

Глава 5. Введение в работу последовательного порта

Краткий обзор

Последовательный порт обычного персонального компьютера часто называется RS-232 и является одним из наиболее распространенных интерфейсов ввода/вывода компьютера. В этой главе мы кратко рассмотрим последовательный порт персонального компьютера и используем порт ввода/вывода программы LabVIEW и соответствующий тип vi [Виртуальный прибор] функции последовательного порта для *выполнения некоторых простых опытов*.

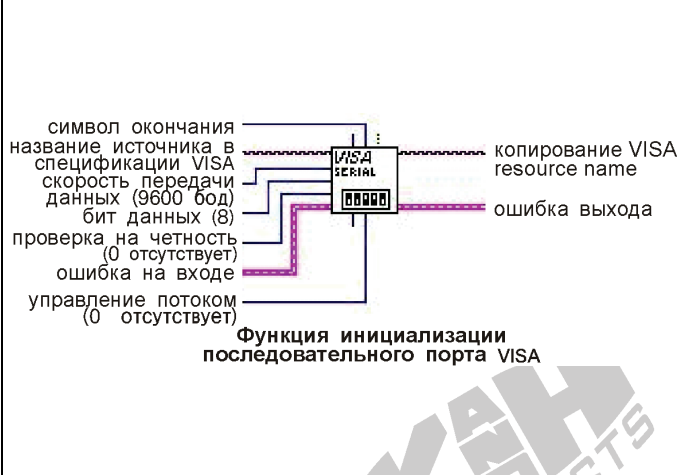
Цели работы

- Понять принцип действия последовательного порта.
- Понять способ использования программы LabVIEW для проверки последовательного порта.

Ключевые слова

- Интерфейс RS-232
- Последовательный порт

■ Описание функции

 <p>Функция инициализации последовательного порта VISA</p>	<table border="1"> <tr> <td data-bbox="898 1503 1034 1574">Путь</td> <td data-bbox="1034 1503 1418 1574">All Functions > Instrument I/O > I/O Serial > VISA Configure Serial Port</td> </tr> <tr> <td data-bbox="898 1574 1034 1709">Описание</td> <td data-bbox="1034 1574 1418 1709">Инициализация последовательного порта, определенного спецификацией VISA (VISA resource name)</td> </tr> <tr> <td data-bbox="898 1709 1034 1906">Вход</td> <td data-bbox="1034 1709 1418 1906">Спецификация VISA определяет открытые ресурсы. Параметры, требующие определения: скорость передачи данных, управление потоком и т.п.</td> </tr> <tr> <td data-bbox="898 1906 1034 1975">Выход</td> <td data-bbox="1034 1906 1418 1975">Если имеется ошибка, то выводится код ошибки</td> </tr> </table>	Путь	All Functions > Instrument I/O > I/O Serial > VISA Configure Serial Port	Описание	Инициализация последовательного порта, определенного спецификацией VISA (VISA resource name)	Вход	Спецификация VISA определяет открытые ресурсы. Параметры, требующие определения: скорость передачи данных, управление потоком и т.п.	Выход	Если имеется ошибка, то выводится код ошибки
Путь	All Functions > Instrument I/O > I/O Serial > VISA Configure Serial Port								
Описание	Инициализация последовательного порта, определенного спецификацией VISA (VISA resource name)								
Вход	Спецификация VISA определяет открытые ресурсы. Параметры, требующие определения: скорость передачи данных, управление потоком и т.п.								
Выход	Если имеется ошибка, то выводится код ошибки								

<p>адрес</p> <p>ошибка входа</p> <p>считанные данные</p> <p>ошибка выхода</p> <p>Функция ввода</p>	Путь	All Functions > Advanced > Port I/O > InPort.vi
	Описание	Считывание знакового целого числа с заданного адреса.
	Вход	Значение address определяет заданный адрес, с которого вы хотите считывать 8-, 16- или 32-разрядное знаковое целое число.
	Выход	Считанные данные являются байтом (8 разрядов) данных, считанным из указанного адреса.
<p>адрес записываемое значение</p> <p>ошибка входа</p> <p>ошибка выхода</p> <p>Функция вывода</p>	Путь	All Functions > Advanced > Port I/O > InPort.vi
	Описание	Запись целого знакового числа по заданному адресу (address).
	Вход	Значение address определяет адрес в памяти ввода/вывода, с которого начинается считывание. Значение write value является байтом (8-разрядное значение), записываемое в системную память.

Постановка задачи

На задней панели персонального компьютера имеются два последовательных порта COM1 и COM2, использующие для телекоммуникационного обмена протокол RS-232. Более ранний вариант последовательных портов, располагающихся на задней панели персонального компьютера, показан на рисунке 1. Здесь имеется один 9-контактный разъем DB-9 (штырьки), называемый COM1, а другой является 25-контактным разъемом DB-25 (штырьки), называемый COM2. Третий разъем является 25-контактным разъемом (гнезда), называемым параллельным принтерным портом. В настоящее время конфигурация внешних портов персонального компьютера немного изменилась и имеет вид, показанный на рисунке 2: два последовательных порта стали разъемами типа DB-9. Сначала определите, какие разъемы имеет ваш компьютер.

