

# SMCCOMM.OCX

Communications Driver for SMC Controllers.

## List of features

|         | Item         | Name                     | Parameters  | Return              |
|---------|--------------|--------------------------|---|---------------------|
| Private | Function     | SignedInteger()          | ByVal NString As String, ByVal MsBytePlace As Long  | As Integer          |
| Private | Function     | SignedLong()             | ByVal NString As String, ByVal MsBytePlace As Long  | As Long             |
| Public  | Function     | CaptureFilename()        | ByVal LogFileName As String   | As String           |
| Public  | Function     | Copy()                   | -   | As Long             |
| Public  | Function     | Disconnect()             | -   | As Long             |
| Public  | Function     | GetArraySize()           | ArrayName As String   | As Integer          |
| Public  | Function     | GetAxisRecord()          | ByVal AxisLetter As String  | As AxisInfo         |
| Public  | Function     | GetEthernetHandleUsage() | Handle As String  | As HandleInfo       |
| Public  | Function     | GetLocalIP()             | -   | As String           |
| Public  | Function     | GetMyHandleOnSMC()       | -   | As String           |
| Public  | Function     | GetOccuranceTotal()      | InputString As String, Separator As String  | As Long             |
| Public  | Function     | GetProgramLines()        | -   | As Integer          |
| Public  | Function     | GetSerialNumber()        | -   | As String           |
| Public  | Function     | GetSMCRecord()           | -   | As SMCInfo          |
| Public  | Function     | LinesInController()      | -   | As Integer          |
| Public  | Function     | Output()                 | StringOut As String   | As Long             |
| Public  | Function     | Paste()                  | -   | As Long             |
| Public  | Function     | SendCommand2()           | CommandToSend As String, Optional CMD As Boolean = true, Optional CheckEcho As Boolean=true, Optional VerifySuccess as Boolean=True | As String           |
| Public  | Function     | Upload()                 | -   | As String           |
| Public  | Function     | UploadArray()            | ArrayName As String, StartIdx As Integer, EndIdx As Integer   | As String           |
| Public  | Property Get | Appearance()             | -   | As Integer          |
| Public  | Property Get | BackColor()              | -   | As OLE_COLOR        |
| Public  | Property Get | BorderStyle()            | -   | ~                   |
| Public  | Property Get | CheckSMCEcho()           | -   | As CheckSMCEchoType |
| Public  | Property Get | CommSettings()           | -   | As String           |
| Public  | Property Get | CommType()               | -   | As CommTypes        |

|        |              |                  |   |                       |
|--------|--------------|------------------|---|-----------------------|
| Public | Property Get | DTREnable()      | -   | As Boolean            |
| Public | Property Get | EchoOn()         | -   | As Boolean            |
| Public | Property Get | Enabled()        | -   | As Boolean            |
| Public | Property Get | Font()           | -   | As Font               |
| Public | Property Get | FontBold()       | -   | As Boolean            |
| Public | Property Get | FontItalic()     | -   | As Boolean            |
| Public | Property Get | FontName()       | -   | As String             |
| Public | Property Get | FontSize()       | -   | As Single             |
| Public | Property Get | FontStrikethru() | -   | As Boolean            |
| Public | Property Get | FontUnderline()  | -   | As Boolean            |
| Public | Property Get | ForeColor()      | -   | As OLE_COLOR          |
| Public | Property Get | Handshaking()    | -   | As HandshakeConstants |
| Public | Property Get | hwnd()           | -   | As Long               |
| Public | Property Get | MouseIcon()      | -   | As Picture            |
| Public | Property Get | MousePointer()   | -   | As Integer            |
| Public | Property Get | MyIP()           | -   | As String             |
| Public | Property Get | OnlineStatus()   | -   | As Boolean            |
| Public | Property Get | Port()           | -   | As Integer            |
| Public | Property Get | Protocol()       | -   | As ProtocolConstants  |
| Public | Property Get | RemoteHost()     | -   | As String             |
| Public | Property Get | RemoteHostIP()   | -   | As String             |
| Public | Property Get | RemotePort()     | -   | As Long               |
| Public | Property Get | ReplyTimeout()   | -   | As Long               |
| Public | Property Get | RTSEnable()      | -   | As Boolean            |
| Public | Property Get | SelLength()      | -   | As Long               |
| Public | Property Get | SelStart()       | -   | As Long               |
| Public | Property Get | SelText()        | -   | As String             |
| Public | Property Get | SendPercentage() | -   | As Integer            |
| Public | Property Get | Text()           | -   | As String             |
| Public | Property Get | ToolTipText()    | -   | As String             |
| Public | Property Let | Appearance()     | ByVal New Appearance As Integer             | -                     |
| Public | Property Let | BackColor()      | ByVal New BackColor As OLE_COLOR            | -                     |
| Public | Property Let | BorderStyle()    | ByVal New BorderStyle As Integer            | -                     |
| Public | Property Let | CheckSMCecho()   | ByVal CheckType As CheckSMCechoType         | -                     |
| Public | Property Let | CommSettings()   | ByVal New CommSettings As String            | -                     |
| Public | Property Let | CommType()       | Value As CommTypes                          | -                     |
| Public | Property Let | DTREnable()      | ByVal New DTREnable As Boolean              | -                     |
| Public | Property Let | EchoOn()         | ByVal New EchoOn As Boolean                 | -                     |
| Public | Property Let | Enabled()        | ByVal Enabled As Boolean                    | -                     |
| Public | Property Let | FontBold()       | ByVal New FontBold As Boolean               | -                     |
| Public | Property Let | FontItalic()     | ByVal New FontItalic As Boolean             | -                     |
| Public | Property Let | FontName()       | ByVal New FontName As String                | -                     |
| Public | Property Let | FontSize()       | ByVal New FontSize As Single                | -                     |
| Public | Property Let | FontStrikethru() | ByVal New FontStrikethru As Boolean         | -                     |
| Public | Property Let | FontUnderline()  | ByVal New FontUnderline As Boolean          | -                     |
| Public | Property Let | ForeColor()      | ByVal New ForeColor As OLE_COLOR            | -                     |
| Public | Property Let | Handshaking()    | ByVal New Handshaking As HandshakeConstants | -                     |
| Public | Property Let | MousePointer()   | ByVal New MousePointer As Integer           | -                     |

