Syntax	Boolean = station.EnableRASCallOnThisStation
Description	Questa proprietà imposta o restituisce l'abilitazione ad eseguire la chiamata via modem della Stazione.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result Boolean
True = chiamata RAS abilitata
False = chiamata RAS disabilitata

Example:
Vedere l'esempio della proprietà "PhoneBookEntry".

1.144. EndConnectionTime, RASStationInterface Property

Syntax	Date = station.EndConnectionTime
Description	Questa proprietà riporta la data e l'ora di fine dell'ultima chiamata via modem della Stazione.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Date

Example:

Vedere l'esempio della proprietà "**StartConnectionTime**".

1.145. LastConnectionTime, RASStationInterface Property

Syntax Date = station.LastConnectionTime

Description Questa proprietà riporta il tempo totale di connessione via modem della Stazione, per quanto riguarda l'ultima chiamata eseguita.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Date

Example:

Vedere l'esempio della proprietà "**StartConnectionTime**".

1.146. LastRASErrorNumber, RASStationInterface Property

Syntax Long = station.LastRASErrorNumber

Description Questa proprietà riporta il codice dell'ultimo errore intervenuto durante una connessione via modem della Stazione.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Long

Example:

Vedere l'esempio della proprietà "**StartConnectionTime**".

1.147. LastRASErrorString, RASStationInterface Property

Syntax String = station.LastRASErrorString

Description Questa proprietà riporta un testo descrittivo sullo stato della connessione via modem della Stazione. Può essere un messaggio di errore o il semplice stato di connessione.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result String

Example:

Vedere l'esempio della proprietà "**StartConnectionTime**".

1.148. NumRetries, RASStationInterface Property

Syntax Byte = station.NumRetries

Description Questa proprietà imposta o restituisce il numero di tentativi da eseguire per la chiamata via modem della Stazione, nel caso la connessione non vada a buon fine.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result Byte

Example:

Vedere l'esempio della proprietà "PhoneBookEntry".

1.149. Password, RASStationInterface Property

Syntax String = station.Password

Description Questa proprietà imposta o restituisce la password per l'utente con il quale verrà eseguita la chiamata RAS. Questo campo non serve se è già stata inserita la proprietà "PhoneBookEntry".

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
-----------	-------------

None	None
------	------

Result String

Example:

Vedere l'esempio della proprietà "PhoneBookEntry".

1.150. PhoneBookEntry, RASStationInterface Property

Syntax String = station.PhoneBookEntry

Description Questa proprietà imposta o restituisce il nome della connessione RAS preventivamente creata e configurata in Windows.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result String

Example:

Sub Main

```
Dim drv As DriverInterface
Dim station As StationInterface
Dim RASStation As RASStationInterface
Dim strPhoneBook As String
Dim strPhoneNumber As String
Dim strUserName As String
Dim strPassword As String
Dim nNumRetries As Byte
Dim IDisconnectAfter As Long
Dim IRetryAfter As Long
Dim bEnableRASCall As Boolean
Dim bPromptForConn As Boolean
Dim bShowConnDlg As Boolean
```

```
Set drv = GetDriverInterface("Modbus TCP/IP")
Set station = drv.GetDriverStation("Station1")
If station Is Nothing Then
    MsgBox "Station not found!", vbExclamation, "ERROR"
    Exit Sub
End If
```

```
Set RASStation = station.GetSubStationInterface
If RASStation Is Nothing Then
    MsgBox "RASStation not found!", vbExclamation, "ERROR"
    Exit Sub
End If
```

```
strPhoneBook = RASStation.PhoneBookEntry
strPhoneNumber = RASStation.PhoneNumber
strUserName = RASStation.UserName
strPassword = RASStation.Password
nNumRetries = RASStation.NumRetries
IDisconnectAfter = RASStation.DisconnectAfterSecs
IRetryAfter = RASStation.RetryAfterSecs
bEnableRASCall = RASStation.EnableRASCallOnThisStation
```

```

bPromptForConn = RASStation.PromptForConnection
bShowConnDlg = RASStation.ShowConnectionDlg

MsgBox "PhoneBookEntry = " & CStr(strPhoneBook) & vbCrLf & _
"PhoneNumber = " & CStr(strPhoneNumber) & vbCrLf & _
"UserName = " & CStr(strUserName) & vbCrLf & _
"Password = " & CStr(strPassword) & vbCrLf & _
"NumRetries = " & CStr(nNumRetries) & vbCrLf & _
"DisconnectAfterSecs = " & CStr(lDisconnectAfter) & vbCrLf & _
"RetryAfterSecs = " & CStr(lRetryAfter) & vbCrLf & _
"EnableRASCallOnThisStation = " & CStr(bEnableRASCall) & vbCrLf & _
"PromptForConnection = " & CStr(bPromptForConn) & vbCrLf & _
"ShowConnectionDlg = " & CStr(bShowConnDlg),vbInformation,"ERROR"

Set drv = Nothing
Set station = Nothing
Set RASStation = Nothing
End Sub