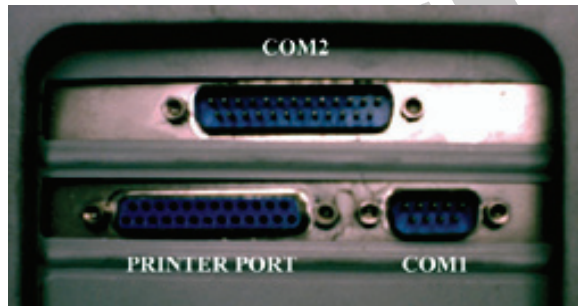


Рисунок 1

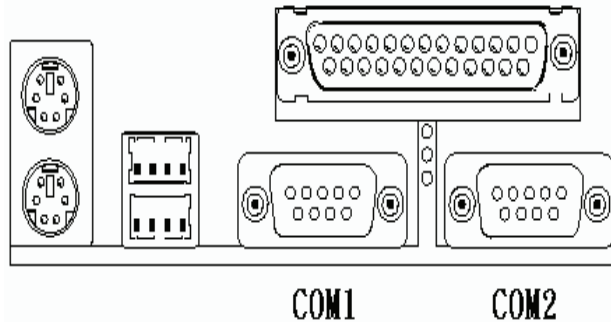


Рисунок 2

Последовательный порт является одним из каналов связи обычного персонального компьютера с внешним миром. Примером является модем, подключенный к Интернету, который использует последовательный порт для передачи и приема сообщений. В системах промышленного управления последовательные порты до сих пор используются для передачи данных или сигналов управления. Причиной этого является их дешевизна. Естественно, последовательный порт должен передавать данные достаточно быстро, в противном случае мы не могли бы использовать его. Предположим, что мы имеем управляющее устройство, которое должно передать команду на удаленно управляемый объект, тогда последовательный порт может иметь несколько режимов, определяемых направлением передачи:

◆ Симплексный режим

Этот режим позволяет передавать сигнал только в одном направлении: управляющее устройство ответственно только за передачу сигнала, а управляемое устройство будет только принимать сигнал. Поток данных может передаваться только в одном направлении, этот режим передачи данных подходит только для односторонней коммуникации и не требует никаких ответов.

◆ Полудуплексный режим

Каждая сторона может передавать поток данных в обоих направлениях, но в одно время могут передаваться данные только в одном направлении. Этот режим передачи данных подходит для двухсторонней коммуникации, но при этом не используется мгновенный ответ. Это обусловлено тем, что когда передатчик отправляет сообщение, приемник не может в это же время передавать. Поэтому если управляемое устройство в экстренной ситуации захочет сразу послать сообщение, то этот режим передачи данных не подходит.

◆ Полнодуплексный режим

Каждая сторона может одновременно передавать данные во взаимно противоположных направлениях. Этот тип передачи данных используется для улучшения полудуплексного режима, хотя он более дорог. Почему? Давайте рассмотрим.

В настоящее время обычный персональный компьютер имеет два последовательных порта, называемых **COM1** или **COM2**. В более старых персональных компьютерах имелась специальная ИС 8250, которая обеспечивала работу последовательного порта, но в новых компьютерах ИС 8250 встроена в южный мост чипсета. Тем не менее функционирование последовательных портов этих двух типов компьютеров аналогично. Эта ИС 8250 называется UART [Универсальный асинхронный приемопередатчик]. Слово «асинхронный» означает, что передатчик и приемник на каждой стороне последовательного порта должны иметь одинаковые скорости передачи, поэтому для обмена данными они не требуют временной синхронизации или специального сигнала. Но для того, чтобы обеспечить точность передаваемых данных, UART должен иметь функцию проверки принятого сигнала на четность, обеспечить квитирование установленной связи и т.п., что позволит расширить использование последовательного порта. Если вы изучили принцип ввода/вывода информации персонального компьютера, то вы знаете, что основой компьютера является центральный процессор (CPU), но CPU общается с периферией через порт ввода/вывода. Последовательный порт компьютера является одним из периферийных

устройств, позволяющих производить обмен данными через порт ввода/вывода. Мы упомянули, что последовательный порт управляется ИС UART, но одного соединения выводов ИС UART с выводами порта ввода/вывода не достаточно. Поскольку в этой ИС имеется несколько управляющих регистров, служащих для программирования некоторых параметров последовательного порта, то мы должны понять работу этого регистра, чтобы уметь управлять последовательным портом, обеспечивающим коммуникации.

Ниже приводится описание внутренних регистров ИС UART, связанных с портом ввода/вывода.

Адрес ввода/вывода		Название регистра		Примечание
COM1	COM2	Английская аббревиатура	Полное название на английском языке	
3F8H	2F8H	THR	Transmitter Holding Register [Регистр-защелка передатчика]	Если разряд DLAB равен 0, то адрес указывает на этот регистр. Этот регистр производит только запись.
3F8H	2F8H	RBR	Receiver Buffer register [Буферный регистр приемника]	Если разряд DLAB равен 0, то адрес указывает на этот регистр. Этот регистр производит только считывание.
3F8H	2F8H	LSB	Baud Rate Divisor Register (LSB fraction) [Регистр делителя скорости передачи (младший разряд)]	Если разряд DLAB равен 1, то адрес указывает на этот регистр.
3F9H	2F9H	MSB	Baud Rate Divisor Register (MSB fraction) [Регистр делителя скорости передачи (старший разряд)]	Если разряд DLAB равен 1, то адрес указывает на этот регистр.
3F9H	2F9H	IER	Interrupt Enable Register [Регистр разрешения прерывания]	Если разряд DLAB равен 0, то адрес указывает на этот регистр.
3FAH	2FAH	IIR	Interrupt Identification Register [Регистр идентификации прерывания]	
3FBH	2FBH	LCR	Line Control Register [Регистр управления линией]	
3FCH	2FCH	MCR	MODEM Control Register [Регистр управления модемом]	
3FDH	2FDH	LSR	Line Status Register [регистр состояния линии]	
3FEH	2FEH	MSR	MODEM Status Register [регистр состояния модема]	