|        |                 |                  |   |           |
|--------|-----------------|------------------|---|-----------|
| Public | Property Let    | Port()           | ByVal New Port As Integer   | -         |
| Public | Property Let    | Protocol()       | ByVal New Protocol As ProtocolConstants   | -         |
| Public | Property Let    | RemoteHost()     | ByVal New RemoteHost As String  | -         |
| Public | Property Let    | RemotePort()     | ByVal New RemotePort As Long  | -         |
| Public | Property Let    | ReplyTimeout()   | NewReplyTime As Long  | -         |
| Public | Property Let    | RTSEnable()      | ByVal New RTSEnable As Boolean  | -         |
| Public | Property Let    | SelLength()      | ByVal New SelLength As Long   | -         |
| Public | Property Let    | SelStart()       | ByVal New SelStart As Long  | -         |
| Public | Property Let    | SelText()        | ByVal New SelText As String   | -         |
| Public | Property Let    | Text()           | ByVal New Text As String  | -         |
| Public | Property Let    | ToolTipText()    | ByVal New ToolTipText As String   | -         |
| Public | Property Set    | MouseIcon()      | ByVal New MouseIcon As Picture  | -         |
| Public | PropertySet     | Font()           | ByVal New Font As Font  | -         |
| Public | Static Function | SendCommand()    | CommandToSend As String   | As String |
| Public | Sub             | AddText()        | Str As String   | -         |
| Public | Sub             | Connect()        | -   | -         |
| Public | Sub             | Download()       | SMCFile As String, Optional Verify As Boolean = false   | -         |
| Public | Sub             | DownloadArray()  | ArrayName As String, ArrayFile As String, Delimiter As String, StartIdx As Integer, EndIdx As Integer, Optional Verify As Boolean=false | -         |
| Public | Sub             | TestProgress()   | Marked for deletion   | -         |
| Public | Event           | Click()          | -   | -         |
| Public | Event           | ConnectError()   | ErrorCode As Integer, ErrorDescription As String  | -         |
| Public | Event           | DbClick()        | -   | -         |
| Public | Event           | KeyDown()        | KeyCode As Integer, Shift As Integer  | -         |
| Public | Event           | KeyPress()       | KeyAscii As Integer   | -         |
| Public | Event           | KeyUp()          | KeyCode As Integer, Shift As Integer  | -         |
| Public | Event           | MouseDown()      | -   | -         |
| Public | Event           | MouseMove()      | -   | -         |
| Public | Event           | MouseUp()        | Button As Integer, Shift As Integer, x As Single, Y As Single   | -         |
| Public | Event           | ProgressStatus() | PercentCompleted As Integer, EventDescription As String   | -         |
| Public | Event           | Resize()         | -   | -         |
| Public | Event           | SendComplete()   | -   | -         |

## - Properties -

- About Property
- Appearance Property
- BackColor Property
- BorderStyle Property
- CheckSMCecho Property
- CommSettings Property
- CommType Property
- DTREnable Property
- EchoOn Property
- Enabled Property
- Font Property
- FontBold Property
- FontItalic Property
- FontName Property
- FontSize Property
- FontStrikethru Property
- FontUnderline Property
- ForeColor Property
- Handshaking Property
- hwnd Property
- MouseIcon Property
- MousePointer Property
- OnlineStatus Property
- Port Property
- Protocol Property
- RemoteHost Property
- RemotePort Property
- ReplyTimeout Property
- RTSEnable Property
- SellLength Property
- SelStart Property
- SelText Property
- SerialUpdateRate Property
- Text Property
- ToolTipText Property

- Methods -

- AddText Method
- CaptureFilename Method
- Connect Method
- Copy Method
- Disconnect Method
- Download Method
- Download\_new Method
- DownloadArray Method
- GetArraySize Method
- GetAxisRecord Method
- GetEthernetHandleUsage Method
- GetLocalIP Method
- GetMyHandleOnSMC Method
- GetOccuranceTotal Method
- GetProgramLines Method
- GetSerialNumber Method
- getSMCMode Method
- GetSMCRecord Method
- Output Method
- Paste Method
- Refresh Method
- SendCommand Method
- SendCommand2 Method
- SetFocus2 Method
- UpgradeSMCfirmware Method
- Upload Method
- UploadArray Method

- Events -

- Click Event
- ConnectError Event
- DbClick Event
- KeyDown Event
- KeyPress Event
- KeyUp Event
- MouseDown Event
- MouseMove Event
- MouseUp Event
- ProgressStatus Event
- PropertyOutOfRange Event
- Resize Event

- Error Constants -

| <b>Error</b>     | <b>Code</b> | <b>Description</b>  |
|------------------|-------------|---|
| cErrComTime      | 1           | Communication Timeout. Communications was silent for the ReplyTimeout interval before the command was acknowledged. |
| cErrTC1ret       | 2           |   |
| cErrComErr       | 3           |   |
| cErrNoHandle     | 4           |   |
| cErrSerrHandle   | 5           |   |
| cErrDLFail       | 6           | Program download failed.  |
| cErrDLLabel      | 7           | Program download failed because of a duplicate label or too many labels.  |
| cErrDLLine       | 8           | Program download failed because a program line was too long.  |
| cErrComEcho      | 9           |   |
| cErrComCmdVerify | 10          |   |
| cErrComQDErr     | 11          |   |
| cErrBadHandle    | 12          |   |
| cErrULFail       | 13          | Program upload failed.  |
| cErrNoAxis       | 14          |   |
| cErrAxRec        | 15          |   |
|                  |             |   |

- Enumerations -

- CheckSMEchoType Enumeration
- CommTypes Enumeration



## Custom Properties

This section contains reference information for each of the custom properties supported by the SMCComm Driver.

