CPS Plus – Data Acquisition Software – ActiveX interface

Overview

CPS Plus ActiveX interface enables communication with devices connected via RS232 COM ports and the devices with keyboard emulation using ActiveX technology.

Use of CPS Plus ActiveX interface makes it easy to access RS232 or devices with keyboard emulation from VBScript, Excel, MS Access using VBA/VBS, C# or any Windows program with support for ActiveX objects.

Example: collecting data from COM port into Access - VBA.

```
' Add CPSPlus.CComInterface to your database -> Click Tools -> References ->  and select "CPS Plus USB-RS232 COM Port Interface"


Public WithEvents CpsConn As CPSPlus.CComInterface

Private Sub CpsConn_DataReceived(ByVal COMPort As Long, ByVal FilteredData As String)

' Event fires when data is received from COM Port. First parameter will contain number of COM port sending data.
Second parameter is data read from COM port (filtered if any of CPS Plus filters is applied).
' COM1 to COM256 are reserved for actual COM ports while 257 and up are for devices with keyboard emulation!

End Sub

Private Sub Form_Load()

   ' First! Create instance of CPSPlus.CComInterface - this is required for all other functions/subs which use CpsConn
   Set CpsConn = New CPSPlus.CComInterface
End Sub

' Clean up
Private Sub Form_Unload(Cancel As Integer)
   Set CpsConn = Nothing
End Sub
```

For more details look at methods and properties which enable full control of devies connected via COM port or devices with keyboard emulation.

CPS Plus – Data Acquisition Software – ActiveX interface

Methods

Method: **WriteData**

Description: Write data to COM port

Paremeters:

COMPort – com port number - range 1-256

OutData – string data. Use curly brackets to specify ASCII code of characters that can't be entered via keyboard.

Example: write to COM4

```
CpsConn.WriteData 4, "Some Data{13}{10}"
```

Method: **WriteHexData**

Description: Write data to rs232 specified as pairs of hexadecimal chars

Paremeters:

COMPort – com port number - range 1-256

OutData – string data. Use two pair of hex characters for each byte.

Example: write to COM3

```
CpsConn.WriteHexData 3, "EE0A0D"
```

Method: **ExecSavedCommand**

Description: Execute command previously saved with CPS Plus command wizard.

Paremeters:

CommandName – string data. Name of existing command (alias) in CPS Plus.

Example:

```
CpsConn.ExecSavedCommand  "InitScale"
```

Method: **COMControl**

Description: Method for controlling COM ports.

Paremeters:

COMPort – com port number - range 1-256

ControlString– string data. One of valid control strings.

VBA Example: Change COM1 DTR line to ON-High

CpsConn.COMControl 1, " [SETDTR]"

Valid controls strings and descriptions:

[OPEN]	open connection to com port or capture device with keyboard emulation (RFID read, magnetic stripe reader, barcode scanner ...etc)
[CLOSE]	close connection
[RELOAD]	close and re-open connection
[FLUSH]	clear COM port buffers
[RESETPORT]	clear COM port buffers
[SETBREAK]	Suspends character transmission and places the transmission line in a break state until [CLRBREAK] control is called.
[CLRBREAK]	Restores character transmission and places the transmission line in a nonbreak state.
[SETXON]	Causes transmission to act as if an XON character has been received.
[SETXOFF]	Causes transmission to act as if an XOFF character has been received.
[SETRTS]	Sends the RTS (request-to-send) signal.
[SETDTR]	Sends the DTR (data-terminal-ready) signal.
[CLRRTS]	Clears the RTS (request-to-send) signal.
[CLRDTR]	Clears the DTR (data-terminal-ready) signal.

Note: If Handshaking/Flow control is enabled in COM port settings, it is an error for the application to adjust the line by using [CLRDTR/CLRRTS/SETDTR/SETRTS/ SETXON/ SETXOFF].

Properties

Property: GetCTS

Type: read-only Boolean

Description: Get current state of CTS (clear-to-send) signal.