※ DLAB является 7-м разрядом регистра LCR [Регистр управления шиной].

Эта таблица приводит соответствие между внутренним регистром UART и вводом/выводом. Из этой таблицы можно видеть, что имеются 10 разных видов регистров, но имеется только 7 адресов ввода/вывода. Почему так может быть? Поскольку некоторые регистры используют один адрес ввода/вывода, например, THR и RBR, то они используют один адрес, когда данные необходимо записывать по этому адресу и они записываются в THR, а когда данные должны считываться из этого адреса, они считываются из RBR. Другим примером является LSB/ MSB и IER, которые также используют одинаковый адрес. А как они могут знать, какой регистр необходимо использовать, когда данные записываются или считываются? Это определяется состоянием бита DLAB (7-й разряд LCR), например, если для порта COM1 это состояние равно 0, то адрес «3F8_H» относится к THR и RBR, а адрес «3F9_H» относится к IER. Если это состояние равно 1, то адрес «3F8_H» относится к LSB, а адрес «3F9_H» относится к MSB. Таким образом, если вы хотите считывать или записывать данные по этому адресу, то сначала вы должны установить состояние 7-го разряда LCR. Далее мы должны кратко описать функцию каждого регистра, приведенного в таблице.

■ Регистр-защелка передатчика

Регистр-защелка передатчика используется для буферизации выводимых данных. Когда данные будут переданы, 5-й разряд регистра состояния линии устанавливается равным 1. Значение 1 указывает на то, регистр-защелка передатчика пуст и в него можно записать новые данные.

■ Буферный регистр приемника

Буферный регистр приемника содержит принятые данные и если данные не извлечены должным образом, то они могут наложиться на следующие принимаемые данные, что приведет к потере данных.

■ Регистр делителя скорости передачи

Для задания скорости передачи данных ИС 8250 используется внешний кварцевый генератор с частотой 18432 МГц, частоту которого можно поделить и получить еще несколько скоростей передачи. Таким образом, значение этого регистра определяет скорость передачи данных через

последовательный порт. Следующая таблица перечисляет 14 различных режимов скорости, связанных со значением регистра делителя.

Все внутренние регистры UART являются 8-разрядными, поэтому все значения в таблице, вносимые в регистр, должны быть разделены на 2 части – верхние 8 разрядов и нижние 8 разрядов, после этого мы можем поместить верхние 8 разрядов в MSB (3F9_H), а нижние 8 разрядов – в LSB(3F8_H).

Скорость в бодах (бит/с)	Значение регистра делителя скорости передачи (десятичная система)
50	2304
75	1536
150	768
300	384
600	192
1200	96
1800	64
2400	48
3600	32
4800	24
7200	16
9600	12
19200	6
38400	3

■ Регистр разрешения прерывания

Этот регистр используется для установки UART, используемой для генерации сигнала прерывания процессора при наступлении необходимого условия. Это требуется для того, чтобы регистр мог информировать компьютер о необходимости произвести прерывание сервисной подпрограммы (ISR) и отреагировать на вызов. Поскольку практические работы в этой книге не используют функцию прерывания, поэтому мы опустим ее подробное описание. Если читатель захочет побольше узнать об этом, то ему следует ознакомиться с соответствующей литературой.

■ Регистр идентификации прерывания

Этот регистр используется для обеспечения информации для прерывания служебных подпрограмм (ISR) персонального компьютера, чтобы определить, какие устройства UART генерируют прерывание, чтобы перейти к следующим шагам. Поскольку практические работы в этой книге не используют функцию прерывания, поэтому мы опустим ее подробное описание. Если читатель захочет побольше узнать об этом, то ему следует ознакомиться с соответствующей литературой.

■ Регистр управления линией

Этот регистр используется для установки некоторых важных параметров последовательной коммуникации, например, длины слова данных, количества стоповых разрядов, проверки на четность и т.п. Описание установок этого регистра приводится в следующей таблице:

7	6	5	4	3	2	1	0
		Установка проверки четности 000 = Отсутствие проверки четности 001 = Проверка по нечетности 011 = Проверка по четности 010 = Метка 111 = Пробел Другие состояния недопустимы				Установка длины слова данных 00 = 5 бит 01 = 6 бит 10 = 7 бит 11 = 8 бит	
					Количество стоповых битов 0 = 1 стоповый бит 1 = 1,5 стоповых бита (для 5-битового слова) 2 стоповых бита (для 6, 7 или 8 битового слова)		
		0 = Прекращение сигнала отключено 1 = Прекращение сигнала включено					
		0 = Перенос адресации 3F8H/3F9H в регистр THR/RBR 1 = Перенос адресации 3F8H/3F9H в регистр LSB/MSB					

■ Регистр управления модемом

Этот регистр используется для установки соответствующих выводов состояния модема. Описание установок этого регистра приводится в следующей таблице:

7	6	5	4	3	2	1	0
Всегда установлены в 0							Соответствие контакта звонка DTR 1 = Терминал данных готов
							Соответствие контакта звонка RTS 1 = Передача запроса
							Выход 1
							Выход 2
							1 = Режим обратной связи

■ Регистр состояния линии

Регистр состояния линии показывает текущее состояние коммуникации. В этом регистре отражаются ошибки. Имеется также доступ к состоянию буферов приемника и передатчика.

