

Опыт разработки беспроводной коммуникационной системы для сбора данных в промышленных системах

Алексей КУЗЬМИНОВ, к. т. н.
compmicrosys@mail.ru

Бурный рост беспроводных технологий, наблюдающийся в последнее время, побуждает многих разработчиков создавать устройства для беспроводной связи уже готовых, серийно выпускаемых приборов, изначально использующих интерфейс RS232 для сопряжения с компьютером. В качестве аппаратных средств, обеспечивающих подобную беспроводную связь, используются радиомодемы, имеющие интерфейс RS232. Однако различия в физических принципах обмена информации в радиомодемах и в проводных интерфейсах, каковым является, например, RS232, создают препятствия, преодолеть которые разработчикам, не сведущим в радиосвязи (и не желающим вдаваться в подробности), представляется отнюдь не тривиальной задачей. В статье рассказывается, по каким параметрам следует выбирать радиомодем, как его протестировать, как обеспечить «прямую видимость» антенн, используя некоторые дополнительные аппаратные средства, а также идет речь о некоторых нюансах в различиях протоколов обмена в модемах и самом интерфейсе RS232, которые необходимо учитывать для решения задач подобного рода.

Постановка задачи

Предположим, имеется серийно выпускаемое некоей фирмой устройство, выполняющее определенную функцию и сопряженное для обмена информацией с компьютером по интерфейсу RS232. К подобным устройствам, например, можно отнести небольшие устройства сбора, обработки, архивирования измеренной информации и передачи ее в компьютер по запросу последнего. В качестве примеров подобных устройств можно привести различные теплосчетчики, газовые корректоры и им подобные устройства, которые уже давно серийно выпускаются промышленностью и годами работают на предприятиях. Для передачи информации с устройства

в компьютер уже разработаны и отлажены определенные протоколы обмена данными, запрограммированные в микроконтроллер устройства и используемые в программе для компьютера. Требуется «разорвать» провод связи по интерфейсу RS232, сопрячь устройство и компьютер с радиомодемами, имеющими интерфейс RS232 и удалить устройство от компьютера, например, на расстояние в 1 км. При этом программное обеспечение микроконтроллера прибора и компьютера изменять не допускается.

Предлагаемое решение задачи

Протоколы обмена информации устройства и модема

Для обеспечения достоверности и надежности информации, передаваемой из устройства в компьютер, для каждого устройства разработаны помехозащищенные протоколы обмена по интерфейсу RS232. Как правило, протокол обмена устроен таким образом, что по определенным командам, поступающим с компьютера в устройство, последнее передает информацию некоторыми порциями или пакетами. Длина такого пакета обычно составляет чуть более 256 байт. Пакет состоит из информационных данных, объем которых не превышает 256 байт, и служебной информации (несколько байт), в которую, в частности, включена длина пакета, контрольная сумма пакета и некоторая дополнительная информация. Принято, что для передачи количества информационных байт в служебной информации отводится не более одного байта (вот почему объем информационных байт не превышает 256). Таким образом, длина пакета может варьироваться от нескольких байт до 256 плюс несколько байт (например, 260–265). Кроме того, связь по протоколу, как правило, является дуплексной, то есть при приеме пакета от устройства компьютер может выдать в интерфейс какой-либо символ (например, символ, обозначающий ошибочно принятые данные и т.п.). Этот символ, подчеркнем, компьютер может выдать по своей линии TxD, в то время как по линии RxD он продолжает принимать информацию.

