

Как связать микроконтроллер и компьютер по каналу RS-232

Настоящая статья задумывалась как пример реализации разработки микроконтроллерного устройства, управляемого персональным компьютером по последовательному каналу. Она предназначена для тех, кто еще не имеет опыта подобных разработок. Разобравшись с тем, как ПК управляет микроконтроллером, отображает, обрабатывает и сохраняет полученную от него информацию, вы сможете применить эти знания для собственных разработок. К тому же описанное устройство имеет еще и самостоятельную ценность: это управляемый цифровой вольтметр, результаты измерения которого перед отображением могут быть обработаны компьютером по заранее заданному алгоритму, а также сохранены в файле на винчестере вашего ПК, просмотрены и распечатаны. Все это делает описанное устройство основой для простой системы сбора, обработки и документирования данных, полезной для электронщиков, имеющих недостаточный для самостоятельных разработок уровень знания микроконтроллерной техники.

Введение

Целью данной работы была разработка и создание простейшего измерительного устройства на базе микро-

предполагалось реализовать измеритель напряжения, который в дальнейшем мог быть дополнен различными приставками, преобразующими другие непосредственно измеряе-

ты и проводить их компьютерную обработку. Подвергнутое не принципиальным изменениям, оно смогло бы легко превратиться в систему дистанционного контроля и управления оборудованием или иными приборами и устройствами.

Общее описание устройства. Электрическая часть устройства

Устройство (рис. 1), по сути, представляет собой цифровой вольтметр. На входе вольтметра стоит операционный усилитель (DA1), имеющий высокое входное сопротивление. За операционным усилителем следует АЦП (DD2), позволяющий оцифровать интересное нас напряжение для последующей передачи в микроконтроллер. Микроконтроллер DD3 является главным управляющим звеном устройства, так как он считывает информацию из АЦП и общается с персональным компьютером по последовательному каналу. В устройство также входят преобразователи питания для выработки +5 В для цифровой части и для выработки +/-10 В для операционного усилителя, а также микросхема преобразования уровней (логические «0» и «1» в -15...+15 В и обратно) для обмена информацией по последовательному каналу типа RS232.

Значение, посылаемое в компьютер, лежит в диапазоне 0...4095 (что соответствует разрядности АЦП), 0 соответствует входному уровню 0В, 4095 — уровню 5В, зависимость линейная.

Скорость обмена информацией может быть выбрана как меньше 9600 бод, так и выше — до 115 200 бод. На достаточно старых компьютерах, типа 386 и более ранних, верхний предел гораздо