7	6	5	4	3	2	1	0
Всегда установлены в 0							1 = Данные выставлены
							1 = Ошибка переполнения
							1 = Ошибка проверки четности
							1 = Ошибка кадра
							1 = Принят обрыв сигнала
							1 = THR пуст
							1 = THR пуст, линия простаивает

■ Регистр состояния модема

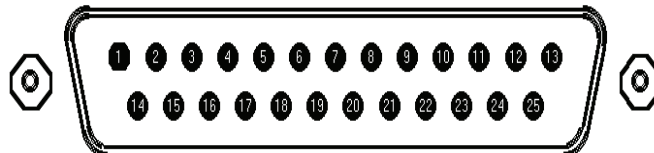
Этот регистр содержит информацию о четырех модемных управляющих линиях, поступающих на устройство. Четыре самых старших разряда

содержат информацию о текущем состоянии входов, а самые младшие разряды используются для указания изменения состояния. Четыре LSB сбрасываются каждый раз, когда регистр считывается. В приведенной ниже таблице показаны функции каждого разряда.

7	6	5	4	3	2	1	0
сигнал обнаружения несущей	индикатор вызова	источник данных готов	разрешение на передачу				1 = Модем готов к приему символа
						1 = Питание на модем подано	
					1 = Звонит телефонный звонок		
				1 = Модем уже соединен с другим модемом			

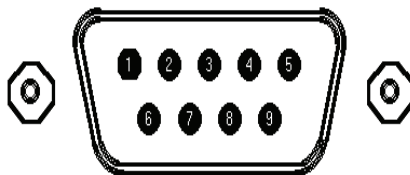
■ Описание разъема RS-232

Назначение контактов разъема RS-232 (DB-25) на персональном компьютере приводится ниже.



Название вывода	Назначение	Название вывода	Назначение	Название вывода	Назначение
1 PG	Защитная земля	10 -	-	19 -	-
2 TxD	Передача данных	11 -	-	20 DTR	Готовность терминала данных
3 RxD	Прием данных	12 -	-	21 -	-
4 RTS	Передача запроса	13 -	-	22 RI	Индикатор звонка
5 CTS	Передача сброса	14 -	-	23 -	-
6 DSR	Данные готовы к передаче	15 -	-	24 -	-
7 SG	Земля сигнала	16 -	-	25 -	-
8 DCD	Обнаружение несущей	17 -	-		
9 -	-	18 -	-		

Назначение контактов разъема RS-232 (DB-9) на персональном компьютере приводится ниже.

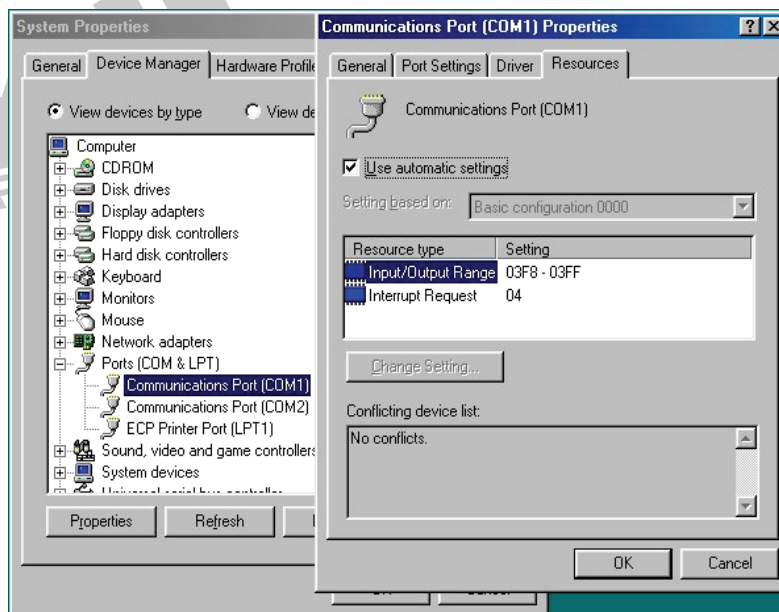


Название вывода	Назначение
1 DCD	Обнаружение несущей
2 RxD	Прием данных
3 TxD	Передача данных
4 DTR	Готовность терминала данных
5 SG	Земля сигнала
6 DSR	Данные готовы к передаче
7 RTS	Передача запроса
8 CTS	Передача сброса
9 RI	Индикатор звонка

Практическая работа 5-1. Изучение последовательного порта

Знаете ли вы, как проверить работу последовательного порта?