Если посмотреть описания большинства радиомодемов, то в глаза сразу бросается несколько проблем. Первая связана с длиной передаваемого и принимаемого пакетов. Длина этих пакетов в модемах не превышает 256 байт. Это вполне можно объяснить: разработчики модемов используют такую длину пакета, поскольку она является, например, максимальной длиной пакета, которую способен передать радиотрансивер, входящий в состав радиомодема. Если, например, прибор передает в компьютер пакет размером в 260 байт, то радиомодем разбивает этот пакет на два, передает вначале 256 байт, а затем еще 4. Причем временная задержка между передачей первого и второго пакета может достигать таких значений, что компьютерная программа, принимающая пакет в 260 байт, может прервать свою работу по тайм-ауту, и, таким образом, обмен сорвется. Поэтому при выборе радиомодема необходимо внимательно отнестись к этой проблеме. К счастью, программное обеспечение некоторых достаточно «навороченных» радиомодемов справляется с этой проблемой. В них предусмотрен специальный буфер (размером намного больше, чем 256 байт, например, 512 или более), куда поступает принятая из эфира информация. Эта информация выдается в интерфейс только тогда, когда весь буфер (например в 260 байт) полностью принят. Причем эти 260 байт выводятся в интерфейс непрерывно и, таким образом, компьютерная программа «не чувствует» различия между приемом пакета с радиомодема и чистого интерфейса RS232 и не прерывает работу по тайм-ауту. А вот тайм-аут между приемом пакетов, как правило, в такой программе выбран достаточно большим, что объясняется относительно медленной реакцией прибора на команды компьютера. Вторая проблема связана с тем, что подавляющее число радиомодемов являются полудуплексными. Этот момент также легко объясним: все радиотрансиверы, насколько это известно автору, не могут одновременно и передавать, и принимать сигнал при помощи одной и той же антенны. В связи с этим, в описании таких радиомодемов указано, что перед передачей информации в модем необходимо проверить какую-либо линию квитирования, например, линию CTS интерфейса, и только при ее разрешающем состоянии передавать информацию в модем из интерфейса. Но дело в том, что большинство компьютерных программ, предназначенных для обмена информацией с устройством, такую проверку не делают, поскольку они «предполагают» дуплексную связь и в такой проверке нет необходимости. Поэтому если в описании модема пользователя «заставляют» проверять линию квитирования, то такой модем использовать в приложениях подобного рода нельзя. Программное обеспечение хорошего радиомодема устроено таким образом, что в модеме помимо приемного буфера организован и передающий буфер, в который независимо от того, в каком состоянии находится модем — принимающим информацию из эфира либо свободным — записывается принятая из интерфейса информация. Другими словами, хороший модем для пользователя представляется именно дуплексным устройством.

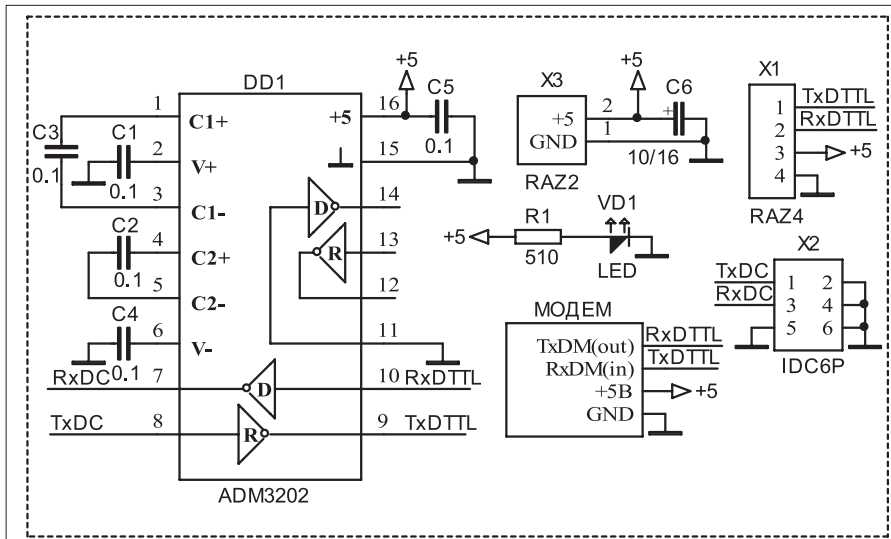


Рис. 1а. Плата сопряжения модема с компьютером (№ 1)