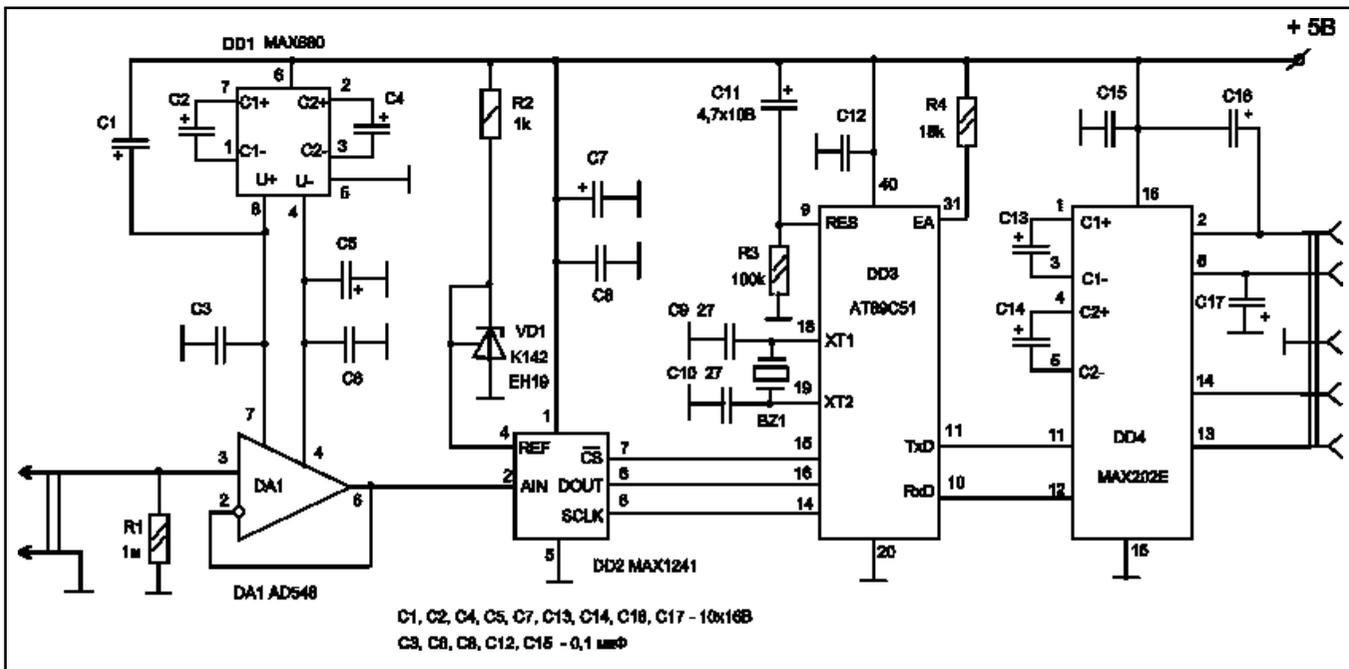


Рис. 1

контроллера все еще самого распространенного на сегодняшний день семейства x51, которое могло бы обмениваться информацией с персональным компьютером. В устройстве

мные физические величины в напряжение. Подобное устройство позволило бы легко проводить серии измерений, будучи управляемым компьютером, а также накапливать результа-

ниже — 19200 бод. Это связано с тем, что микросхемы последовательного порта, установленные в этих компьютерах, не были рассчитаны на более высокие скорости.

Описание микросхем

Преобразователь питающего напряжения MAX680

Операционные усилители обычно требуют подачи на них двуполярного питания (например, +10 В и -10 В относительно общего провода). Радиолюбители, мало знакомые с современной элементной базой, используют обычно для получения такого напряжения трансформатор с двумя вторичными обмотками (или с одной, но с отводом от середины), два фильтрующих конденсатора, два стабилизатора и т. д. Однако если у вас есть в распоряжении стабилизированное напряжение 5В, а используемый операционный усилитель, требующий двуполярного питания, может обойтись всего $\pm 7 \dots 10$ В, потребляя при этом 1...2 мА, то упомянутые две обмотки и два стабилизатора не понадобятся. Достаточно использовать микросхему MAX680 фирмы Maxim (отметим, что подобные микросхемы выпускает Linear Technology и ряд других известных фирм). На вход микросхемы подается напряжение $U_{вх}$ величиной от 3...5 до 6...10 В (в зависимости от типа), на выходах ее формируются напряжения, равные примерно $\pm 2U_{вх}$. Замечательно то, что, во-первых, для формирования этих напряжений помимо 8-выводных MAX680 или LT1026 нужно всего лишь 4 небольших электролитических конденсатора (см. рис. 1), а во-вторых, при изменении входного напряжения удвоенные выходные изменяются синфазно, что практически не сказывается на выходном сигнале ОУ. Для более подробного ознакомления с подобными микросхемами автор рекомендует обратиться к соответствующим фирменным описаниям.

АЦП MAX1241

В последние годы в микроконтроллерной технике получили широкое развитие микросхемы, управляемые по последовательному каналу. Одной из таких микросхем является 12-разрядный АЦП MAX1241. Как и в случае с MAX680, MAX1241 имеет достаточно много точных и приближенных аналогов (MAX187 от Maxim, LTC1286, LTC1298 от Linear Technology, AD7894 от Analog Devices и ряд других).

MAX1241 упакована в 8-выводный корпус, питается напряжением от 2,7 до 5 В, потребляет ток около 5 мА. Она требует применения внешнего источника опорного напряжения (в данном случае применен прецизионный стабилитрон KP142ЕН19, формирующий напряжение 2,50 В) и использует для связи с микроконтроллером всего 3 линии.