Сначала необходимо убедиться в том, что на вашем компьютере есть хотя бы один последовательный порт. Вы можете выполнить следующие действия, чтобы найти «device manager» [Менеджер устройств]. Начните с пункта «start» [Пуск], затем «setting» [Установки], выберите «control» [Управление], выберите «system» [Система] и найдите «device manager». Раскройте «device manager», чтобы найти пункт «COM&LPT», а если вы не можете найти на вашем компьютере пункт «COM&LPT», то это означает, что компьютер устарел и придется приобрести новый. Когда вы раскроете пункт «COM&LPT», то вы можете найти один принтерный порт (LPT1) и два коммуникационных порта (COM1, COM2), но на ноутбуке может быть только один коммуникационный порт (COM1), хотя это не имеет никакого значения, поскольку во всех практических работах этой книги мы используем только один коммуникационный порт. На следующем рисунке показаны свойства коммуникационного порта (COM1).



After Our PC owned at least one serial port, we can use the LabVIEW to do a simple self-examination test to inspect the status of the serial port.

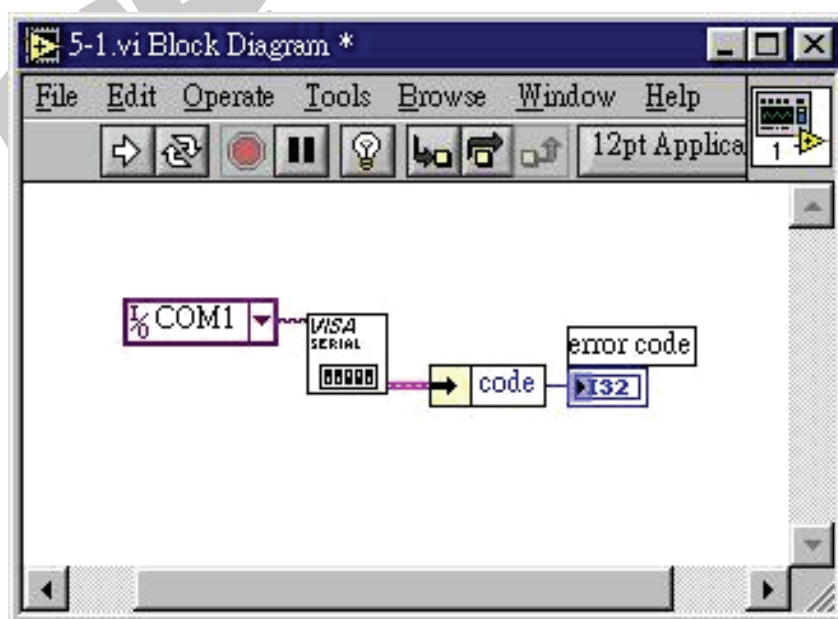
Если ваш компьютер имеет по крайней мере один последовательный порт, то мы сможем использовать программу LabVIEW и выполнить простое самотестирование, чтобы проверить состояние последовательного порта.

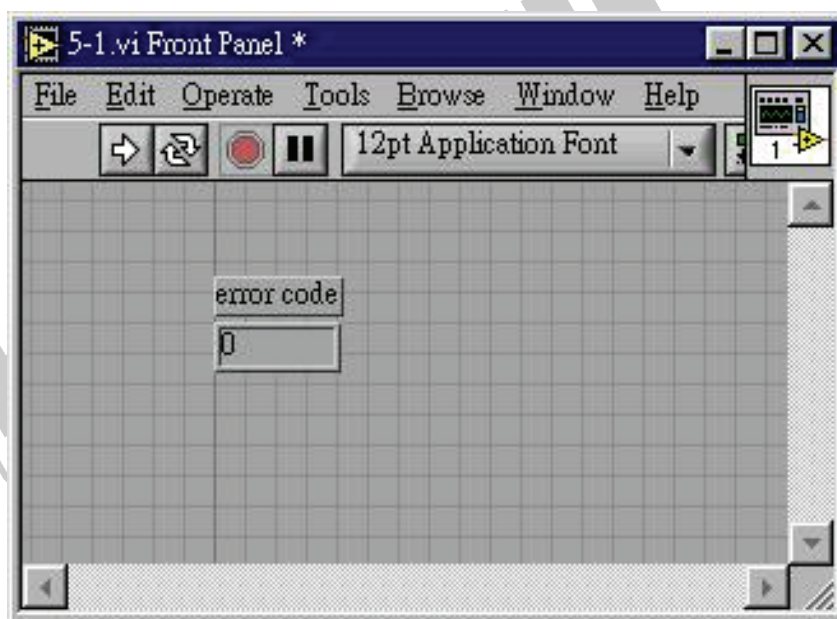
1. Выберите в окне **Block Diagram** функцию **All Functions > Instrument I/O > Serial > VISA Configure Serial Port.VI**

Создайте новый ввод/вывод для одного входного параметра «**COM port number**» [Номер COM-порта].

Создайте индикатор выхода «**error code**» [Код ошибки].

Next to decide what number should be filled into this parameter of **port number**. We should go back to find how many communication ports we have. Right now we use COM1 or COM2 as activated port. Assume that it is COM1 then according to the above table we can find out the corresponding number.