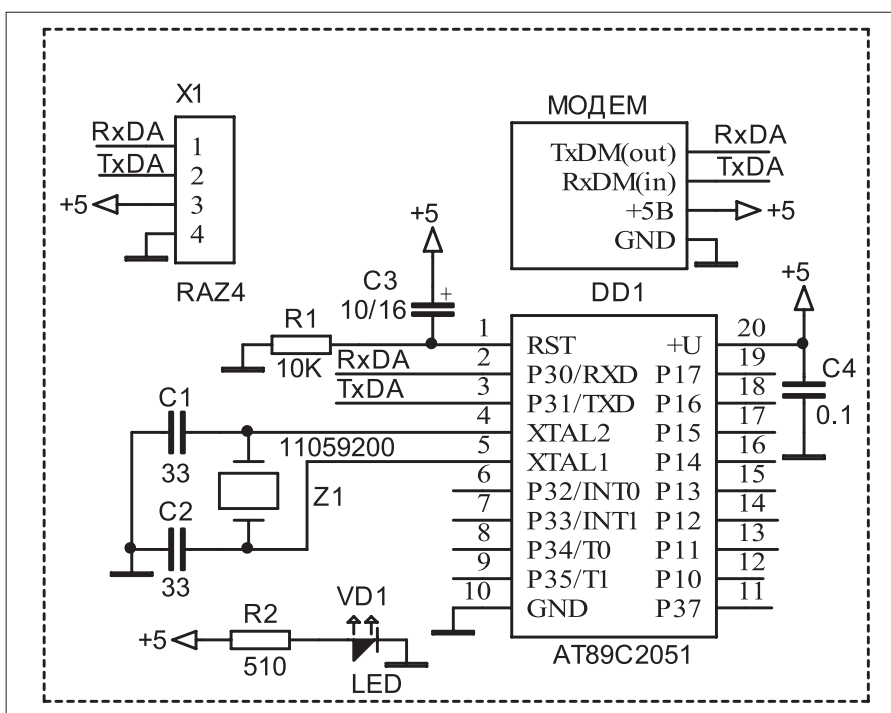


Рис. 1б. Плата сопряжения модема с AT89C2051 (№ 2)

Третья проблема связана с размером пакетов, передаваемых модемом непосредственно в эфир. Хороший модем должен предоставлять пользователю выбор объема в байтах такого пакета (не путать этот объем с объемом пакета, используемым прибором для передачи информации, о котором говорилось ранее). Дело в том, что чем меньше длина пакета передаваемого в эфир, тем больше вероятность его безошибочного приема. Если такой пакет принят с ошибками, то он передается и принимается еще раз (так, как правило, организована работа многих радиотрансиверов и программного обеспечения работы модемов). С другой стороны, чем больше длина пакета, тем меньше вероятность его безошибочной передачи, и в случае приема пакета с ошибками он должен быть повторно передан. Но, и в этом вся суть возможности выбора длины пакета пользователем, чем длиннее пакет, тем больше времени требуется на его повторную передачу и прием. Если это время

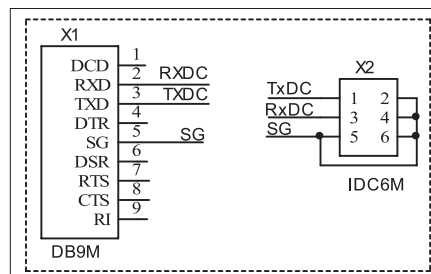


Рис. 1в. Кабель связи с компьютером

превысит тайм-аут программы для компьютера, то программа остановит свою работу и сообщит об ошибке, в результате чего обмен сорвется. Выбирая экспериментально длину пакета, можно значительно улучшить надежность обмена. Если такого выбора в модеме не предоставляется пользователю, то к такому модему следует относиться с осторожностью. В большинстве случаев его лучше не использовать в приложениях подобного рода.

И, наконец, четвертая проблема, которую хотелось бы отметить, — это предоставляемая пользователю возможность выбрать скорость передачи/приема в эфире и интерфейсе. А эти скорости могут отличаться. Пусть, например, компьютерная программа и устройство обмениваются информацией по интерфейсу RS232 на скорости 9600 бод. Не факт, что скорость обмена информацией приемного и передающего модема в эфире должна соответствовать также 9600 бод. Дело в том, что чем ниже скорость обмена в эфире, тем больше расстояние, на которое могут быть разнесены модемы при одинаковой надежности обмена информацией и наоборот. Но, с другой стороны, если скорость обмена информацией в эфире меньше скорости на интерфейсе, увеличивается время задержки между приемами пакетов. Если это время превысит тайм-аут программы для компьютера, то обмен сорвется. Экспериментально подбирая скорость обмена в эфире, можно добиться компромисса между расстоянием и надежностью обмена. Поэтому этот фактор также следует принимать в расчет. Как правило, хороший модем позволяет выбирать и скорость обмена по интерфейсу, и скорость обмена в эфире.