- CheckSMCecho
- CheckSMCechoType
- CommSettings
- CommType
- DTREnable
- EchoOn
- Handshaking
- MyIP
- OnLineStatus
- Port
- Protocol
- RemoteHost
- RemoteIP
- RemoteProtocol
- ReplyTimeOut
- RTSEnable
- SelLength
- SelStart
- SelText

# CheckSMCecho

## (Custom Property)

- **Purpose:**  
This property configures the control to either: detect echo automatically(Auto), check for echo to be on(SMCechoOn), or ignore echo(SMCechoOff).

- **Syntax:**  
`Terminal11.CheckSMCecho = 1`

- **Design Time Interface:**  
Drop Down List Box

- **Data Type:**  
Integer. The choices are enumerated as CheckSMCEchoType. The default value is Auto.

| Choice     | Value |
|------------|-------|
| Auto       | 0     |
| SMCechoOn  | 1     |
| SMCechoOff | 2     |

- **Applicability**

|          |                                     |
|----------|-------------------------------------|
| Serial   | Yes                                 |
| Ethernet | No, SMC does not echo over Ethernet |

- **Call Tree**  
`CheckSMCecho [Get]`  
`CheckSMCecho [Let]`
  - CheckRange
  - PropertyOutOfRange

- **Related Information:**  
The Property *ReplyTimeout* is the amount of time the driver will wait for an echo.

Auto: The control will check the response string from the SMC controller and determine if the original command exists in the return string. This is the echo, and the driver will detect this condition. This assures that the original message was received by the SMC controller exactly. This check will be performed every transmission and slow down communications when there is no echo. If the controller is set for echo (EO Command) the delay will be minimal. During no echo conditions the delay will equal the transmission timeout property called *ReplyTimeout*.

SMCEchoOn: The driver will require that the response string from the SMC controller contains the original message. If the original message is not found, then communications is assumed to have failed, and a message retry is automatically attempted. The controller must be in an EO1 (echo on) state.

SMCEchoOff: The driver will not check the response string for the original message, and therefore never attempt a retry caused by a bad transmission.

- **Example**

```
Terminal1.CheckSMCEcho = 0
```

- **Error Event**

If the CheckSMCEcho property is set to a value that is out of range the PropertyOutOfRange event will be raised.

# CommSettings

(Custom Property)

- **Purpose:**

Sets/returns the baud rate, parity, data bits, and stop bit parameters.

- **Syntax:**

```
Terminal1.CommSettings = "19200,n,8,1"
```

- **Design Time Interface:**

String Entry

- **Data Type:**

String of four parameters. The default is "19200,n,8,1"

| Parameter | Description | Value                           |
|-----------|-------------|---------------------------------|
| 1         | Baud Rate   | 2400, 4800, 9600, 19200, 38400  |
| 2         | Parity      | must be "n" for SMC controllers |
| 3         | Data Bits   | must be "8" for SMC controllers |
| 4         | Stop Bits   | must be "1" for SMC controllers |

- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | No  |

- **Call Tree**

```
CommSettings [Get]  
CommSettings [Let]
```

- **Related Information:**

This property is referenced by the *Connect* method.

The SMC2000 can connect at any of the listed baud rates provided that the controllers internal DIP switches are configured properly. 19200 is the default baud rate. The switches configure port one on the SMC2000 only. Port two's baud rate is configured by software.

The SMC3010/4000 can connect at either 9600 or 19200. 19200 is the default baud rate, and is set by DIP switch on the controller board.

- **Example**

```
Terminal1.Port = 1  
Terminal1.CommType = serial  
Terminal1.CommSettings = "19200,n,8,1"
```

Terminal1.Connect

- **Error Events**

**Currently there are no error events for CommSettings. Should have error if baud rate is out of range, and if last three parameters are not n,8,1 or not take the last three parameters as part of the property. Just add them to the baud rate the user enters.**

# CommType

## (Custom Property)

- **Purpose:**  
This property configures the control to direct communications to either the serial port or an Ethernet connection.

- **Syntax:**  
`Terminal1.CommType = 1`

- **Design Time Interface:**  
Drop Down List Box

- **Data Type:**  
Integer. The default is Serial.

| Choice   | Value |
|----------|-------|
| Serial   | 0     |
| Ethernet | 1     |

- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | Yes |

- **Call Tree**  
`CommType [Let]`  
- CheckRange  
-- PropertyOutOfRange  
`CommType [Get]`

- **Related Information:**  
This property is referenced by the *Connect* method.

- **Example**  
`Terminal1.CommType = Ethernet`  
`Terminal1.RemoteHost = TempS`  
`Msg = "IP: " & TempS`  
`StatusBar.Panels(2).ToolTipText = " PCs IP: " & Terminal1.MyIP`  
`HostAdress.Text = TempS`

- **Error Event**  
None

- **Raised Event**  
PropertyOutOfRange

# DTREnable

## (Custom Property)

- **Purpose:**  
Determines whether to enable the Data Terminal Ready line during communications.
- **Syntax:**  
`Terminal1.DTREnable=true`
- **Design Time Interface:**  
Drop Down List Box
- **Data Type:**  
Select True or False. The default is true.

- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | No  |

- **Call Tree**
  - `DTREnable [Get]`
  - `DTREnable [Let]`
    - CheckRange
    - PropertyOutOfRange
- **Example**
- **Error Event**  
None
- **Raised Event**  
PropertyOutOfRange

# EchoOn

## (Custom Property)

- **Purpose:**  
Sets/Returns if the terminal window should echo the characters sent and received from the controller.
- **Syntax:**  
`Terminal1.DTREnable=true`
- **Design Time Interface:**  
Drop Down List Box

- **Data Type:**  
Select True or False. The default is True.

- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | Yes |

- **Call Tree**
  - EchoOn [Get]**
  - EchoOn [Let]**
    - CheckRange
    - PropertyOutOfRange

- **Example**

```
If CommTypeProperty = Ethernet Then
  Winsock.SendData StringOut
  If EchoOn Then AddText StringOut
Else
  If MSComm.OutBufferCount > 0 Then
    MsgBox "here in output"
    MsgBox "in= " & MSComm.InBufferCount
    MsgBox "out= " & MSComm.OutBufferCount
    MsgBox "hds= " & MSComm.Handshaking
  End If
  MSComm.Output = StringOut
End If
```

- **Error Event**  
None

- **Raised Event**  
PropertyOutOfRange



# Handshaking

## (Custom Property)

- **Purpose:**  
Sets or returns the serial port handshaking protocol.