Работу MAX1241 иллюстрируют временные диаграммы, изображенные на рис. 2. До начала преобразования и

обмена вход CS# MAX1241 должен поддерживаться микроконтроллером в единичном состоянии. Для старта преобразования на этот вход необходимо подать уровень логического нуля. Процесс преобразования в MAX1241 занимает чуть менее 8 мкс. В течение всего времени преобразования MAX1241 поддерживает на своем выходе DOUT уровень логического 0. После завершения преобразования MAX1241 переводит выход DOUT в единичное состояние.

Перед началом преобразования микроконтроллер на входе SCLK MAX1241 должен установить нулевой логический уровень. Когда процесс преобразования внутри АЦП завершится, микроконтроллер должен сформировать на входе SCLK последовательность не менее чем из 12 положительных импульсов (рис. 2). Фронт первого импульса готовит MAX1241 к передаче данных. По спаду импульса на DOUT появляется в виде логического нуля или единицы старший 12-й бит. Микроконтроллер считывает этот бит, формирует на SCLK фронт второго импульса, а спустя некоторое время — его спад. По спаду второго импульса на DOUT появляется считываемый затем микроконтроллером 11-й бит и т. д.

По спаду 12-го импульса на выходе DOUT устанавливается младший 1-й бит. Спад 13-го импульса переводит DOUT в нулевое состояние, в котором он находится до установки в 1 входа CS#. Переводом CS# в единичное состояние микроконтроллер информирует MAX1241 о завершении процесса чтения результата преобразования. Следующее преобразование MAX1241 может осуществить примерно через 1 мкс после установки CS# в 1.

Алгоритмы работы LTC1286, LTC1298 от Linear Technology и AD7894 от Analog Devices незначительно отличаются от описанного для MAX1241. Более подробно с ними можно ознакомиться, обратившись к соответствующим фирменным описаниям.

Преобразователь уровней MAX202E

Мало для кого является секретом, что в стандартной логике единица представляется уровнем напряжения от 2,4 до 5 В, а ноль — от 0 до 0,8 В. Однако начинающим может быть неизвестно, что при передаче по каналу RS-232 ноль и единица кодируются одинаковыми по величине (от 5 до 12 В), но разными по знаку сигналами. В рамках настоящей статьи не предполагается объяснять, почему принято делать так, а не иначе, — мы ограничимся лишь констатацией этого факта.

Коль скоро для передачи по RS-232 стандартные логические сигналы должны быть преобразованы в сигна-

лы другого уровня, необходимо предусмотреть в схеме соответствующие средства преобразования. Лет 10 назад для этой цели применялись специально разработанные каскады из трех-четырех транзисторов, пары диодов и почти десятка резисторов. Сейчас ситуация значительно изменилась: ведущие производители микросхем выпускают полностью законченные преобразователи, требующие минимального количества дополнительных элементов. К ним относятся MAX202E от MAXIM и полностью идентичная ей, вплоть до цоколевки, AD232 от Analog Devices. Внутри себя обе микросхемы содержат преобразователь напряжения +5 В в ± 10 В, идентичный вышеописанному MAX680, и каскады, осуществляющие преобразование логических сигналов стандартного уровня в сигналы уровня по стандарту RS-232. Каждая из упомянутых микросхем содержит преобразователи логического уровня для двух приемников и двух передатчиков. Мы воспользуемся только одним приемопередающим каналом.

Режим работы МК с последовательным каналом

Как известно (смотри, например, номера 10 и 11 журнала «Радио» за 1994 г.), у микроконтроллеров семейства x51 существуют четыре режима работы приемопередатчика. Нас будет интересовать режим 1 как наиболее простой и приемлемый.

Режим 1 характеризуется следующими параметрами:

- обмен информацией асинхронный;
- передаются 10 бит за один акт обмена (старт-бит (0), 8 бит данных и стоп-бит (1));
- скорость приема/передачи программируема и задается таймером.

Это удобный режим для программирования: требуется очень немного программного кода для настройки приемопередатчика и работы с ним. Хотя по желанию можно использовать и другие режимы работы. Целью же данной статьи является описание некоего устройства, имеющего возможность общаться с персональным компьютером.

Мы не будем приводить здесь описание того, как именно работает приемопередатчик. Эту информацию можно будет почерпнуть из упомянутых журналов «Радио» или другой литературы.