Помимо этих четырех моментов, которые, с точки зрения автора, следует учитывать, существует и масса других настроек модемов, предоставляемых пользователю. Причем чем больше таких возможностей, тем вероятнее, что такой модем следует использовать в приложении ввиду его высокой гибкости.

Устройство тестирования модема

Но вот, наконец, вы приняли решение и приобрели два радиомодема. С чего следует начать, чтобы удостовериться, что они подойдут для работы приложений? Прежде всего, конечно, необходимо тщательно изучить документацию, попробовать запрограммировать модем при помощи компьютера, используя различные его настройки. Если все это удалось проделать с модемом, на взгляд автора, следует его протестировать.

Для проверки функционирования модемов автор рекомендует изготовить (пусть даже на макетных платах) нехитрое устройство, схема которого приведена на рис. 1.

Как видно из рис. 1а, б, устройство состоит из двух плат: № 1 и № 2. Первая плата предназначена для сопряжения модема с компьютером по интерфейсу RS232. В схеме используется достаточно надежный преобразователь интерфейса ADM3202. Питание (+5 В) на плату подается через двухконтактный разъем X3. Для индикации включения питания используется светодиод VD1. Сигналы интерфейса TxDC и RxDC подаются на плату через разъем X2, к которому подключается кабель связи с компьютером (COM1). Модем, показанный на плате условно, устанавливается в панельку; к нему подходят сигналы RxDTTL и TxDTTL уже TTL уровня. На схеме можно заметить еще один разъем X1, служащий для прямого соединения двух плат между собой для проверки совместной работы программ компьютера и микроконтроллера. В таком режиме работы плат модемы, естественно, вытаскиваются из панелек. На плате № 2 в панельке установлен второй модем, также показанный условно; к нему подходят сигналы TxDA и RxDA с микроконтрол-

лера AT89c2051 (DD1). Разъем X1, как и в плате № 1, служит для прямого (безмодемного) соединения двух плат между собой. Питание (и «земля») на плату № 2 в таком режиме работы плат поступает с 3 и 4 контактов разъема X1. При работе плат с модемами к контактам 3 и 4 подключается отдельный блок питания.

Кабель связи с компьютером, показанный на рис. 1в прямоугольнике, очерченным пунктиром, представляет собой обычный плоский ленточный кабель, в котором чередуются сигнальные и «земляные» шины. Цепочка R1-С3, подключенная к выводу RST микроконтроллера, обеспечивает его сброс/запуск при включении питания.

Программы для микроконтроллера и компьютера выполняют следующие действия. Программа для микроконтроллера выводит 150 пакетов (строк) по 260 байт каждый. После вывода очередной строки микроконтроллер ожидает приема от компьютера символа ('@' — 40h), означающего, что пакет принят компьютером, и после поступления такого символа выводит следующий пакет. Компьютер, приняв строку из 260 байт (символы русского и латинского алфавита), выводит ее на экран монитора, и посылает в интерфейс символ '@'. В программе для компьютера предусмотрен тайм-аут, при превышении которого выдается сообщение о сбое, после которого программа заканчивает работу.

Скорость обмена в программах установлена в 9600 бод. Текст программы для компьютера (tstmodem. cl), написанной на языке Clarion v. 3.100 и работающей в ОС MS-DOS, приведен ниже. Далее приведен файл-проект для трансляции программы: tstmodem. pr. Вслед за ним приведен текст программы для микроконтроллера на макро-ассемблере. Программы достаточно просты и в комментариях (кроме тех, что приведены в текстах) не нуждаются.

Необходимо отметить, что тестовые платы, схемы которых показаны на рис. 1, и приведенные ниже программы, в совокупности являются достаточно жестким тестом, и не каждый модем способен его выдержать без сбоев. С другой стороны, если модем выдержит такой тест, то его можно смело использовать в приложениях.

```

Программа TSTMODEM.CLA:
-----
! Программа тестирования модема.
-----

PROGRAM

!-----
! Определение переменных
!-----
S1      string(65) !1-я строка - 65 байт
M1      byte.dim(65).over(S1) !Массив M1, совмещенный со строкой S1

S2      string(65) !2-я
M2      byte.dim(65).over(S2)

S3      string(65) !3-я
M3      byte.dim(65).over(S3)

S4      string(65) !4-я
M4      byte.dim(65).over(S4)

M       byte.dim(100)      !Массив принимаемых байт INBYTE.