Parameters:

COMPort – com port number - range 1-256

VBA Example: *Get current state of CTS line for COM5*

```
If CpsConn.GetCTS(5) = True Then …
```

Property: GetDSR

Type: read-only Boolean

Description: Get current state of DSR (data-set-ready) signal.

Parameters:

COMPort – com port number - range 1-256

VBA Example: *Get current state of DSR line for COM7*

```
If CpsConn.GetDSR(7) = True Then ...
```

Property: GetRLSD

Type: read-only Boolean

Description: Get current state of RLSD (receive-line-signal-detect) signal.

Parameters:

COMPort – com port number - range 1-256

VBA Example: *Get current state of RLSD for COM3*

```
If CpsConn.GetRLSD(3) = True Then ....
```


Property: GetRING

Type: read-only Boolean

Description: Get current state of the ring indicator signal.

Parameters:

COMPort – com port number - range 1-256

VBA Example: *Get current state of for COM1*

```
If CpsConn.GetRING(1) = True Then…
```

Property: SignalBREAK

Type: read-write Boolean

Description: Get/Set current state of Break Line

Set Parameters:

COMPort – com port number - range 1-256

NewState – boolean :

True -> Suspends character transmission and places the transmission line in a break state.

False -> Restores character transmission and places the transmission line in a nonbreak state.

If used as a getter return true if COM port hardware detected a break condition or false otherwise.

VBA Example:

```
If CpsConn.SignalBREAK(1) = True Then
    CpsConn.SignalBREAK(1) = False
End If
```

Property: SignalDTR

Type: write only Boolean

Description: Set current state of DTR line.

Parameters:

COMPort – com port number - range 1-256

VBA Example: *Set current state of DTR signal for COM3.*

```
CpsConn.SignalDTR(3) = False
```

Note: If Handshaking/Flow control is enabled in COM port settings, command is ignored.

Property: SignalRTS

Type: write only Boolean

Description: Set current state of *RTS* line.

Parameters:

COMPort – com port number - range 1-256

VBA Example: *Set current state of RTS signal for COM5.*

```
CpsConn.SignalRTS(5) = True
```

Note: If Handshaking/Flow control is enabled in COM port settings, command is ignored.

Property: GetDeviceType

Type: read-only - Numeric

Description: Return 1 for COM port or 2 for devices with keyboard emulation.

Parameters:

COMPort – port number - range [1-256 com ports] - [257 and up – Keyboards]

VBA Example:

```
If CpsConn.GetDeviceInfo(258) = 2 Then ' Keyboard Device - Else ' COM Port
```

Property: GetDeviceInfo

Type: read-only String

Description: return full device description.

Parameters:

COMPort – port number - range [1-256 com ports] - [257 and up – Keyboards]

VBA Example: *Get current device description for COM258*

```
MsgBox CpsConn.GetDeviceInfo(258)
```

Note: For COM ports – output will be com port number and manufacturer name (if available), while for devices with keyboard emulation it will typically include unique driver ID, vendor id, device number, and manufacturer name (if available).

Property: GetDeviceUniqueID

Type: read-only String

Description: return unique device identifier

Parameters:

COMPort – port number - range [1-256 com ports] - [257 and up – Keyboards]

VBA Example: *Get current*

```
MsgBox CpsConn.GetDeviceUniqueID(257)
```

Note: For COM ports – output will be COM1, COM2, COM3 … while for devices with keyboard emulation return value will be unique driver ID assigned by Windows (for example output may be: \\?\HID#VID_0B38&PID_0010&MI_01&Col02#8&206e85c&0&0001#{4d1e55b2-f16f-11cf-88cb-001111000030}).

CPS Plus – Data Acquisition Software – ActiveX interface

Events

Event: DataReceived

Description: Event that is triggered each time new data is received on COM port or device with keyboard emulation sends trigger character (end of transmission character).

Parameters:

COMPort – numeric value. First parameter will contain port number of device sending data.

FilteredData – string value. Second parameter contains actual value received from com port or sent by keyboard emulated device. Data will be filtered if any of CPS Plus filters is active.

Note: Multiple clients may subscribe to receive this event when new data is available.