```

1.151. PhoneNumber, RASStationInterface Property

Syntax	String = station.PhoneNumber
Description	Questa proprietà imposta o restituisce il numero di telefono da chiamare. Questo campo non serve se è già stata inserita la proprietà "PhoneBookEntry".
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result String

Example:
Vedere l'esempio della proprietà "PhoneBookEntry".

1.152. PromptForConnection, RASStationInterface Property

Syntax	Boolean = station.PromptForConnection
Description	Questa proprietà consente di attivare o disattivare la finestra di conferma per l'esecuzione della chiamata via modem della Stazione. Quando abilitata, prima di eseguire la chiamata il Supervisore mostrerà una finestra di dialogo tramite la quale l'utente dovrà dare conferma. Fintanto che la conferma non verrà inoltrata la chiamata non verrà eseguita.
Remarks	Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result Boolean
True = mostra la finestra di conferma per la chiamata
False = esegue la chiamata senza chiedere alcuna conferma

Example:
Vedere l'esempio della proprietà "PhoneBookEntry".

1.153. RetryAfterSecs, RASStationInterface Property

Syntax Long = station.RetryAfterSecs

Description Questa proprietà imposta o restituisce il tempo espresso in Millisecondi dopo il quale verrà eseguita di nuovo la chiamata in caso il tentativo precedente sia fallito.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result Long

Example:
Vedere l'esempio della proprietà "PhoneBookEntry".

1.154. ShowConnectionDlg, RASStationInterface Property

Syntax Boolean = station.ShowConnectionDlg

Description Questa proprietà consente di attivare o disattivare la visualizzazione della finestra di stato di connessione della chiamata via modem della Stazione.

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result Boolean
True = mostra la finestra di connessione in corso
False = non mostra la finestra di connessione in corso

Example:
Vedere l'esempio della proprietà "PhoneBookEntry".

1.155. StartConnectionTime, RASStationInterface Property

Syntax Date = station.StartConnectionTime

Description	Questa proprietà riporta la data e l'ora di inizio dell'ultima chiamata via modem della Stazione.
Remarks	Proprietà di sola lettura.

Parameter	Description
None	None

Result Date

Example:

```

Sub Main
    Dim drv As DriverInterface
    Dim station As StationInterface
    Dim RASStation As RASStationInterface
    Dim dStartConn As Date
    Dim dEndConn As Date
    Dim dTotalConn As Date
    Dim dLastConn As Date
    Dim lLastError As Long
    Dim strLastErrorString As String

    Set drv = GetDriverInterface("Modbus TCP/IP")
    Set station = drv.GetDriverStation("Station1")
    If station Is Nothing Then
        MsgBox "Station not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    Set RASStation = station.GetSubStationInterface
    If RASStation Is Nothing Then
        MsgBox "RASStation not found!", vbExclamation, "ERROR"
        Exit Sub
    End If

    dStartConn = RASStation.StartConnectionTime
    dEndConn = RASStation.EndConnectionTime
    dTotalConn = RASStation.TotalConnectionTime
    dLastConn = RASStation.LastConnectionTime
    lLastError = RASStation.LastRASErrorNumber
    strLastErrorString = RASStation.LastRASErrorString

    MsgBox "StartConnectionTime = " & CStr(dStartConn) & vbCrLf & _
        "EndConnectionTime = " & CStr(dEndConn) & vbCrLf & _
        "TotalConnectionTime = " & CStr(dTotalConn) & vbCrLf & _
        "LastConnectionTime = " & CStr(dLastConn) & vbCrLf & _
        "LastTAPIError = " & CStr(lLastError) & vbCrLf & _
        "LastTAPIErrorString = " & CStr(strLastErrorString), vbInformation, "ERROR"

    Set drv = Nothing
    Set station = Nothing
    Set RASStation = Nothing
End Sub

```

1.156. TotalConnectionTime, RASStationInterface Property

Syntax Date = station.TotalConnectionTime

Description Questa proprietà riporta il tempo totale di connessione via modem della Stazione. Il tempo viene calcolato su tutte le chiamate eseguite a partire dall'avvio del progetto.