- **Syntax:**  
`Terminal1.Handshaking=comRTS`

- **Design Time Interface:**  
Drop Down List Box

- **Data Type:**  
Integer. The default is comRTS.

| Value | Description |
|-------|-------------|
| 0     | comNone     |
| 2     | comRTS      |

- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | No  |

- **Call Tree**  
`Handshaking [Get]`  
`Handshaking [Let]`  
- PropertyOutOfRange

- **Related Information:**  
Hardware handshaking is the recommended communications method. Only comNone or comRTS are possible selections, the SMC controllers do not support software handshaking (XOnXOff)

- **Example**  
`If CommTypeProperty <> Ethernet Then`  
`Terminal1.Handshaking = comRTS`  
`End If`

- **Error Event**  
None

- **Raised Event**  
PropertyOutOfRange

# OnLineStatus

(Custom Property)

- **Purpose:**  
Read only property that indicates the connection status of the control.
- **Syntax:**  
If OnLineStatus=true then  
    MsgBox "Online!"  
Endif
- **Design Time Interface:**  
None.
- **Data Type:**  
Boolean. There is no default because this is a read only property.

- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | Yes |

- **Call Tree**  
OnlineStatus [Get]
- **Example**
- **Error Event**  
None.

# Port

## (Custom Property)

- **Purpose:**  
Sets/returns the communications port number.

- **Syntax:**  
`Terminal1.Port=1`

- **Design Time Interface:**  
Integer. The default is 1.

- **Data Type:**  
Integer. Typically the computer ports are numbered 1 through 4.

- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | No  |

- **Call Tree**  
Port [Get]  
Port [Let]

- **Example**

```
Dim Port As Integer
Port = 0
While ((Port < 1) Or (Port > 7))
    TempS = InputBox("Enter Com Port Number (1 to 4) ", "Port",
    "1")
    If TempS = "" Then
        Form1.Enabled = True
        Exit Sub
    End If
    Port = Val(TempS)
Wend
Terminal1.Port = Port
```

- **Error Event**  
None

- **Raised Event**  
None

# Protocol

## (Custom Property)

- **Purpose:**  
Sets / Returns the protocol to use for communications over Ethernet.

- **Syntax:**  
`Terminal1.Protocol=0`

- **Design Time Interface:**  
Drop Down List Box

- **Data Type:**  
Integer. The default is `sckTCPProtocol`.

| Value | Description    |
|-------|----------------|
| 0     | sckTCPProtocol |
| 1     | sckUDPProtocol |

- **Applicability**

|          |     |
|----------|-----|
| Serial   | No  |
| Ethernet | Yes |

- **Call Tree**  
`Protocol [Get]`  
`Protocol [Let]`
  - `CheckRange`
  - `PropertyOutOfRange`

- **Example**

- **Error Event**  
None

- **Raised Event**  
`PropertyOutOfRange`

## RemoteHost

### (Custom Property)

- **Purpose:**  
Returns/Sets a name used to identify the remote SMC Controller. This can be either the IP address of the device or an alias.
- **Syntax:**  
`Terminal1.RemoteHost="192.168.3.100"`
- **Design Time Interface:**  
Text Entry

- **Data Type:**  
String. The default is an empty string.

- **Applicability**

|          |     |
|----------|-----|
| Serial   | No  |
| Ethernet | Yes |

- **Call Tree**  
`RemoteHost [Get]`  
`RemoteHost [Let]`  
- PropertyOutOfRange

- **Example**

```
TempS = InputBox("Enter an IP address ", "IP", HostAddress.Text)
If TempS = "" Then
    Form1.Enabled = True
    Exit Sub
End If
Terminal1.CommType = Ethernet
Terminal1.RemoteHost = TempS
```

- **Error Event**  
None
- **Raised Event**  
PropertyOutOfRange

# ReplyTimeout

(Custom Property)

- **Purpose:**  
Returns or sets the maximum allowable time for the application to wait for the SMC controller start responding to a command. This is the maximum amount of silent time that can exist while waiting for the controller to complete it's response.

- **Syntax:**  
`Terminal11.ReplyTimeout = 1000`

- **Design Time Interface:**

Text entry.

- **Data Type:**  
Long. The default is 300 milliseconds.

- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | Yes |

- **Related Information:**  
ReplyTimeout is in milliseconds

- **Example**  
`Terminal11.ReplyTimeout = 400`

- **Error Event**  
Currently there is no error event for ReplyTimeout.

## RTSEnable

(Custom Property)

- **Purpose:**  
Determines whether to enable the Request To Send line.

- **Syntax:**  
`Terminal1.RTSEnable = True`

- **Design Time Interface:**  
True / False drop down box.

- **Data Type:**  
Boolean. The default is true.

- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | No  |

- **Related Information:**  
This property is only necessary if you wish to manually set the hardware handshake line.

- **Example**

- **Error Events**  
Currently there are no error events.

# SelLength

(Custom Property)

- **Purpose:**  
Returns/sets the number of characters selected in the terminal window.
- **Syntax:**  
`Terminal1.SelLength=25`
- **Design Time Interface:**  
Text Entry
- **Data Type:**  
Integer. The default is 0.
- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | Yes |

- **Related Information:**
- **Example**
- **Error Events**  
Currently there are no error events for SelLength.



# SelStart

## (Custom Property)

- **Purpose:**  
Returns/sets the start point of characters selected in the terminal window.
- **Syntax:**  
`Terminal1.SelStart=25`
- **Design Time Interface:**  
Text Entry
- **Data Type:**  
Integer. Default value is zero.

- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | Yes |

- **Related Information:**
- **Example**
- **Error Events**

## SelText

(Custom Property)

- **Purpose:**  
Returns/sets the string containing the currently selected text.
- **Syntax:**  
`Terminal1.SelStart=25`
- **Design Time Interface:**  
Text Entry
- **Data Type:**  
Integer. The default is an empty string

- **Applicability**

|          |     |
|----------|-----|
| Serial   | Yes |
| Ethernet | Yes |

- **Related Information:**

- **Example**

```
Select Case Index
  Case 0 'Copy Selected Text
    Clipboard.Clear
    Clipboard.SetText Terminal1.SelText
  Case 1 'Copy All Text
    Clipboard.Clear
    Clipboard.SetText Terminal1.Text
  Case 2 'Paste Text
    Terminal1.Output Clipboard.GetText
  Case 7 'Clear Window
    Terminal1.Text = ""
End Select
```

- **Error Events**

Currently there are no error events for SelText.

## Methods

This section contains reference information for each of the methods supported by the SMCComm Driver.

- AddText
- CaptureFileName
- Connect
- Copy
- Disconnect
- Download
- DownloadArray
- GetArraySize
- GetAxisRecord
- GetEthernetHandleUsage
- GetLocalIP
- GetMyHandleOnSMC
- GetOccuranceTotal
- GetProgramLines
- GetSerialNumber
- GetSMCRecord
- LinesInController
- SendCommand
- Upload
- UploadArray

# AddText

(Method)

- **Purpose:**  
Adds text to the terminal window
- **Syntax:**  
`Terminal1.AddText(Str as string)`
- **Data Type:**  
String. The default is an empty string.
- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**
- **Example**
- **Error Event**

A standard Vb Error code will be raised in the event of an error.

# CaptureFileName

(Method)

- **Purpose:**  
Specify a full path filename which will be the location where all communications that occurs between the SMC controller and the computer will be stored.
- **Syntax:**  
`Terminal1.CaptureFilename (LogfileName as String) as String`
- **Data Type:**  
String. The default is an empty string.
- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**
  - If a file already exists with the same name, the terminal characters will be appended to it.
  - If communications is already being logged to the specified file, the method exits, leaving the log function enabled.
  - If an empty string filename is specified, logging stops.
- **Example**
- **Error Event**  
The CaptureFileName method has an error handler that will pass back error descriptions of standard file access errors.

# Connect

(Method)

- **Purpose:**  
To connect the computer to an SMC controller via either serial port or Ethernet.
- **Syntax:**  
`Terminal1.Connect`
- **Data Type:**  
None
- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**

This subroutine uses the following properties to make a connection:

| Data                          | Usage   |
|-------------------------------|---|
| <code>CommTypeProperty</code> | Preset to either 1 for Serial or 2 for Ethernet |

This subroutine sets the following properties during execution:

| Data                  | Usage              |
|-----------------------|--------------------|
| <code>m_EchoOn</code> | True if Successful |
| <code>Enabled</code>  | True if Successful |
| <code>OnLine</code>   | True if Successful |

This subroutine immediately exits without error if the OCX is already `Online` with a controller.

- **Example**
- **Error Event**  
If a connection could not be made, an error event is raised with the `cErrComTime (1)` error code.

# Disconnect

(Method)

- **Purpose:**  
To disconnect the computer from an SMC controller via either serial port or Ethernet.
- **Syntax:**  
`Terminal1.Disconnect()`
- **Data Type:**  
Returns a Long. Zero if successful, otherwise it returns the error code.
- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**

This subroutine uses the following properties to make a connection:

| Data             | Usage   |
|------------------|---|
| CommTypeProperty | Preset to either 1 for Serial or 2 for Ethernet |

This subroutine sets the following properties during execution:

| Data    | Usage               |
|---------|---------------------|
| Enabled | False if Successful |
| OnLine  | False if Successful |

- **Example**
- **Error Event**

This method returns a zero if successful, otherwise it returns the error code. Only standard VB error codes will be returned.

# GetArraySize

(Method)

- **Purpose:**  
To get the defined size of an existing array in the SMCcontroller.
- **Syntax:**  
`Terminal1.GetArraySize(ArrayName as string) as integer`
- **Data Type:**  
ArrayName as String. Returns an integer.

- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**

This method uses the LA command in the controller to obtain a listing of all the defined arrays. If the array does not exist in the controller, the method returns a size of zero.

- **Example**

```
UploadString = ""  
aSize = GetArraySize(ArrayName)  
If aSize = 0 Then  
    MsgBox "The array < " & ArrayName & " > is not defined in the  
        controller!"  
End If
```



# GetAxisRecord

(Method)

- **Purpose:**  
Returns the information pertaining to an axis in the form of a record.
- **Syntax:**  
`Terminal1.GetAxisRecord(AxisLetter as String) as AxisInfo`

- **Data Type:**  
AxisInfo is a record.

```
Type AxisInfo
  Header1 As Byte
  Header2 As Byte
  Header3 As Byte
  Header4 As Byte
  Status As Long
  Switches As Byte
  StopCode As Byte
  PosCommanded As Long
  PosActual As Long
  PosError As Long
  PosAux As Long
  Velocity As Long
  Torque As Integer
  AnalogIN1 As Integer
  AllAsString As String
End Type
```

- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | No  |
| Ethernet      | Yes | SMC3010 only | Yes |

- **Related Information:**
- **Example**
- **Error Event**

# GetEthernetHandleUsage

(Method)

- **Purpose:**
- **Syntax:**  
`Terminal1.GetEthernetHandleUsage(Handle as String) as HandleInfo`
- **Data Type:**  
HandleInfo is a record.

```
Type HandleInfo
  ConnectionType As Long 'IHx2
  ConnectionTypeDescr As String
  SlavePort As Long
  IPAddressS As String
  IPAddressH As String
End Type
```

- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | No  |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**
- **Example**
- **Error Event**

# GetLocalIP

(Method)

- **Purpose:**  
Returns the IP Address of the computer.

- **Syntax:**  
`Terminal1.GetLocalIP`

- **Data Type:**  
Returns a string.