```

```

S       string(100).over(M)      !Строка принимаемых байт.совмещенная с массивом M.

N       byte
k       byte
B       byte
BOU     byte      !Переменная для выводимого байта
i       byte
j       byte
l       byte

Y00h   byte(00h)
Y01h   byte(01h)
Y02h   byte(02h)
Y03h   byte(03h)
Y04h   byte(04h)
Y06h   byte(06h)
Y07h   byte(07h)
Y12    byte(12)
Y47h   byte(47h)
Y80h   byte(80h)
Y0fh   byte(0fh)

ERR     byte(0)      !Переменная для ОШИБКИ
!-----
CODE
!-----
blank   !Очистка экрана

N=150
!-----

INIT    do INITRS !Инициализация RS232.

        in(3f8h,B)      !Холостой ввод - для сброса бита 0(DR) в 3fdh (в "0").

        setcursor(1,1) !Установка курсора в левый верхний угол экрана.

        BOU=N           !Количество циклов вывода по 260 байт.
        do OUTBYTE     !Вывод N в AT89C2051.
!-----

        loop k=1 to N  !Начало цикла вывода
!-----
        loop j=1 to 65 !Обнуление
M[j]=0      !принимаемых
M1[j]=0     !строка
M2[j]=0
M3[j]=0
M4[j]=0
.
!-----
        loop i=1 to 65 !Прием 1-й строки - 65 байт.
do INBYTE
!type(chr(M[i]))
if ERR=1 then goto WYKL.
.
S1=S
!-----
        loop i=1 to 65 !Прием 2-й строки
do INBYTE

if ERR=1 then goto WYKL.
.
S2=S
!-----
        loop i=1 to 65 !Прием 3-й строки
do INBYTE
if ERR=1 then goto WYKL.
.
S3=S
!-----
        loop i=1 to 65 !Прием 4-й строки
do INBYTE
if ERR=1 then goto WYKL.
.
S4=S
!-----
        type(S1)      !Вывод 1-й строки - 65 байт.
type('<0dh.0ah>')

```

```

type(S2)      ! 2-й
type('<0dh,0ah>')

type(S3)      ! 3-й
type('<0dh,0ah>')

type(S4)      ! 4-й
type('<0dh,0ah>')

type('k=')    ! Вывод k
type(k)
type('<0dh,0ah>')

! do DELIS

      BOUT=40h      ! Вывод '@' в AT89C2051
      do OUTBYTE
!-----
      . ! Конец цикла вывода (по N)

W      type('<0ah,0ah>') !Прокрутка экрана вверх на
      2 строки
      type('<0dh>')

WYKL type('Для выхода нажмите ESC')
      loop until keyboard()
      ask
      if keycode()=256 then break. !esc
!-----

E      return ! Окончание программы и выход
      в DOS

##### Подпрограммы
#####
! П/п инициализации RS232.
! Инициализация COM-порта.
!-----
INTRS routine
      out(3fh,Y80h) ! DLAB=1 для установки делителя.
      out(3fh,Y12) ! Установка мл.б.скор.:Y01h-
      115200,Y02h-576000,Y12-9600бод.
      out(3fh,Y00h) ! Установка ст.б. скор.=0.
      out(3fh,Y07h) ! DLAB=0.Режим: 8 бит данных,2
      стоп,нет пар.,сброс TxD.
      out(3fh,Y00h) ! Запрет всех прерываний по порту
      3f8h.
!-----

! П/п ввода байта (байт в M[i])
!-----
! INBYTE routine
! loop.in(3fdh,B):if band(B,1):break..
! in(3f8h,M[i]) !Ввод байта в i-й элемент
      массива M.

INBYTE routine
      T#=CLOCK() !Запоминание времени в начале
      ввода байта
      INS1 in(3fdh,B)
      if band(B,1) then in(3f8h,M[i]): goto RETO. !Ввод
      байта в i-й элемент
      массива M.
!-----
! Организация TimeOut'a в 10 сек.
!-----
      if CLOCK()-T#<130 ! TimeOut=10 сек.
      goto INS1
      else
      ERR=1
      type('Выход по тайм-ауту')
      type('<0dh,0ah>')

      type('S1=')
      type(S1) ! Вывод 1-й строки - 65 байт.
      type('<0dh,0ah>')
      loop i=1 to 65 ! Вывод кодов 65 символов 1-й
      строки
      type(M1[i]) ! в десятичном виде
      type('_')
      type('<0dh,0ah>')

      type('S2=')
      type(S2) ! 2-й
      type('<0dh,0ah>')