Remarks Proprietà di sola lettura.

Parameter	Description
None	None

Result Date

Example:

Vedere l'esempio della proprietà "**StartConnectionTime**".

1.157. UserName, RASStationInterface Property

Syntax String = station.UserName

Description Questa proprietà imposta o restituisce il nome dell'utente con il quale verrà eseguita la chiamata RAS. Tale utente dovrà essere riconosciuto dall'eventuale Server. Questo campo non serve se è già stata inserita la proprietà "PhoneBookEntry".

Remarks Un'eventuale modifica di questo valore verrà acquisito solo alla successiva chiamata.

Parameter	Description
None	None

Result String

Example:

Vedere l'esempio della proprietà "PhoneBookEntry".

1.158. Errori

1.159. Descrizione Errori

Gli errori sono raggruppati per area di pertinenza e sono ordinati in ordine alfabetico.

Sviluppo				
Messaggio di errore	Fase	Gravità	Possibile causa	Possibile soluzione
Internal Error in file '<file_name>' at the line <line_number>. Please contact the Technical Support	sviluppo	serio	errore interno al codice.	call support center
Invalid Address Offset Variable. Enter the name of an existing integer variable	sviluppo	semplice segnalazione	la variabile selezionata non è del tipo corretto.	Inserire il nome di una variabile di tipo intero presente nel RealTimeDB
Invalid conditional variable'<variable_name>' for the task '<task_name>'	sviluppo	semplice segnalazione	la variabile selezionata non è valida: non esiste o è stata	Inserire il nome di una variabile di tipo intero presente nel RealTimeDB

			cancellata o è del tipo errato.	
Invalid conditional variable. Enter only one existing variable of numeric type.	runtime	semplice segnalazione	la variabile selezionata non è valida: non esiste o è stata cancellata o è del tipo errato.	Inserire il nome di una variabile di tipo intero presente nel RealTimeDB
ListView out of memory	sviluppo	serio	memoria insufficiente per la gestione della lista dei task o delle stazioni.	liberare memoria chiudendo eventualmente processi o applicazioni non necessari, riavviare la supervisione o se necessario anche il PC.
Name <name>' is already in use. Please, choose a different name	sviluppo	semplice segnalazione	nome di stazione o task già presente. Le duplicazioni non sono consentite.	cambiare il nome della stazione o del task.
Task size exceeds the maximum allowed size for this protocol	runtime	semplice segnalazione	la dimensione del task eccede la dimensione massima per pacchetto consentita dal protocollo.	ridurre il numero di variabili associate al task.
The selected Dynamic Settings String is either invalid or not allowed for this Variable!	sviluppo	semplice segnalazione	l'indirizzo dinamico specificato per la variabile è invalido o non consentito.	verificare il formato dell'indirizzo dinamico o il tipo di variabile.
The selected COM port is either not supported or is being used by another application. Please select another port.	sviluppo	semplice segnalazione	il numero di porta impostato per la proprietà Station/Serial port settings/Port non è valido	per la proprietà Station/SerialPort settings/Port sono ammessi solo valori interi positivi (non zero)
Value cannot be empty! Please enter a value	sviluppo	semplice segnalazione	solo per driver Ethernet. I valori inseriti nella proprietà TCPIP station/Server address/Server port non sono validi.	inserire un indirizzo valido: indirizzi IP nel formato xxx.xxx.xxx.xxx , o nomi di server i.e. 'server1' inserire un numero di porta: positivo intero (non zero)
Value cannot be zero! Please enter a number greater then 0	sviluppo	semplice segnalazione	il valore per la proprietà 'Rx queue size' o 'Tx queue size' in Station/Queue size' non è valido	inserire un valore intero positivo (non zero)
Variable '<variable_name>' is not valid for the task '<task_name>'	runtime	semplice segnalazione	la variabile selezionata non è valida: non esiste o è stata cancellata o è del tipo errato.	Inserire il nome di una variabile di tipo intero presente nel RealTimeDB. verificare le impostazioni del Task e le regole di indirizzamento.