Основные

подпрограммы для МК

Основными подпрограммами для микроконтроллера будут являться: считывание данных из АЦП, инициализация УАПП, прием байта и посылка байта.

Считывание данных из АЦП

Настройка микроЭВМ для обмена информацией по последовательному каналу

Для того чтобы настроить микроЭВМ на обмен информацией по последовательному каналу, необходимо сделать следующее:

- запретить все прерывания, так как в их использовании необходимости нет;
- настроить таймер 1 на работу в режиме 2 с авто-

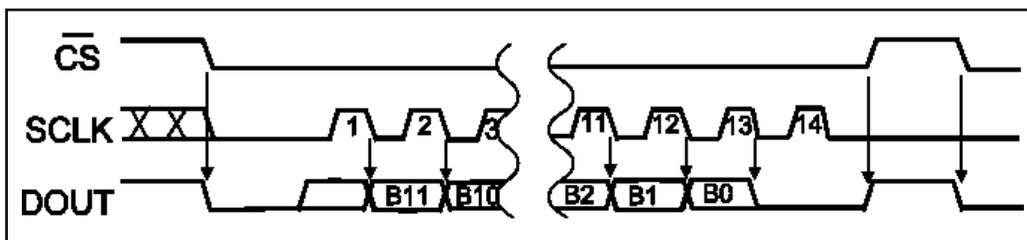


Рис. 2

```

GET_VOLT:
SETB DOUT ;РАЗРЕШИЛИ ВВОД ДАННЫХ ИЗ ADC
SETB CS ;УСТАНОВИЛИ НАЧАЛЬНОЕ СОСТОЯНИЕ ADC
CLR SCLK ;УСТАНОВИЛИ НАЧАЛЬНОЕ СОСТОЯНИЕ ADC

CLR CS ;СООБЩИЛИ О ЖЕЛАНИИ ПРОЧЕСТЬ ДАННЫЕ

MUL AB ;4 МКС НА 12 МНЗ \
MUL AB ;4 МКС
MUL AB ;4 МКС } ДОЖДАЛИСЬ КОНЦА
; ; } ОЦИФРОВКИ
MUL AB ;4 МКС
MOV R0, #12 ;СЧИТЫВАТЬ 12 БИТ

GET_VC:
SETB SCLK ;\
NOP ;/
NOP ;/
CLR SCLK ;} СФОРМИРОВАЛИ ИМПУЛЬС ДЛЯ ЧТЕНИЯ БИТА
NOP ;/
NOP ;/
MOV C, DOUT ;ПРОЧИТАЛИ БИТ
MOV A, R2 ;\
RLC A ;/
MOV R2, A ;/
MOV A, R3 ;} ЗАДВИНУЛИ БИТ В СЛОВО
MOV A, R3 ;} РЕЗУЛЬТАТА — R3R2
RLC A ;/
MOV R3, A ;/
DJNZ R0, GET_VC ;ЗАЦИКЛИВАЕМСЯ
ANL A, #0FH
MOV R3, A ;ОЧИСТИЛИ СТАРШИЕ БИТЫ R3R2

SETB CS ;БОЛЬШЕ НЕ ХОТИМ СЧИТЫВАТЬ
; (ОСТАЛЬНЫЕ БИТЫ=0)
MUL AB ;4 МКС НА 12 МНЗ \
MUL AB ;4 МКС
MUL AB ;4 МКС
MUL AB ;4 МКС } МИН ЗАДЕРЖКА
; ; } ПЕРЕД СЛЕД.
MUL AB ;4 МКС
MUL AB ;4 МКС
RET

```

матической перезагрузкой содержимого. Это необходимо для задания скорости обмена информацией и поддержания ее постоянной;

- загрузить в счетчик таймера начальные значения;
- запустить таймер 1.

Пример кода, рассчитанного на скорость обмена 9600 бит/с для кварцевого резонатора с резонансной частотой равной 11,059 МГц:

Эта подпрограмма вызывается самой первой в основной программе микроЭВМ. В принципе ее можно даже и не оформлять как подпрограмму.