- **Applicability**

|          |  |
|----------|--|
| Serial   | Yes, if the computer has an Ethernet adapter card. |
| Ethernet | Yes  |

- **Related Information:**  
For diagnostic purposes. It is recommended to set the first three octets of the IP Address the same as the Computer's address. An invalid address, or a loopback address on the PC side will signal that the PC is not properly configured to communicate over the Ethernet. Please go to network settings and define a valid IP address...

- **Example**  
`ComputerIP = Terminal1.GetLocalIP`  
`MsgBox "Computer IP Address is " & ComputerIP`

- **Error Event**

# GetMyHandleOnSMC

(Method)

- **Purpose:**  
To determine which of the SMC3010/4000's handles the computer is connected. This command is only applicable if the computer is connected to a SMC3010/4000.
- **Syntax:**  
`MyHandle= Terminal1.GetMyHandleOnSMC`
- **Data Type:**  
Returns a string.

- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | No  | SMC2000      | No  |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**

This method uses the SMC command \_IH to determine the handle.

- **Example**

```
Terminal1.Connect
  If x = 0 Then
    If Terminal1.CommType = Ethernet Then Msg = Msg & "
      Handle " & Terminal1.GetMyHandleOnSMC
    StatusBar.Panels(1).Text = "Connected to " & Msg
  EndIf
EndIF
```

- **Error Event**

Possible Errors:

| Error Code | Error Name     | Description                                     |
|------------|----------------|---|
| 4          | CErrNoHandle   | "Can't Find Handle."                            |
| 5          | CErrSerrHandle | "Serial connection does not require a handle."  |
| ???        | ???            | "Could not get a valid IP address for this PC." |

# GetProgramLines

(Method)

- **Purpose:**  
Returns the number of program lines of the program in the controller.
- **Syntax:**  
`Lines = GetProgramLines`
- **Data Type:**  
Returns an integer. There is no default. The minimum is 0 and the maximum is 999.
- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**
- **Example**  
`Lines = GetProgramLines`
- **Error Event**

Possible Errors:

| Error Code | Error Name | Description                                  |
|------------|------------|--|
| 3          | CErrComErr | "Could not get the number of program lines!" |

# GetSerialNumber

(Method)

- **Purpose:**  
Returns the serial number of the controller.
- **Syntax:**  
`Terminal1.GetSerialNumber`
- **Data Type:**  
Returns as string
- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**  
This method uses the command MG \_BN to obtain the serial number of the controller.
- **Example**
- **Error Event**

# GetSMCRecord

(Method)

- **Purpose:**  
Returns a record of data containing detailed information about the controller.
- **Syntax:**  
Terminal1.GetSMCRecord() as SMCInfo
- **Data Type:**  
Returns a record.

```
Type SMCInfo
  Header1 As Byte
  Header2 As Byte
  Header3 As Byte
  Header4 As Byte
  SMCTime As Long
  InpByte(9) As Byte
  OutByte(9) As Byte
  TCCode As Byte
  TBcode As Byte
  SplSegCount As Long
  SplMovStat As Long
  SplDistance As Long
  NotDefined1 As Long
  NotDefined2 As Long
  NotDefined3 As Long
  AxisN As Integer
  Axis(7) As AxisInfo
  AllAsString As String
End Type
```

- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | No  |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**  
This method uses the command MG \_BN to obtain the serial number of the controller.
- **Example**
- **Error Event**

# Download

(Method)

- **Purpose:**

Downloads an SMC program to a controller.

- **Syntax:**

Download(SMCFile As String, Optional Verify As Boolean = False)

- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**

The program file must contain the end of line character <CR> or <CR> <LF> or the controller will reject the program because of a line being too long. No error checking is done to the program in this routine concerning it's validity, however, one of the three errors might occur as a result of downloading:

- Line Too long
- Duplicate Label or line number
- Too many lines

- **Example**

```

Lines = GetProgramLines
UploadString = ""
For L = 0 To Lines - 1
    ProgLine = SendCommand("LS" & L & ", " & L)
    SpaceLoc = InStr(2, ProgLine, Chr(32), vbTextCompare)
    UploadString = UploadString & Right$(ProgLine,
        Len(ProgLine) - SpaceLoc) & vbCrLf
    RaiseEvent ProgressStatus(L / Lines * 100, "Uploading
        Program.")
Next
RaiseEvent ProgressStatus(0, "Upload Complete.")
Upload = UploadString
    
```

- **Error Event**

Possible Errors:

| Error Code | Error Name | Description   |
|------------|------------|---|
| 6          | cErrDLFail | "Command Error after download "                     |
| 6          | cErrDLFail | "Download failed because program line is too long." |
| 6          | cErrDLFail | "Uploaded & Downloaded versions do not match!"      |



# DownloadArray

(Method)

- **Purpose:**  
Downloads an array of data to a controller.

- **Syntax:**  

```
Public Sub DownloadArray(ArrayName As String, ArrayFile As String, Delimiter As String, StartIdx As Integer, EndIdx As Integer, Optional Verify As Boolean = False)
```

- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**

Arrayname is the name of an array that exists in the SMC controller.  
ArrayFile is the data to store in the array.  
Delimiter is the character or characters that separate the data in the ArrayFile string  
StartIdx is the first element to start entering data. (0 based system)  
Endidx is the last element to enter data.(0 based system)  
Verify will cause the OCX to upload the array contents and verify that they match the original data.

- **Example**

```
Terminal1.DownloadArray("Demo", ArrayData, vbCR, 1, 1000)
```

- **Error Event**

Possible Errors:

| Error Code | Error Name   | Description   |
|------------|--------------|---|
| 11         | cErrComQDErr | "Command Error after download "                     |
| 11         | cErrComQDErr | "Download failed because program line is too long." |
| 11         | cErrComQDErr | "Uploaded & Downloaded versions do not match!"      |

# SendCommand

(Method)

- **Purpose:**  
Send commands and retrieve replies from the controller.
- **Syntax:**  
`Terminal1.SendCommand(CommandToSend As String) As String`
- **Data Type:**  
Returns a string
- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**  
SendCommand will automatically add a <CR> to the *CommandToSend* string. SendCommand uses a number of properties, including CheckSMCEcho to determine if an echo should be received. If an exact echo is not received, the method automatically retries after flushing the controller buffer with the "\n" character. Four silent retries will be attempted before generating an error. The Method also checks for a valid response signature, which typically consists of a ":" and carriage return <CR>.