```

```

loop i=1 to 65
type(M2[i])
type('_')

type('<0dh,0ah>')

type('S3=')
type(S3) ! 3-й
type('<0dh,0ah>')
loop i=1 to 65
type(M3[i])
type('_')

type('<0dh,0ah>')

type('S4=')
type(S4) ! 4-й
type('<0dh,0ah>')
loop i=1 to 65
type(M4[i])
type('_')

type('<0dh,0ah>')

type('k=') ! Вывод k
type(k)
type('<0dh,0ah>')
type('<0dh,0ah>')

!-----
RETO

! П/п вывода байта (байт в BOUT)
!-----
OUTBYTE routine
loop.in(3fdh,B):if band(B,20h):break.. !Ожид. готовнос-
ти передатчика (transmitter empty).
out(3f8h,BOUT) !Вывод байта из пере-
менной BOUT.
loop.in(3fdh,B):if band(B,40h):break.. !Ож-е выхода
байта из PC(OK to send).
!-----
##### конец подпрограмм
#####
DELIS routine
      T#=CLOCK():loop while CLOCK()-T#<500. !задержка
      1 сек.

Файл-проект программы TSTMODEM.CLA:
#system auto exe
#model extended

#pragma data(stack_size=>24576,threshold=>4096)
#pragma check(stack=>on,index=>on)
#pragma link_option(map=>off,case=>on)
#compile tstmodem.cla /check(stack=>off,nil_
ptr=>off,index=>off) /debug(vid=>off,line_
num=>off)
#pragma link(%clapfx%dos.lib)
#set exports='tstmodem'
#link tstmodem

Программа RSTSTMOD.ASM для микроконтроллера
!-----
: Программа вывода 4-х строк по 65 байт
: (итого 260 байт) с помощью AT89C2051
: для проверки модема.
!-----

:-----
: Макросы
!-----
:Макрос ввода байта
!-----
INBYTE .macro
      jnb ri,$
      mov a,sbuf
      clr ri
      .endm
!-----
: Макрос вывода байта.
: Байт в a.
!-----
OUTBYTE .macro
      mov sbuf,a
      jnb ti,$
      clr ti
      .endm
!-----

```

```

.DATA
:-----
: Переменные - нет
:-----

.CODE
:-----
      org $
      mov sp,#50h
!-----

:-----
: Инициал. посл. порта
!-----
SPINIT: mov scon,#50h
      mov pcon,#80h
!-----
TINIT: mov tmod,#20h
      mov tcon,#40h
      mov tl1,#250
      mov th1,#250 ;9600 бод.
      setb tr1
!-----
: Принимаем N - количество раз
: вывода по 260 байт.
!-----
START: INBYTE
      mov r0,a ; N->r0
!-----
OUTN: mov b,#4 ;Выводим 4 строки по 65
      символов.
OUT4: mov a,#40h ;1-й символ - '@'
!-----
OUT65: OUTBYTE ;Вывод 65 символов с кодами
      inc a ;от 40h до 81h.
      cjne a,#81h,OUT65 ;Вывод 65-го символа 'A'-рус-
      ская.
      djnz b,OUT4
!-----
: Прием "@" от компьютера
!-----
INBYTE
      cjne a,#40h,ER
!-----
      djnz r0,OUTN
!-----
      jmp START
!-----

:-----
: Выводим ERROR
!-----
ER: mov a,'E'
      OUTBYTE
      mov a,'R'
      OUTBYTE
      mov a,'R'
      OUTBYTE
      mov a,'O'
      OUTBYTE
      mov a,'R'
      OUTBYTE
      jmp E
!-----

E: jmp $
      .end