Прием и посылка байта

Подпрограммы приема и посылки байта по последовательному каналу очень просты.

```

SERINIT:
MOV IE, #0 ;Запретить все прерывания
MOV TMOD, #20H ;Установить режим 2 для таймера 1
MOV TH1, #REL96 ;Значение для автоперезагрузки счетчика
MOV TL1, #REL96 ;Начальное значение счетчика для 9600
; бит/с
; при SMOD=0
ANL PCON, #7FH ;Очистили SMOD
MOV SCON, #50H ;Режим для 8 бит данных и скорости
; передачи,
; зависящей от таймера
SETB TR1 ;Старт таймера/счетчика 1
RET
где REL96 — константа, равная 0FDh.

```

Считывать байт из порта ввода/вывода SBUF можно только при установленном бите RI регистра управления/статуса SCON, сигнализирующего о наличии байта в буфере приема. После считывания этого байта бит RI необходимо сбросить.

После записи байта в порт ввода/вывода нужно дождаться установления бита TI, который будет сигнализировать окончание посылки байта в линию. Затем бит TI также будет нужно сбросить.

Подпрограмма приема байта в аккумулятор:

```

GETCH:
JNB RI, GETCH
MOV A, SBUF
CLR RI
RET

```

Подпрограмма посылки байта из аккумулятора:

```

PUTCH:
MOV SBUF, A
SEND:

```

```

JNB TI, SEND
CLR TI
RET

```

Следует также отметить, что никаких средств для обнаружения ошибок ввода/вывода микроЭВМ не имеет.

Для того чтобы организовать проверку программно-аппаратным образом, можно расширить количество линий ввода/вывода, по которым будут передаваться дополнительные сигналы, и по ним можно будет определять состояния, в которых находятся участники диалога, а также выявлять ошибки.

Можно повысить надежность приема/передачи информации и другим путем: передавать с восемью битами данных еще один бит — бит четности, вычисляющийся аналогично флагу паритета в слове состояния программы (бит 0 PSW). Только вычисляться он должен для передаваемого или принятого байта. После принятия байта и бита четности необходимо сравнить их на соответствие друг другу. Если они не соответствуют, значит, имела место ошибка ввода/вывода. Для передачи дополнительного 9-го информационного бита нужно использовать режим 2 или 3 работы таймера/счетчика.

Общая программа для МК. Диаграмма состояний устройства

Общая программа для микро-ЭВМ базируется на нижеописанном алгоритме. Алгоритм довольно непростой, т.к. все же нужно каким-то образом, хотя бы программным методом, выявлять ошибки ввода/вывода и реагировать на их появление.

Для большей наглядности к алгоритму, описанному обычными словами, прилагается рисунок — так называемая диаграмма состояния устройства (рис.3), на которой приведены четыре основных состояния устройства с точки зрения обмена информацией с ЭВМ.

Заранее оговорим тот факт, что наша микро-ЭВМ является ведомой, а персональный компьютер — ведущим при обмене данными. Иными словами, устройство само по себе, без приказа от ПК, ничего делать не должно. Оно всецело подчиняется управляющему компьютеру. Персональный компьютер выбран ведущим по той простой причине, что он обладает большей мощностью и способен без особых проблем управлять устройством. Кроме того, он может дать пользователю больше сервисных функций.

Состояние первое — Wait

В этом состоянии устройство оказывается сразу же после включения питающего напряжения. Здесь оно ожидает от компьютера запроса на инициализацию, который выражается в посылке компьютером символа NUL. Устройство же, в свою очередь, должно в ответ на полученный запрос включить и настроить, если требуется, дополнительные модули и ресурсы, а затем, если все прошло нормально, послать в ЭВМ символ ACK. В случае же ошибки оно должно послать NAK. Таким образом, происходит первое «общение» двух «собеседников». Если хотите, они должны «обменяться приветствиями» или «пожать друг другу руки».

При удачной инициализации устройства с последующей посылкой символа ACK, оно автоматически переходит в следующее состояние. Этот переход обозначен стрелкой 1 на диаграмме.

Состояние Ready

В этом состоянии наша микро-ЭВМ ожидает запроса ПК на посылку измеренного значения, считанного с АЦП. Запросом является символ XON. По принятии этого символа устройство переходит в новое состояние — Sending. Переходу соответствует стрелка 2.

Состояние Sending.