SendCommand will not add a <CR> to the CommandToSend string if the last character is a control character. Typically control characters are used to do special things, like terminate program downloads. The control character for this is <CTRL> Z. If you send a string such as <CTRL> R <CTRL> V to obtain the firmware revision, add the <CR> to the CommandToSend string.

If a bad command is sent to the controller, i.e. a string that is not a valid command, an error will be returned indicating the controllers error code, which is obtained by sending the "TC1" command.

- **Example**

```
Select Case Index
Case 0 'SH
Terminal1.SendCommand ("SH")
Case 1 'MO
Terminal1.SendCommand ("MO")
Case 3 'Lines is a program ?
MsgBox Terminal1.LinesInController
Case 7 'XQ
Terminal1.SendCommand ("XQ")
Case 8 'ST
Terminal1.SendCommand ("ST")
Case 9 'HX
Terminal1.SendCommand ("HX")

Etc....
```

- **Error Event**

Possible Errors:

| <b>Error Code</b> | <b>Error Name</b> | <b>Description</b>   |
|-------------------|-------------------|--|
| 1                 | cErrComTime       | "Communication Timed Out sending <Command> to the controller." |
| 2                 | cErrTC1ret        | "Error was received while trying to send <Command>"            |
| 3                 | cErrComErr        | "Command Error sending <Command> to the controller."           |
| 11111             | ???               | "Send retry #" & Str\$(Retry))                                 |
| 11113             | ???               | "Command not valid in Offline mode."                           |

# Upload

(Method)

- **Purpose:**  
Uploads an application program from the controller.
- **Syntax:**  
`Terminal1.Upload() As String`
- **Data Type:**  
Returns a string

- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**

Upload returns the entire program from the controller, without program lines. The program lines will be separated by the vbCRLF bytes. Use VB's Split command to break the uploaded program into lines if necessary.

If a bad command is sent to the controller, i.e. a string that is not a valid command, an error will be returned indicating the controllers error code, which is obtained by sending the "TC1" command.

- **Example**

```
Form2.Caption = "Output from UL command"  
Form2.txtOutput = Terminal1.Upload  
Form2.Show true
```

- **Error Event**

# UploadArray

(Method)

- **Purpose:**  
Uploads an array from the controller.
- **Syntax:**  
*UploadArray(ArrayName As String, StartIdx As Integer, EndIdx As Integer) As String*
- **Data Type:**  
Returns a string containing the array data separated by commas.
- **Applicability**

| Communication |     |  | Product      |     |
|---------------|-----|--|--------------|-----|
| Serial        | Yes |  | SMC2000      | Yes |
| Ethernet      | Yes |  | SMC3010/4000 | Yes |

- **Related Information:**
- **Example**
- **Error Event**

## - Events -

**ConnectError**  
**ProgressStatus**

# ConnectError

(Event)

- **Purpose:**  
Executed when there is an Connection Error
- **Syntax:**  
`ConnectError(ErrorCode As Integer, ErrorDescription As String)`
- **Data Type:**  
Subroutine
- **Applicability**

| Communication |     |  | Product      |     |
|---------------|-----|--|--------------|-----|
| Serial        | Yes |  | SMC2000      | Yes |
| Ethernet      | Yes |  | SMC3010/4000 | Yes |

- **Related Information:**  
Methods that might invoke the ConnectError Event:
  - Conect
  - SendCommand
  - GetSMRecord
  - GetAxisRecord
- **Example**
- **Error Event**

# ProgressStatus

(Event)

- **Purpose:**  
Executed when there is a change in progress of the Upload, Download, UploadArray, or DownloadArray methods.
- **Syntax:**  
ProgressStatus(PercentCompleted As Integer, EventDescription As String)
- **Data Type:**  
Subroutine
- **Applicability**

| Communication |     | Product      |     |
|---------------|-----|--------------|-----|
| Serial        | Yes | SMC2000      | Yes |
| Ethernet      | Yes | SMC3010/4000 | Yes |

- **Related Information:**  
Methods that update the ProgressStatus Event:  
Download  
UpLoad  
DownLoadArray  
UploadArray
- **Example**
- **Error Event**



## Procedure call tree - SMCComm

### UserControl Terminal

```
Resize
MouseUp
MouseMove
MouseDown
DbClick
Click
KeyUp
KeyPress
KeyDown
ConnectError
ProgressStatus
PropertyOutOfRange
About [Let]
About [Get]
CaptureDebugFile
- Output
  -- EchoOn [Get]
  -- AddText
CheckRange
- PropertyOutOfRange
GetArraySize
- SendCommand
  -- ConnectError
  -- Output
    --- EchoOn [Get]
    --- AddText
  -- ManualOnComm
    --- AddText
    --- ConnectError
    --- Disconnect
GetLocalIP
getSMCMode
- ManualOnComm
  -- AddText
  -- ConnectError
  -- Disconnect
- Output
  -- EchoOn [Get]
  -- AddText
ManualOnComm
- AddText
- ConnectError
- Disconnect
OnComm_Old
- AddText
- ConnectError
- Disconnect
Refresh
SerialUpdateRate [Get]
SerialUpdateRate [Let]
```

```

- CheckRange
  -- PropertyOutOfRange
SignedLong
SignedInteger
UpgradeSMCfirmware
- CheckSMCecho [Let]
  -- CheckRange
    --- PropertyOutOfRange
- Output
  -- EchoOn [Get]
  -- AddText
- ProgressStatus
UploadArray
- GetArraySize
  -- SendCommand
    --- ConnectError
    --- Output
      ---- EchoOn [Get]
      ---- AddText
    --- ManualOnComm
      ---- AddText
      ---- ConnectError
      ---- Disconnect
- SendCommand
  -- ConnectError
  -- Output
    --- EchoOn [Get]
    --- AddText
  -- ManualOnComm
    --- AddText
    --- ConnectError
    --- Disconnect
- ProgressStatus
GetSMRecord
- ConnectError
- Output
  -- EchoOn [Get]
  -- AddText
- ManualOnComm
  -- AddText
  -- ConnectError
  -- Disconnect
- SignedLong
- SignedInteger
GetAxisRecord
- ConnectError
- Output
  -- EchoOn [Get]
  -- AddText
- ManualOnComm
  -- AddText
  -- ConnectError
  -- Disconnect
- SignedLong
- SignedInteger
Upload
- GetProgramLines