2. Необходимо указать число, показывающее работоспособность COM1. Если мы нажмем кнопку выполнения, что при этом будет показано на экране? Если мы получим 0, то это будет означать, что все нормально, качество порта отличное (порт работает). Если же мы получим любое другое число, это будет означать, что коммуникационный порт (COM1) не работает. Возможно, это ошибка установки. Возвратитесь к предыдущим пунктам и еще раз проверьте систему и установки.

Конец практической работы 5-1.

Практическая работа 5-2.

Самодиагностика схемы UART [Универсальный асинхронный приёмопередатчик]

Еще одним формальным тестированием является внутренняя самодиагностика схемы UART. В качестве примера рассмотрен следующий процесс для порта COM1.

Мы можем проверить работоспособность UART, используя 4-й разряд функции циклического тестирования.

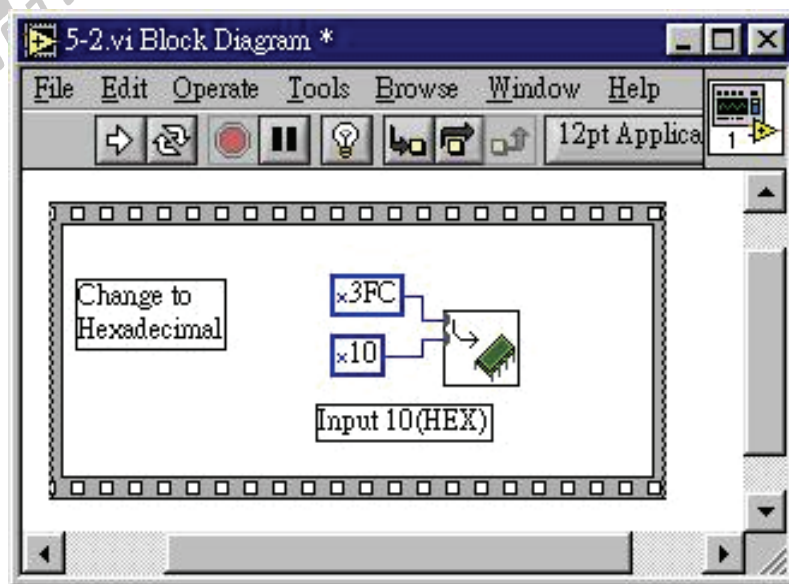
Функция цикла 4-го разряда MCR (установлен равным 1) может выполнять следующее действие:

- ◆ Установка последовательного выхода UART в состояние метки (1).
- ◆ Отключение последовательного входа UART от внешнего мира.
- ◆ Автоматическое внутреннее соединение последовательного выхода UART внутри с входом.
- ◆ Поочередное соединение четырех разрядов MSR (DSR, CTS, RI и DCD) с DTR, RTS, OUT1 и OUT2 MCR.

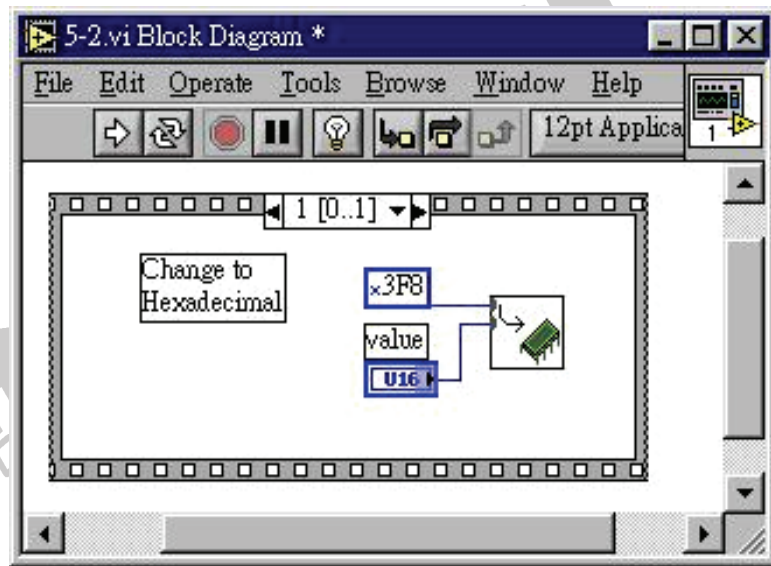
[MCR]	[MSR]
DTR	DSR
RTS	CTS
OUT 1	RI
OUT 2	DCD

Имея такую удобную функцию, вы можете быстро протестировать последовательный порт. С помощью встроенной циклической функции диагностики UART, выход и вход последовательного порта можно соединить внутри. Тогда, если мы выводим программным образом байт данных на последовательный порт, он будет получен внутренним приемником. Таким образом, мы можем использовать эту функцию для тестирования работоспособности последовательного порта.

1. Сначала в окне **Blocked Diagram** создайте блок **Stacked Sequence Structured** и отрегулируйте его размер. Затем в **Sequence** создайте новую функцию **OutPort.vi**. Чтобы из каждого входного параметра «**register address**» [Адрес регистра] и «**value**» [Значение] создать новую переменную функции **OutPort.vi**, мы должны преобразовать формат этих двух переменных в шестнадцатеричный. В параметр адреса регистра вводим число **3FC_H**, которое является адресом MCR порта COM1, а в параметр «**value**» временно не вводится никакое число.