Попадая сюда, микроконтроллер считывает двоичное двенадцатиразрядное число из АЦП ранее указанным методом и посылает частями в ЭВМ. В данной реализации происходит преобразование двоичного числа в трехсимвольный шестнадцатеричный эквивалент, например, в «1FF» для десятичного числа 511. Сначала посылается «1», затем «F» и еще один «F».

По окончании передачи значения в компьютер микро-ЭВМ переходит в следующее состояние по стрелке 4.

Состояние Sent

Это состояние является последним и как бы замыкает круг единичного акта общения устройства с компьютером. Здесь от компьютера ожидается подтверждение того, что он правильно принял значение, которое было ему адресовано.

Тут возможными являются несколько вариантов ответа ПК на посланное число: он может ответить об успешном приеме

символом XOFF, который будет означать, что больше пока не требуется других значений, а может ответить символом XON, означающим, что нужно еще одно значение. Если принят XOFF, то устройство возвращается в состояние готовности Ready (переход 7 на диаграмме). Если же принят символ XON, то устройство опять оказывается в состоянии Sending (переход 5) и повторяет считывание из АЦП с последующей передачей числа в линию. Не рассмотренным оказался лишь тот случай, когда ПК не понравится то, что он получил: например, вместо символов диапазона «0»...«9», «А»...«F», он получил нечто иное, «G» или «,». В этой ситуации он посылает нашему устройству символ NAK, который должен трактоваться как запрос на повторную посылку последнего значения, что и происходит: устройство опять переводится в состояние Sending (переход по стрелке 6).

Остались не описанными переходы, обозначенные стрелками 3 и 8 диаграммы. Если компьютер обнаружит серьезную ошибку ввода/вывода или ему понадобится прекратить обмен с устройством, то он просто пошлет инициализационный NUL, по которому произойдет инициализация устройства и оно окажется в состоянии готовности Ready.

Т.е. в каком бы состоянии не находилось наше устройство, оно обязано ответить на инициализационный запрос, таким же образом, как и при первичной инициализации (см. Пункт состояние Wait). Если же микро-ЭВМ получила какой-то неожиданный или неверный символ или запрос, то оно всегда должно ответить на него символом NAK.

Такая стратегия является выигрышной, так как при подобная организация программы для устройства облегчает выполнение сразу нескольких задач: микро-ЭВМ и ПК не будут играть в испорченный телефон, во-первых, а, во-вторых, они смогут просто и эффективно «общаться» друг с другом.

Переходим к ПК. Общая программа для ПК. Диаграмма состояний ПК

Принципиально общая программа для компьютера не будет ни чем отличаться от используемой в микроконтроллере. Алгоритм будет аналогичным, похожей будет и диаграмма состояний, которая представлена на рисунке 4.

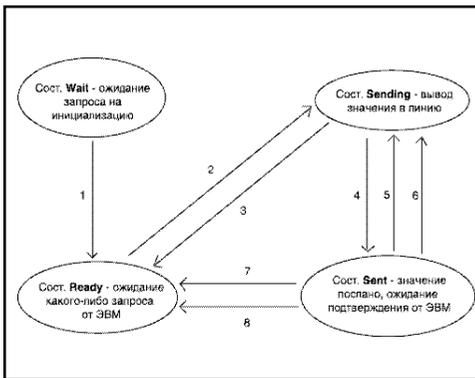


Рис. 3

Состояние Receiving

Здесь ПК просто считывает три символа значения, измеренного и преобразованного с помощью АЦП. Как было сказано ранее, существует некоторое объективное время ожидания символа компьютером. Если символ не был считан за это время, то такая ситуация интерпретируется как ошибочная, т.е. имела место ошибка ввода/вывода.

Кстати говоря, при довольно высоких скоростях обмена информацией (больше 19200 бит/с) или при работе в операционной системе MS-Windows (любой версии) часто бывает, что компьютер из посылаемых ему трех символов принимает только два, а иногда и того меньше — один. Чтобы компьютер «не висел» — ожидал бесконечно долго недостающего или пропущенного символа — вводится некоторое время, ограничивающее это ожидание. К сожалению, эти пропуски аппаратным методом никак не выявляются.