Validazione indirizzi				
Messaggio di errore	Fase	Gravità	Possibile causa	Possibile soluzione

Job <job_name> (station <station_name>) is invalid and cannot be promoted to the protocol mng	runtime	serio	il task non è valido e non verrà eseguito	verificare la configurazione del task
Setting of addresses for fields of the structure <structure_name> interrupted on unsupported field <field_name>	runtime	semplice segnalazione	errore nella gestione dell'indirizzamento dinamico dei membri di tipo struttura (ad es. stringhe). Un membro è di un tipo non supportato: verranno scambiati i soli membri che precedono nella struttura il membro di tipo errato.	modificare la struttura o utilizzare tag singoli per tipi come ad esempio stringhe.
The Task <task_name> for the Station <station_name> is invalid	runtime	serio	il task non è valido e quindi non verrà eseguito	verificare i parametri di comunicazione del task

Hardware				
Messaggio di errore	Fase	Gravità	Possibile causa	Possibile soluzione
Error loading <drivername>.dll	runtime	fatale	la dll del driver non è stata trovata	verificare se il file <nomedriver>.dll è presente nella sottodirectory 'Drivers' all'interno della cartella di installazione del Supervisore

Comunicazione				
Messaggio di errore	Fase	Gravità	Possibile causa	Possibile soluzione
Communication error : station <station_name> , error <error_description>	runtime	various	questo è un messaggio generico per la comunicazione: la descrizione dell'errore fornisce una spiegazione più dettagliata dell'errore	verificare il possibile errore seguendo le informazioni riportate dal messaggio
Error ! Driver has already been initialized !	runtime	fatale	comportamento inaspettato, si è tentata una inizializzazione del driver quando questo era già in esecuzione.	contattare il supporto
Failed to open the communication device. The configured COM port is either not supported or is being used by another application	communication test	serio	fallito il tentativo di apertura della porta seriale (solo per driver seriali) o di un socket	verificare il numero della porta inserita, verificare la corretta connessione dei cavi, verificare che la porta non sia già usata da un'altra applicazione, verificare l'accessibilità del dispositivo

Messaggi di comunicazione				
Messaggio di errore	Fase	Gravità	Possibile causa	Possibile soluzione
Communication established : station <station_name>	runtime	informazione	la comunicazione è stata correttamente stabilita	
Communication restored	runtime	informazione	la comunicazione è stata ripristinata dopo un problema di comunicazione	

Station quality changed : station <station_name>, quality bitfield <quality_value>	runtime (debug window)	informazione	la qualità dei dati della stazione è cambiata. I possibili valori sono bad, good, uncertain, not connected.	
Task <task_name> (Station <station_name>) is now in use...	runtime (debug window)	informazione	se la gestione delle variabili InUso è attiva , i task dinamici sono attivi se le variabili sono in uso	
Task <task_name> (Station <station_name>) is now NOT in use...	runtime (debug window)	informazione	se la gestione delle variabili InUso è attiva , i task dinamici NON sono attivi se le variabili NON sono in uso	
The station <station_name> has switched from server <primary_server_name> to <secondary_server_name>	runtime	informazione	specifico per driver Ethernet. Se è stato definito un server di backup nella proprietà Network services/Redundancy e il server primario non è disponibili, la stazione viene gestita dal server di backup e viceversa	

Comunicazione Seriale (solo per driver seriali)				
Messaggio di errore	Fase	Gravità	Possibile causa	Possibile soluzione
Break Detect !	runtime	serio	la linea è interrotta per cui non c'è comunicazione	verificare la connessione dei cavi e le impostazioni dei dispositivi sia sull'applicazione che sui dispositivi stessi
Carrier Detect Timeout !	runtime	serio	il driver non ha ricevuto la risposta attesa nel tempo massimo disponibile (timeout)	verificare la connessione dei cavi e le impostazioni dei dispositivi sia sull'applicazione che sui dispositivi stessi
Clear To Send Timeout !	runtime	serio	il driver non ha ricevuto la risposta attesa nel tempo massimo disponibile (timeout)	verificare la connessione dei cavi e le impostazioni dei dispositivi sia sull'applicazione che sui dispositivi stessi
Data Set Ready Timeout !	runtime	serio	il driver non ha ricevuto la risposta attesa nel tempo massimo disponibile (timeout)	verificare la connessione dei cavi e le impostazioni dei dispositivi sia sull'applicazione che sui dispositivi stessi
FATAL ERROR! The Windows message queue is full! Serial Comm messages lost! Increase the message queue depth.	runtime	fatale	la coda dei messaggi è piena, aumentare le dimensioni della coda dei messaggi (alcuni messaggi sono andati perduti)	aumentare le dimensioni della coda dei messaggi
Frame Error !	runtime	serio	rumore sulla linea di comunicazione	rimuovere la possibile sorgente di rumore, verificare il bit di parità o lo stop bit
Overrun Error !	runtime	serio	dati in arrivo prima che il pacchetto precedente sia stato ricevuto; il	ridurre il valore di baud rate, verificare l'hardware, chiudere eventuali applicazioni