2. Создайте из Sequence новый страничный кадр и соедините его с обратной стороны. В этот новый страничный кадр мы введем функцию **port.vi**. Чтобы получить переменную величину из параметра «**register address**», мы должны преобразовать формат этой переменной в шестнадцатеричный. Чтобы создать управляющий элемент из параметра «**value**», его входной формат будет преобразовываться в шестнадцатеричную систему автоматически. В параметр переменной «**register address**» введем число **3F8_H**, что означает передачу входной величины управляющего значения в регистр-защелку передатчика COM1. Обратите, пожалуйста, внимание на тот факт, что если мы не преобразуем переменную входного параметра «**register address**» в шестнадцатеричный формат, то входное значение **3F8_H** визуально отображено не будет.



3. Создайте из **Sequence** новый страничный фрейм и снова соедините его с обратной стороны. В этот новый страничный фрейм мы поместим блок, который предназначен для создания задержки, необходимой для ожидания выхода данных из регистра-защелки передатчика. Поскольку мы не установили скорость передачи COM1, то мы должны использовать исходные установки, задаваемые по умолчанию. Обычно порт COM1 имеет следующие параметры: скорость передачи данных 9600 бит/с; проверка на четность отсутствует; ширина данных 8 бит; стоповый бит один. Поэтому суммарное время задержки передачи вычисляется следующим образом:

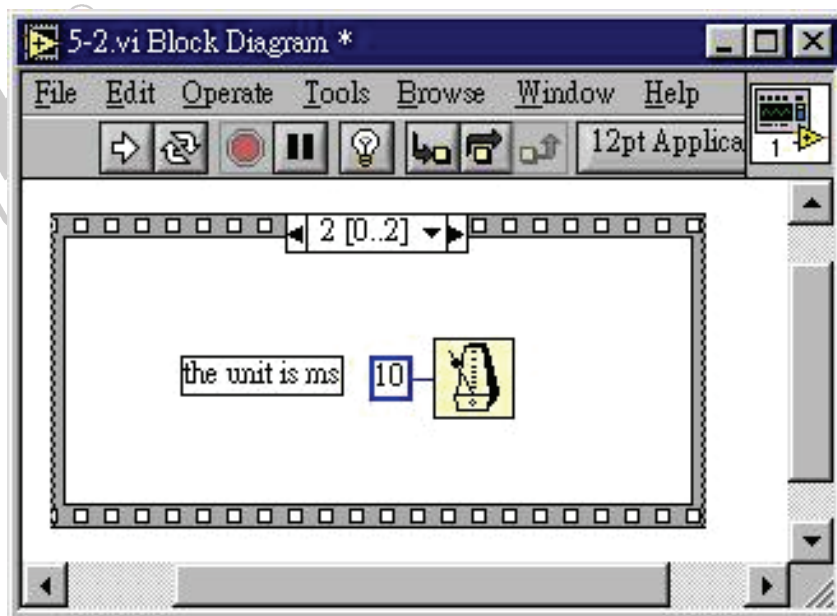
Полное количество передаваемых битов \times время передачи одного бита =
полное время передачи

Полное время передачи = 1 стартовый бит + 8 бит данных + 1 стоповый
бит = 10 бит

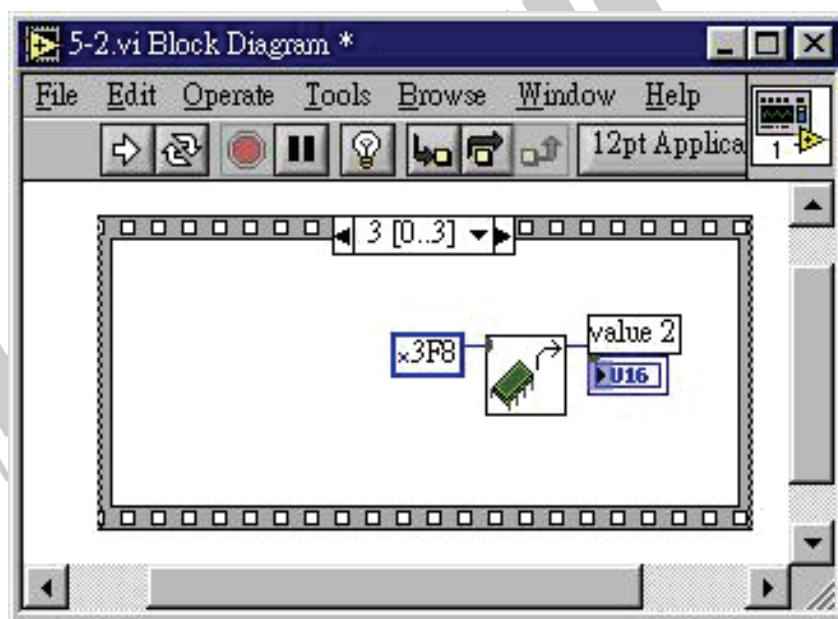
Время передачи одного бита = время, эквивалентное скорости 9600 Гц =
104,2 мкс

