

Элемент управления Microsoft Comm Control 6.0

Элемент управления **Microsoft Comm Control 6.0** обеспечивает последовательной связью вашу программу с внешним миром, позволяя передачу и прием данных через последовательный порт.

Управляемая событиями связь – очень мощный метод обработки при взаимодействии с последовательным портом. Во многих ситуациях вам необходимо получать извещение в тот момент, когда происходит событие, такое как 'прибытие' символа или когда происходит изменение в Carrier Detect (CD) или REquest To Send (RTS-линии). Для перехвата и обработки этих событий используется событие **OnComm**. Событие **OnComm** также обнаруживает и обрабатывает ошибки связи.

Каждая коммуникационная связь управляемая вами использует один последовательный порт. Если вы должны обратиться к более чем одному COM – порту в вашей программе, вы должны будете использовать больше чем одно средство управления связи. Элемент управления **Microsoft Comm Control 6.0** имеет много важных свойств, но имеется несколько, с которыми вы должны быть обязательно знакомы (Таб. 1).

Таб. 1

Свойство	Описание
CommPort	Устанавливает и возвращает номер используемого COM-порта
Handshaking	Устанавливает и возвращает протокол подтверждения связи
Settings	Устанавливает и возвращает скорость в бодах, контроль по четности, биты данных, стоповые биты
PortOpen	Устанавливает и возвращает состояние порта связи, а также открывает и закрывает COM-порт
Input	Возвращает и удаляет символы из получающего буфера
InputLen	Устанавливает и возвращает число считываемых символов свойством Input из буфера
InBufferCount	Возвращает количество символов ждущих в получающем буфере
InBufferSize	Устанавливает и возвращает размер получающего буфера в байтах
Output	Записывает последовательность символов в передающий буфер
OutBufferCount	Возвращает число символов ждущих в передающем буфере
OutBufferSize	Устанавливает и возвращает размер передающего буфера в байтах
OnComm	Событие

Свойство **CommPort**

Свойство **CommPort** устанавливает и возвращает номер используемого COM-порта.

Синтаксис

MSComm.CommPort[= portNumber%]

Вы можете устанавливать portNumber равным любому числу лежащим в интервале от 1 до 99 (по умолчанию portNumber устанавливается равным 1). Однако элемент управления Microsoft Comm Control сгенерирует ошибку 68, когда вы попытаетесь открыть его свойством **PortOpen**, если такого номера COM-порта не существует.

Примечание. Свойство **CommPort** устанавливается перед открытием COM-порта. Тип данных – Integer

Свойство **Handshaking**

Свойство **Handshaking** устанавливает и возвращает протокол подтверждения связи.

Синтаксис

MSComm.Handshaking[= protocol%]

Подтверждение связи относится к внутреннему протоколу связи, в соответствии с которым данные перемещаются с аппаратного порта на получающий буфер. Когда символ

данных достигает последовательного порта, устройство связи должно переместить его в получающий буфер так, чтобы ваша программа могла считать его. Если этого не произойдет и ваша программа будет читать непосредственно из аппаратных средств, то может произойти потеря данных, потому что скорость поступления символов может быть очень велика. Протокол обеспечения связи обеспечивает, чтобы ваши данные не потерялись из-за переполнения буфера (когда данные поступают слишком быстро для устройства связи). Ниже в таблице (Таб. 2) перечислены типы используемых протоколов.

Таб. 2

Установка	Значение	Описание
ComNone	0	Нет подтверждения связи (Значение по умолчанию)
ComXOnXoff	1	XON/XOFF – подтверждение связи
ComRST	2	RTS/CTR – подтверждение связи
comRTSXOnXOff	3	RTS и XON/XOFF – подтверждение связи

Тип данных – Integer

Свойство **Settings**

Свойство **Settings** устанавливает и возвращает скорость в бодах, контроль по четности, биты данных, стоповые биты.

Синтаксис

MSComm.Settings[= paramString\$]

paramString\$ - состоит из четырех установок и имеет следующий формат – "BBBB, P, D, S", где:

BBBB - скорость в бодах (таб. 3);

P - контроль по четности (таб. 4);

D - число битов данных (таб. 5);

S - число стоповых битов (таб. 6);

Таб. 3

Установка скорости в бодах
9600 (по умолчанию)
14400
19200
38400
56000
115200
256000 (резервная)

Таб. 4

Установка	Описание
E	Четный
M	Метка
N	Нет (по умолчанию)
O	Нечетный
S	Пробел

Таб. 5

Установка	4	5	6	7	8 (по умолчанию)
-----------	---	---	---	---	------------------

Таб. 6

Установка	1 (по умолчанию)	1.5	2
-----------	------------------	-----	---

Тип данных – String

По умолчанию элемент управления **Microsoft Comm Control** установлен в "9600, N, 8, 1", т.е. устройство управления устанавливает порт для связи со скоростью 9600 бод, без проверки контроля четности, 8 бит данных и 1-го стопового бита.

Свойство **PortOpen**

Свойство **PortOpen** устанавливает и возвращает состояние порта связи, а также открывает и закрывает COM-порт (Таб. 7). Это свойство недоступно во время проектирования.