			sistema operativo è intasato	che possono occupare risorse di sistema
Parity Error !	runtime	serio	rumore sulla linea di comunicazione	rimuovere la possibile sorgente di rumore, verificare il bit di parità o lo stop bit
Rx Timeout !	runtime	serio	il driver non ha ricevuto la risposta attesa nel massimo tempo ammissibile (timeout)	verificare la connessione dei cavi e le impostazioni dei dispositivi sia sull'applicazione che sui dispositivi stessi, verificare, se esistono, le impostazioni dei convertitori RS232/RS485
Rx Buffer Overflow !	runtime	serio	il buffer di ricezione è pieno, possibile perdita di dati	aumentare le proprietà Station/Queue size/Rx, attivare la proprietà Station/serial port settings/Flow control o hardware flow control se supportato dal dispositivo
The selected COM port is either not supported or is being used by another application. Please select another port.	runtime	serio	la porta COM non esiste o è attualmente in uso da un'altra applicazione.	verificare il numero della porta in Station/SerialPort settings/Port; liberare la porta chiudendo l'applicazione che la tiene occupata e riavviare il PC se necessario
Tx Buffer Full !	runtime	serio	il buffer di trasmissione è pieno, i messaggi in uscita non vengono inviati	aumentare la proprietà Station/Queue size/Tx queue
Unexpected WM_QUIT received !	runtime	informazione	il driver ha ricevuto un messaggio di chiusura dal sistema operativo e si sta chiudendo	

TAPI (solo per driver seriali)				
Messaggio di errore	Fase	Gravità	Possibile causa	Possibile soluzione
Call to <phone_number> has been hung up	runtime	informazione	la comunicazione è stata interrotta dal chiamante	
Disconnected: Bad Address	runtime	serio	l'indirizzo di destinazione non è valido	verificare il numero telefonico da chiamare
Disconnected: Busy	runtime	serio	la stazione remota è occupata	attendere che la stazione remota si liberi
Disconnected: Congestion	runtime	serio	la rete è occupata per cui la chiamata non andrà a buon fine	attendere che la linea si liberi
Disconnected: Incompatible	runtime	serio	si è verificata una incompatibilità rispetto alla chiamata richiesta	verificare l'hardware della stazione remota
Disconnected: Local phone picked up	runtime	serio	la comunicazione è stata interrotta da qualcuno	
Disconnected: No Answer	runtime	serio	la chiamata è stata effettuata ma non vi è stata risposta	
Disconnected: No Dial Tone	runtime	serio	non è stato rilevato il tono di linea quando atteso	verificare l'integrità delle connessioni e della linea

Disconnected: Unknown reason	runtime	serio	la chiamata è stata interrotta per ragioni sconosciute	
Disconnected: Unreachable	runtime	serio	il numero telefonico non è raggiungibile	
Line busy	runtime	semplice segnalazione	si sta tentando di chiamare con la linea occupata	attendere che la linea sia liberata
Placing call to <phone_number> ...	runtime	informazione	si specifica il numero che si sta tentando di chiamare	
Remote Party Disconnected	runtime	serio	la comunicazione è stata terminata dal chiamato	
Remote Party rejected call	runtime	serio	la chiamata è stata rifiutata dal chiamato	
TAPI in special information state, probably couldn't dial number	runtime	serio	la connessione è fallita	
Unable to find modem associated with <port>	runtime	serio	il modem non è stato rilevato sulla porta specificata	verificare nel pannello di controllo se è stata assegnata la porta giusta al modem, verificare le impostazioni e i cablaggi del modem