```

Обеспечение прямой видимости антенн

Как правило, все производители радиомодемов настоятельно рекомендуют использовать направленных антенн, расположенных таким образом, чтобы обеспечивалась прямая видимость. Это неудивительно, поскольку эта рекомендация отражает физическую сущность радиопередачи в диапазоне ВЧ и СВЧ. Однако обеспечить прямую видимость антенн не всегда удается, используя только кабель интерфейса RS232 или фидер, высокочастотный кабель, соединяющий модем с антенной. Дело в том,

что расположение объектов, в том числе зданий на территории, где используются модемы, в подавляющем большинстве случаев не позволяет обеспечить прямую видимость антенн. Если прибор (например, газовый корректор) расположен где-нибудь в подвале (а именно там проходят все газовые магистрали), а компьютер находится в здании (скажем, на 2 этаже), да еще с другой его стороны (надеюсь, понятно, о чем идет речь), то единственным способом обеспечить прямую видимость антенн является их установка на крышах обоих зданий. Но, предположим, расстояние от корректора, расположенного в подвале, до крыши и расстояние от компьютера до крыши превышает длину фидера (как правило, менее 10 м) и максимальную длину кабеля связи прибора и компьютера с модемами, составляющую по стандарту RS232 не более 15–20 м. Что можно предпринять в такой ситуации?

Одно из решений, предлагаемое автором, — «удлинить» интерфейс RS232, преобразовав его в интерфейс RS485/RS422 с помощью двух одинаковых и достаточно простых устройств. Как известно, длина кабеля связи для интерфейса RS485 может достигать 1200 м. Принципиальная схема подобных устройств — преобразователей интерфейса RS232 в интерфейс RS485 и обратно — была опубликована в [1]. В настоящей статье также приведены примеры несколько модифицированных устройств, использующихся для аналогичных целей.

Конструкция модема

При прочих равных условиях предпочтение следует отдавать такой конструкции модема, чтобы она удовлетворяла хотя бы двум условиям. Во-первых, чтобы разъем сопряжения модема с интерфейсом располагался бы только с одной стороны платы модема и был бы, по возможности, небольшим. Тогда на основной плате, куда включается модем, будет достаточно места для расположения собственных элементов. Если же модем имеет два разъема, то, как правило, между ними уже ничего расположить не удастся, и это место на плате просто пропадет.

Во-вторых, модем должен иметь крепежные отверстия (или свободные места, где их можно просверлить), с помощью которых его можно достаточно прочно прикрепить к основной плате (например, винтами к стойкам с просверленными вдоль них отверстиями с нарезанной резьбой). Кроме того, желательно, чтобы к модему уже был бы припаян разъем для соединения с фидером (например, типа SMA).

Результаты решения задачи

Пример использования модема для беспроводного сопряжения газового корректора Corus с компьютером по интерфейсу RS232

В качестве демонстрации применения вышеописанных рассуждений на практике далее приводится пример использования модема для беспроводного сопряжения газового корректора Corus с компьютером по интерфейсу RS232. Вначале несколько слов о выборе модема. Были проверены и протестированы три модема (правда, в разное время), два из кото-