Синтаксис

MSComm.PortOpen[= {True | False}]

Таб. 7

Установка	Описание
True	Порт открыт
False	Порт закрыт

Установка свойства **PortOpen** в True открывает порт. Установка в False закрывает порт и очищает принимающий и передающий буфер. Управление связи автоматически закрывает последовательный порт, когда ваша программа завершается. Перед открытием порта, удостоверьтесь, что свойство **CommPort** установлено к существующему номеру порта, а свойство **Settings** имеет установки, поддерживаемые вашими аппаратными средствами.

Свойство **Input**

Свойство **Input** возвращает и удаляет символы из получающего буфера. Это свойство недоступно при проектировании и только доступно для чтения во время выполнения программы.

Синтаксис

MSComm.Input

Примечание. Свойство **InputLen** определяет количество символов, которые читаются свойством **Input**. Установка свойства **Input** в 0 приводит к тому, что свойство **InputLen** читает все содержимое буфера.

Тип данных – String

Пример:

```
'Восстановить все доступные данные
Comm1.InputLen = 0
'Проверка данных
If Comm1.InBufferCount Then
'Данные для чтения
InString$ = Comm1.Input
End If
```

Свойство **InputLen**

Свойство **InputLen** устанавливает и возвращает число считываемых символов свойством **Input** из буфера.

Синтаксис

MSComm.InputLen[= numChars%]

Если свойству **InputLen** недоступны символы в получающем буфере, свойство **Input** возвращает последовательность нулевой длины (" "). Пользователь может факультативно проверять свойство **InBufferCount** для определения наличия символов перед использованием свойства **Input**. Свойство **InputLen** полезно применять при чтении данных из 'машины' чей вывод форматируется в блоки определенной длины.

Пример:

```
'Выбрать 10 символов из блока данных
Comm1.InputLen = 10
'Считать данные
CommData$ = Comm1.Input
Тип данных для установки свойства – Integer
```

Свойство **InBufferCount**

Свойство **InBufferCount** возвращает количество символов ждущих в получающем буфере. Это свойство недоступно при проектировании.

Синтаксис

MSComm.InBufferCount[= count%]

Тип данных – Integer

Примечание. Не путайте свойство **InBufferCount** со свойством **InBufferSize**, которое отражает полный размер буфера.

Свойство **InBufferSize**

Свойство **InBufferSize** устанавливает и возвращает размер получающего буфера в байтах.

Синтаксис

MSComm.InBufferSize[= numBytes%]

По умолчанию размер получающего буфера равен – 1024 байт. Чем больше вы делаете размер получающего буфера, тем меньшее количество памяти становится доступным вашей прикладной программе, однако слишком маленький размер буфера повышает риск переполнения и потери информации, если не используется подтверждение связи.

Тип данных – Integer

Свойство **Output**

Свойство **Output** записывает последовательность символов в передающий буфер. Это свойство недоступно при проектировании.

Синтаксис

MSComm.Output[= outString\$]

Тип данных – String

Пример:

```
'Показывается, как посылается при печати каждый символ на последовательный
порт
Private Sub Form_KeyPress (KeyAscii As Integer)
Comm1.Output = Chr$(KeyAscii)
End Sub
```

Свойство **OutBufferCount**

Свойство **OutBufferCount** возвращает число символов ждущих в передающем буфере. Вы также можете использовать это свойство для того, чтобы очистить буфер (**MSComm.OutBufferCount** = 0).

Синтаксис

MSComm.OutBufferCount[= 0]

Тип данных – Integer

Примечание. Не путайте свойство **OutBufferCount** со свойством OutBufferSize, которое отражает полный размер передающего буфера.

Свойство **OutBufferSize**

Свойство **OutBufferSize** устанавливает и возвращает размер передающего буфера в байтах.

Синтаксис

MSComm.OutBufferSize[= NumBytes%]

Размер передающего буфера по умолчанию установлен – 512 байт. Чем больше вы делаете размер передающего буфера, тем меньшее количество памяти становится доступным вашей прикладной программе, однако слишком маленький размер буфера повышает риск переполнения и потери информации, если не используется подтверждение связи.

Тип данных - Integer

Событие **OnComm**

Событие **OnComm** генерируется всякий раз, когда происходит изменение значения свойства CommEvent, указывая на возникшее событие или ошибку в передачи данных.

Синтаксис

Private Sub MSComm_ **OnComm** ()

Примечание. Событие CommEvent содержит числовой код фактической ошибки или события, которое сгенерировало событие **OnComm**.

Небольшой пример использования элемента управления MS Comm

```
'Назначаем порт с которым будем работать
Comm1.CommPort = 1
'Делаем установки элемента управления MS Comm
Comm1.Settings = "9600,N,8,1"
'Используем весь буфер
Comm1.InputLen = 0
Открываем порт
Comm1.PortOpen = True
'Посылаем команду модему
Comm1.Output = "AT" + Chr$(13)

'Ожидаем возврата данных
Do
Dummy = DoEvents()
Loop Until Comm1.InBufferCount >= 2
'Читаем ответ "OK" в последовательном порту
InString$ = Comm1.Input
'Закрываем порт
Comm1.PortOpen = False
```