Bridging (solo per driver seriali)				
Messaggio di errore	Fase	Gravità	Possibile causa	Possibile soluzione
A modem has been connected to the Bridging Port, station <station_name>	runtime	informazione		
A remote device or computer has taken control through the Bridging Port! - station <station_name>	runtime	informazione	l'operazione di bridging è stata avviata	
Connection window creation failed : station <station_name>	runtime or communication test	serio	possibile quantità di memoria insufficiente per le risorse	verificare la memoria a disposizione per le risorse
Could not initialise the Modem on the Bridging Port, station <station_name>	runtime	serio	il comando di inizializzazione è stato inviato ma non vi è stata una risposta corretta	verificare lo stato del modem, i settaggi e i cablaggi
Could not perform read operation on the Bridging Port, station <station_name>	runtime	serio	impossibile leggere i dati	verificare lo stato del modem, i settaggi e i cablaggi
Could not perform write operation on the Bridging Port, station <station_name>	runtime	serio	impossibile scrivere i dati	verificare lo stato del modem, i settaggi e i cablaggi
Device control has been restored - Bridging ended! - station	runtime	informazione	terminata operazione di bridging	
Disconnection sequence received on the Bridging Port, station	runtime	informazione	non c'è più alcuna attività di bridging quindi la comunicazione verrà interrotta	
The modem has been disconnected from the Bridging Port, station <station_name>	runtime	informazione		

Too many characters received on the Bridging Port, station	runtime	serio	sono stati ricevuti più caratteri di quelli ammessi dalla dimensione interna del buffer	resettare l'operazione di bridging
--	---------	-------	---	------------------------------------

RAS (solo per driver ethernet)				
Messaggio di errore	Fase	Gravità	Possibile causa	Possibile soluzione
Connection window creation failed : station <station_name>	runtime or communication test	serio	errore interno: possibile quantità di memoria insufficiente per le risorse	verificare la memoria a disposizione per le risorse
RAS Dial Error : <error-description>, Station <station_name>	runtime or communication test	serio	verificare l'errore specifico descritto	verificare l'errore specifico descritto
Station <station_name> ' is about to dial-up. Continue ?	runtime	informazione	richiesta di connessione. Questo messaggio viene visualizzato se la proprietà ' Prompt before connect' Station\RAS è stata abilitata.	
Undefined RAS Dial Error <error_number>, Station<station_name>	runtime	serio	errore generico sulla connessione RAS	verificare i parametri RAS nelle proprietà della stazione, verificare i cablaggi, verificare le impostazioni RAS su PC chiamato

Interfacce Basic Script				
Messaggio di errore	Fase	Gravità	Possibile causa	Possibile soluzione
Closing driver ...	runtime	informazione	Questo messaggio viene visualizzato quando viene invocata la funzione DriverInterface.Close Driver	
Driver closed	runtime	informazione	La funzione DriverInterface.Close Driver è stata completata con successo	
Error ! Driver has already been initialized !	runtime	semplice segnalazione	La funzione DriverInterface.Open Driver è stata richiamata con il driver già avviato	Se si desidera re-inizializzare il driver, richiamare DriverInterface.Close Driver prima di DriverInterface.Open Driver
Error! Driver already closed or application is shutting down!	runtime	semplice segnalazione	La funzione DriverInterface.Close Driver è stata richiamata con il driver già chiuso	

1.160. About

Tramite questa finestra è possibile verificare la **Versione del Driver** selezionato ed eventuali commenti e descrizioni associati.



Si consiglia di verificare sempre che la versione del file del Driver di Supervisione sia l'ultima disponibile, ed in caso contrario, si consiglia di aggiornare il file (.DLL) scaricandolo dal sito Web del supporto tecnico secondo le modalità previste.



Movicon è un sistema SCADA/HMI per Windows™ interamente progettato e realizzato da Progea.

© 2016 - Tutti i diritti riservati.

E' vietata la riproduzione o la divulgazione del presente manuale o del programma senza la preventiva autorizzazione scritta di Progea Srl.

Le informazioni contenute in questo manuale sono soggette a modifica senza preventiva segnalazione e senza comportare alcun vincolo all'ente realizzatore.



Via D'annunzio 295
41123 Modena - Italy
Tel. +39 059 451060
Fax +39 059 451061
Email: info@progea.com
Http://www.progea.com



Via XX Settembre, 30
Tecnocity Alto Milanese
20025 Legnano (MI) Italy
Tel. +39 0331 486653
Fax +39 0331 455179
Email: willems@progea.com



Progea Deutschland GmbH
Marie-Curie-Str. 12
D-78048 VS-Villingen
Tel: +49 (0) 7721 / 99 25 992
Fax: +49 (0) 7721 / 99 25 993
info@progea.de



Progea International Ltd
via Sottobisio 28
6828 Balerna - Switzerland
tel +41 (91) 9676610
fax +41 (91) 9676611
international@progea.com



Progea North America Corp.
2380 State Road 44 suite C
Oshkosh, WI 54904
Tel. +1 (888) 305 2999
Fax. +1 (920) 257 4213
info@progea.us