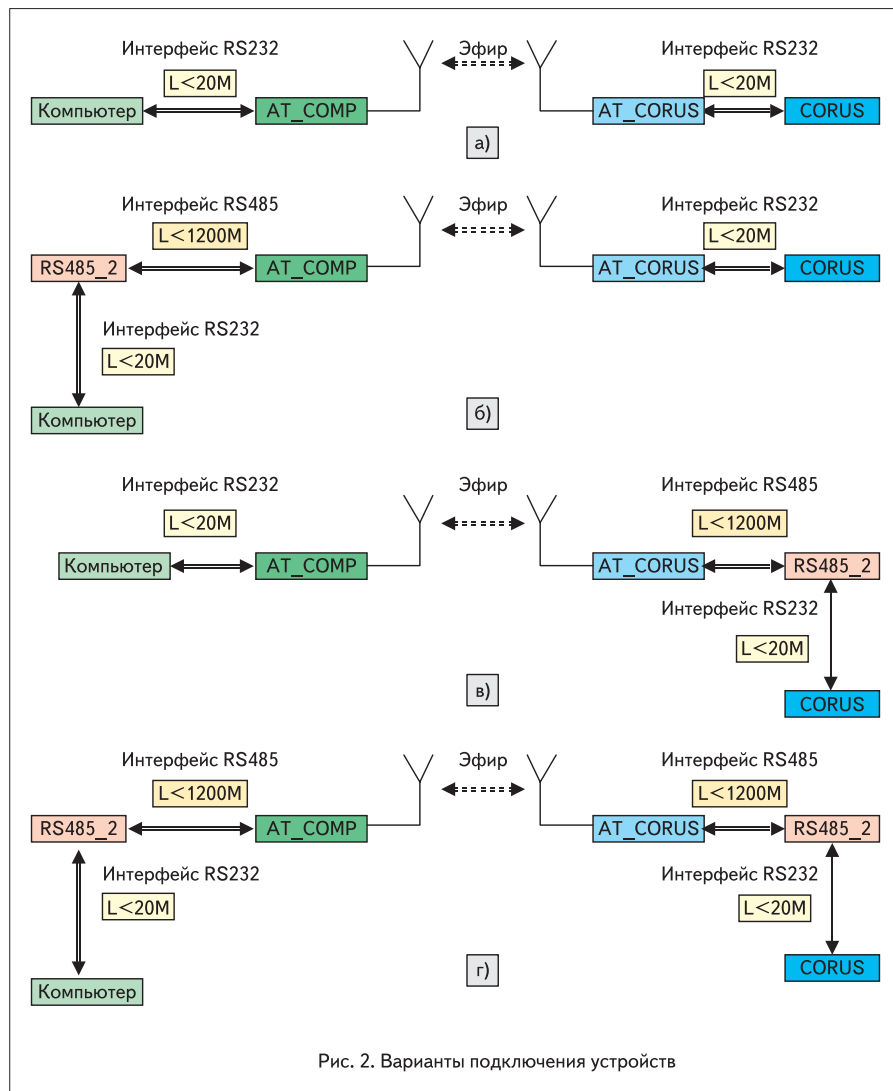


Рис. 2. Варианты подключения устройств

Рис. 2. Варианты подключения устройств

рых — отечественного производства и один импортный. Два из них были отбракованы, так как не прошли тест, о котором упоминалось выше. Единственный модем, с успехом прошедший тест и, тем самым, подходящий для нашего приложения — отечественный. Кроме того, его свойства, как читатель смог уже догадаться, удовлетворяют всем параметрам, указанным ранее.

В результате был разработан комплект устройств, названный Cogus-модем. Ниже представлено краткое техническое описание разработки и пример схемы его включения в общую систему.

Технические характеристики Cogus-модем

Как уже упоминалось выше, комплект устройств Cogus-модем предназначен для беспроводной передачи информации с газового корректора Corus в компьютер по радиоканалу и позволяет производить обмен информацией между компьютером и газовым корректором по радиоканалу в нелегализуемом диапазоне частот 433...435 МГц на расстоянии до 1 км при использовании направленных антенн, находящихся в условиях прямой видимости. Для работы устройств используются стабилизированные источники питания +5 В 0,2 А и +9 В 0,2

А. Для того, чтобы обеспечить требуемую функциональность, был разработан комплект связи, в состав которого входят три устройства: устройство AT_COMP, предназначенное для сопряжения компьютера с OEM-модемом по интерфейсам RS232 и RS485; устройство AT_CORUS, предназначенное для сопряжения газового корректора Corus Actaris с OEM-модемом по интерфейсам RS232 и RS485 и устройство RS485_2 — двунаправленный преобразователь интерфейсов RS232-RS485, предназначенный для сопряжения компьютера с AT_COMP по интерфейсу RS485, а также для сопряжения газового корректора с AT_CORUS по интерфейсу RS485. Кроме того, для обеспечения функционирования системы понадобятся два OEM-модема, направленные антенны для диапазона частот 433...435 МГц, а также вспомогательные комплектующие: необходимые для соединения устройств кабели и разъемы, а также блоки питания с выходными стабилизированными напряжениями +5 В и +9 В. Варианты подключения устройств приведены на рис. 2. Рассмотрим по очереди устройства, входящие в комплект.

Устройство AT_COMP

Устройство AT_COMP заключено в пластиковый корпус и состоит из двух плат: основной платы, на которой расположены все необхо-