Полное время передачи = 104,2 мкс \times 10 разрядов = 1,042 мс

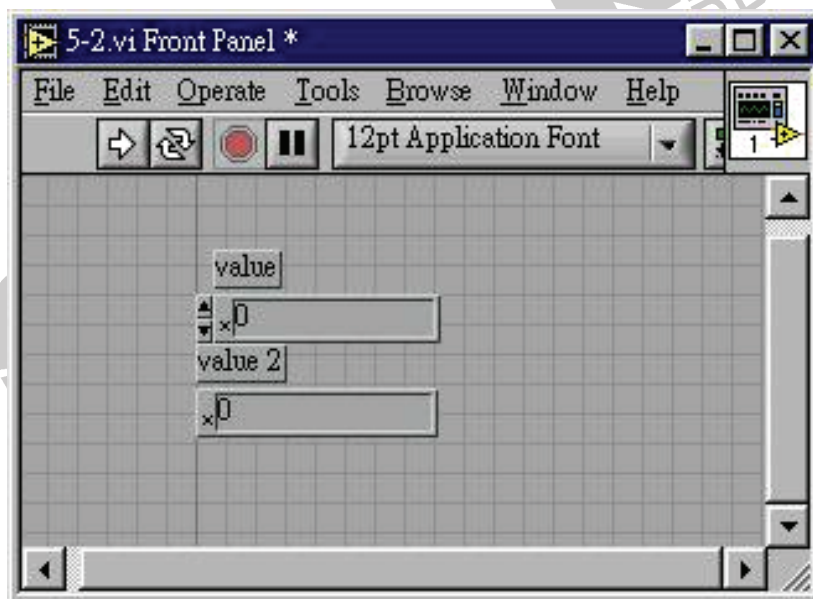
Итак, время задержки этой страницы должно быть не менее 1,042 мс, а для гарантии мы установим время задержки 10 мс. Обычно параметр времени задержки для .vi задается с помощью временного блока в мс, следовательно, мы можем присвоить переменной значение 10, что будет означать задержку 10 мс.



4. Создайте из **Sequence** новый страничный фрейм и снова соедините его с обратной стороны. Поскольку в качестве способа тестирования используется самодиагностика, то это означает, что данные передаются внутри UART. Следовательно, данные должны считываться из регистра «Receiver Buffer register». Далее, чтобы считать значение по адресу **3F8_H**, мы используем функцию «**InPort.vi**», при этом не забудьте преобразовать формат адреса регистра в шестнадцатеричную форму. Мы также должны обеспечить индикацию величины с выхода «**InPort.vi**».



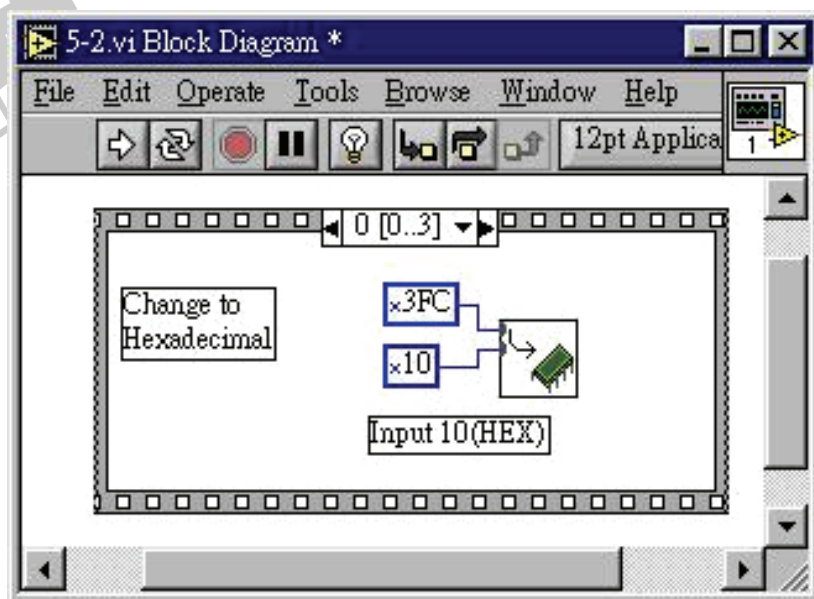
5. В окне **Front Panel** будет показан новый управляющий элемент и только что созданный элемент индикации.



6. Попробуйте ввести в показанный выше элемент управления значение **256** и щелкнуть по кнопке однократного исполнения. Теперь проверьте, соответствует ли этому значения элемент индикации. Ответ должен быть отрицательным. Вы помните, что значение первого страничного фрейма

равно нулю? Именно это значение мы хотим установить в MCR. Теперь в первом страничном фрейме мы можем задать значение **10 (шестнадцатеричное)** и, щелкнув на кнопке однократного исполнения еще раз, посмотреть, что отображает элемент индикации.

 Run [Исполнение]



Если после щелчка по кнопке однократного исполнения вы увидите, что значение элемента индикации соответствует значению управляющего элемента, то это означает успешное прохождение эксперимента и нормальную работу последовательного порта.

Конец практической работы 5-2

Упражнения и обсуждение

1. Знаете ли вы, почему регистр MCR должен содержать число 10 (шестнадцатеричный формат)? Что это означает?
2. Ранее мы упоминали, что четыре считываемых разряда MSR были соединены с DTR, RTS, OUT 1 и OUT 2 MCR в соответствии со следующей таблицей:

[MCR]	[MSR]
DTR	DSR
RTS	CTS
OUT 1	RI
OUT 2	DCD

Знаете ли вы, как использовать виртуальный прибор программы LabVIEW, чтобы управлять этими сигналами и считывать их?