```

```

-- SendCommand
  --- ConnectError
  --- Output
    ---- EchoOn [Get]
    ---- AddText
  --- ManualOnComm
    ---- AddText
    ---- ConnectError
    ---- Disconnect
-- Output
  --- EchoOn [Get]
  --- AddText
- SendCommand
-- ConnectError
-- Output
  --- EchoOn [Get]
  --- AddText
-- ManualOnComm
  --- AddText
  --- ConnectError
  --- Disconnect
- ProgressStatus
GetProgramLines
- SendCommand
  -- ConnectError
  -- Output
    --- EchoOn [Get]
    --- AddText
  -- ManualOnComm
    --- AddText
    --- ConnectError
    --- Disconnect
- Output
  -- EchoOn [Get]
  -- AddText
CaptureFilename
- Output
  -- EchoOn [Get]
  -- AddText
Enabled [Let]
Enabled [Get]
GetEthernetHandleUsage
- SendCommand
  -- ConnectError
  -- Output
    --- EchoOn [Get]
    --- AddText
  -- ManualOnComm
    --- AddText
    --- ConnectError
    --- Disconnect
GetOccuranceTotal
AddText
GetSerialNumber
- SendCommand
  -- ConnectError
  -- Output

```

```

    --- EchoOn [Get]
    --- AddText
-- ManualOnComm
    --- AddText
    --- ConnectError
    --- Disconnect
GetMyHandleOnSMC
- SendCommand
-- ConnectError
-- Output
    --- EchoOn [Get]
    --- AddText
-- ManualOnComm
    --- AddText
    --- ConnectError
    --- Disconnect
ReplyTimeout [Let]
- CheckRange
-- PropertyOutOfRange
ReplyTimeout [Get]
SendCommand
- ConnectError
- Output
-- EchoOn [Get]
-- AddText
- ManualOnComm
-- AddText
-- ConnectError
-- Disconnect
SetFocus2
Output
- EchoOn [Get]
- AddText
MSComm_OnComm
- ConnectError
- AddText
- Disconnect
terminal_Change
terminal_Click
- Click
terminal_DblClick
- DblClick
terminal_KeyDown
- KeyDown
terminal_KeyPress
- KeyPress
terminal_MouseDown
- MouseDown
terminal_MouseMove
- MouseMove
terminal_MouseUp
- MouseUp
Timer1_Timer
tmrCommUpdate_Timer
- ManualOnComm
-- AddText
-- ConnectError

```

```

-- Disconnect
UserControl_Initialize
UserControl_MouseDown
- MouseDown
UserControl_MouseMove
- MouseMove
UserControl_MouseUp
- MouseUp
UserControl_ReadProperties
- MouseIcon [Set]
UserControl_Resize
- Resize
UserControl_Terminate
UserControl_WriteProperties
- MouseIcon [Get]
ToolTipText [Get]
ToolTipText [Let]
Text [Get]
Text [Let]
SelText [Get]
SelText [Let]
SelStart [Get]
SelStart [Let]
- CheckRange
-- PropertyOutOfRange
SelLength [Get]
SelLength [Let]
- CheckRange
-- PropertyOutOfRange
ForeColor [Get]
ForeColor [Let]
FontUnderline [Get]
FontUnderline [Let]
FontStrikethru [Get]
FontStrikethru [Let]
FontSize [Get]
FontSize [Let]
FontName [Get]
FontName [Let]
FontItalic [Get]
FontItalic [Let]
FontBold [Get]
FontBold [Let]
Font [Get]
Font [Set]
BackColor [Get]
BackColor [Let]
Appearance [Get]
Appearance [Let]
BorderStyle [Get]
BorderStyle [Let]
MousePointer [Get]
MousePointer [Let]
MouseIcon [Get]
MouseIcon [Set]
terminal_KeyUp
- KeyUp

```

```

- AddText
UserControl_InitProperties
hwnd [Get]
RTSEnable [Get]
RTSEnable [Let]
- CheckRange
  -- PropertyOutOfRange
RemotePort [Get]
RemotePort [Let]
Winsock_Close
Winsock_Connect
Winsock_DataArrival
- AddText
Winsock_Error
- ConnectError
- Disconnect
Connect
Disconnect
Copy
Paste
Winsock_SendComplete
Winsock_SendProgress
Download
- SendCommand2
  -- Output
    --- EchoOn [Get]
    --- AddText
  -- ManualOnComm
    --- AddText
    --- ConnectError
    --- Disconnect
- ProgressStatus
- SendCommand
  -- ConnectError
  -- Output
    --- EchoOn [Get]
    --- AddText
  -- ManualOnComm
    --- AddText
    --- ConnectError
    --- Disconnect
DownloadArray
- SendCommand2
  -- Output
    --- EchoOn [Get]
    --- AddText
  -- ManualOnComm
    --- AddText
    --- ConnectError
    --- Disconnect
- GetArraySize
  -- SendCommand
    --- ConnectError
    --- Output
      ---- EchoOn [Get]
      ---- AddText
    --- ManualOnComm

```

```
        ---- AddText
        ---- ConnectError
        ---- Disconnect
- SendCommand
  -- ConnectError
  -- Output
    --- EchoOn [Get]
    --- AddText
  -- ManualOnComm
    --- AddText
    --- ConnectError
    --- Disconnect
- ProgressStatus
SendCommand2
- Output
  -- EchoOn [Get]
  -- AddText
- ManualOnComm
  -- AddText
  -- ConnectError
  -- Disconnect
```

## **Module Module1**

### **Form Form1**

**Text1\_DblClick**

*Project Analyzer 6.2.02 (12/10/2003) SMCComm v1.1.40*