В данной реализации определено два типа времени ожидания, которые могут быть заданы пользователем с клавиатуры. Первый тип — это время ожидания 1-го из 3-х символов. Он позволяет устройству спокойно, никуда не торопясь, измерить, оцифровать необходимое нам число и преобразовать его в символьный эквивалент. А второй тип — это временной лимит на посылку второго и третьего символов.

Перейдем теперь к возможным переходам из состояния Receiving в другие состояния.

Если так и не было принято всех 3-х символов значения за отведенное время, то компьютер должен попросить наше устройство послать ему значение повторно. Этой ситуации соответствует переход по стрелке 5, т.е. компьютер делает запрос символом NAK и переходит обратно в состояние Ready.

Если в процессе приема компьютером была зафиксирована ошибка ввода/вывода (а у ПК есть такая возможность достаточно проанализировать регистр состояния последовательного порта), то лучше привести и компьютер и микро-ЭВМ в исходное состояние, т.е. повторить инициализацию. Поэтому на диаграмме также присутствует и стрелка 3.

И, наконец, если компьютер получил от устройства все три символа, то он переходит в состояние анализа полученного значения — в состояние Received по стрелке 8.

Алексей Фрунзе, Москва

Продолжение следует

Первое состояние Initialization

Сюда компьютер попадает, когда пользователь нажимает на его клавиатуре клавишу, соответствующую принятию единственного значения. В этом состоянии компьютер посылает символ инициализации NUL в устройство и ожидает ответа на него символом ACK или NAK. Если был получен ACK, значит инициализация прошла нормально и можно продолжить работу — перейти в следующее состояние по стрелке 2 на диаграмме. В случае получения NAK работа должна прекратиться и компьютеру следует перейти в заключительное состояние Done по стрелке 1.

Состояние Ready

В этом состоянии компьютер подготавливается к приему символов, из которых будет состоять запрошенное из микро-ЭВМ значение. Запросов на посылку значения существует два. Первый — это обычный запрос значения, ему соответствует символ XON.

Второй же запрос — это запрос на повторную посылку последнего значения. Это необходимо в том случае, если значение не было принято полностью за какое-то объективное время или были приняты неверные символы, не попадающие в диапазоны от «0» до «9» и от «А» до «F». Для запроса на повторную посылку отводится символ NAK.

Далее, после подготовки к приему символов значения происходит один из двух вышеуказанных запросов к нашему устройству, затем компьютер переходит по стрелке 4 в состояние приема значения.

ЭЛЕКТРОННЫЕ КОМПОНЕНТЫ
ВАШЕГО УСПЕХА

www.platan.ru

• ВИРТУАЛЬНЫЙ МАГАЗИН РЕАЛЬНЫХ ЭЛЕКТРОННЫХ КОМПОНЕНТОВ

Наш сайт предоставляет широкие возможности оформления заявок на поставку электронных компонентов:

- Вы можете разместить заявку на требуемые компоненты и отслеживать ее прохождение на всех этапах. Очень удобный интерфейс и абсолютная безопасность!
- Воспользуйтесь нашей удобной системой приема заявок на компоненты, которые находятся на складе или на которые можно разместить заказ.
- С нашего сайта Вы можете скачать нашу базу в формате EXCEL, проставить нужное Вам количество компонентов и отправить этот файл по нашему адресу.

ВСЕ ЗАЯВКИ, ПОЛУЧЕННЫЕ В ЭЛЕКТРОННОМ ВИДЕ, ОБРАБАТЫВАЮТСЯ В ПЕРВУЮ ОЧЕРЕДЬ.

• НОВОСТИ, ТЕХНИЧЕСКАЯ ИНФОРМАЦИЯ, СТАТЬИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Новые поступления на склад ПЛАТАН, новости от мировых производителей. Подробная техническая документация на поставляемые компоненты, статьи и книги по электронной тематике. Бесплатное программное обеспечение.

MITSUBISHI
ELECTRIC

Infineon
TECHNOLOGIES

BOURNS

EPCOS

ROHM

MOTOROLA

intersil

SHARP

DATA VISION

Kingbright

JANSON

121351, Москва, ул. Ивана Франко, д. 40, к. 1, стр.2
тел./факс: (095) 417-52-45, 417-08-11, 417-86-45

Почта: 121351, Москва, а/я 100
E-mail: ir@platan.ru