

Термометр на основе DS18S20

Широко распространенная микросхема цифрового термометра DS18S20, выпускаемая фирмой DALLAS, обеспечивает измерение температуры в диапазоне $-55..+125^{\circ}\text{C}$ с дискретностью 0.5°C . Стоимость микросхемы DS18S20 составляет примерно 2\$, стоимость деталей адаптера для подключения её к COM-порту компьютера - еще меньше. Подробное описание микросхемы DS18S20 можно найти по ссылке: <http://pdfserv.maxim-ic.com/arpdf/DS18S20.pdf>.

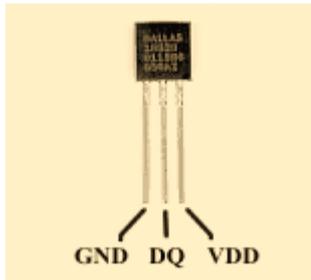


Рис. 1. Внешний вид микросхемы цифрового термометра DS18S20.

С помощью дополнительных вычислений дискретность представления температуры можно уменьшить, в нашем случае она равна 0.1°C . Самым привлекательным является то, что такой термометр уже откалиброван на заводе, гарантированная точность составляет $\pm 0.5^{\circ}\text{C}$ в диапазоне $-10..+85^{\circ}\text{C}$ и $\pm 2^{\circ}\text{C}$ во всем диапазоне рабочих температур. Типичная кривая ошибки измерения температуры приведена на рис. 2.

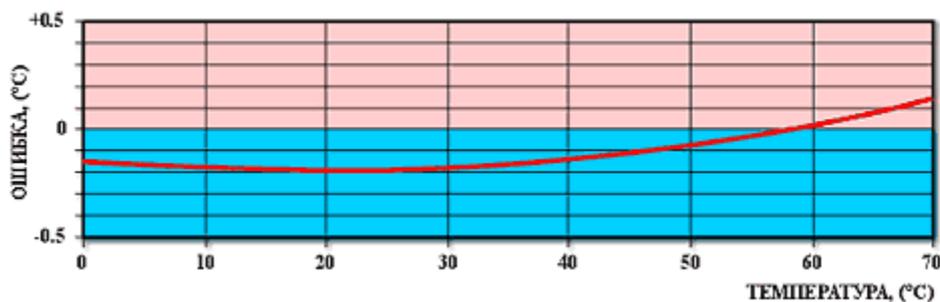


Рис. 2. Типичная кривая ошибки термометра DS18S20.

Несмотря на ограниченную абсолютную точность, малая дискретность представления температуры является весьма желательной, так как очень часто на практике требуются относительные измерения.

DS18S20 допускает напряжение питания от +3 до +5.5В. В режиме ожидания потребляемый ток близок к нулю (менее 1мкА), а во время преобразования температуры он равен примерно 1мА. Процесс преобразования длится максимум 750мс.

Принцип действия цифровых датчиков температуры фирмы DALLAS основан на подсчете количества импульсов, вырабатываемых генератором с низким температурным коэффициентом во временном интервале, который формируется генератором с большим температурным коэффициентом. Счетчик инициализируется значением, соответствующим -55°C (минимальной измеряемой температуре). Если счетчик достигает нуля перед тем, как заканчивается временной интервал (это означает, что температура больше -55°C), то регистр температуры, который также инициализирован значением -55°C , инкрементируется. Одновременно счетчик предустанавливается новым значением, которое задается схемой формирования наклона характеристики. Эта схема нужна для компенсации параболической зависимости частот генераторов от температуры. Счетчик снова начинает работать, и если он опять достигает нуля, когда интервал еще не закончен, процесс повторяется снова. Схема формирования наклона загружает счетчик значениями, которые соответствуют количеству импульсов генератора на один градус Цельсия для каждого конкретного значения температуры. По окончании процесса преобразования регистр температуры будет содержать значение температуры.

Для DS18S20 температура представляется в виде 9-битного значения в дополнительном коде. Поскольку это значение занимает 2 байта, все разряды старшего байта равны знаковому разряду. Дискретность представления температуры составляет 0.5°C . Зависимость выходного кода от температуры приведена в таблице:

Температура	Выходной код (Binary)		Выходной код (Hex)
	Ст. байт	Мл. байт	
+125°C	0000 0000	1111 1010	00FAh
+25°C	0000 0000	0011 0010	0032h
+0.5°C	0000 0000	0000 0001	0001h
0°C	0000 0000	0000 0000	0000h
-0.5°C	1111 1111	1111 1111	FFFFh
-25°C	1111 1111	1100 1110	FFCEh
-55°C	1111 1111	1001 0010	FF92h

Более высокая разрешающая способность может быть получена, если произвести дополнительные вычисления на основе значений COUNT_REMAIN (значение, оставшееся в счетчике в конце измерения) и COUNT_PER_C (количество импульсов на один градус для данной температуры), которые доступны. Для вычислений требуется взять считанное значение

температуры и отбросить младший бит. Полученное значение назовём TEMP_READ. Теперь действительное значение температуры может быть вычислено по формуле:

$$\text{TEMPERATURE} = \text{TEMP_READ} - 0.25 + (\text{COUNT_PER_C} - \text{COUNT_REMAIN}) / \text{COUNT_PER_C}$$

В нашем случае такой расчет позволяет получить дискретность представления температуры 0.1°C.

Каждый экземпляр DS18S20 имеет уникальный 48-битный номер, записанный с помощью лазера в ПЗУ в процессе производства. Этот номер используется для адресации устройств. Кроме серийного номера в ПЗУ содержится код семейства (для DS18S20 это 10h) и контрольная сумма.

Кроме ПЗУ DS18S20 имеет промежуточное ОЗУ объемом 8 байт, плюс два байта энергонезависимой памяти. Карта памяти DS18S20 показана на рисунке:

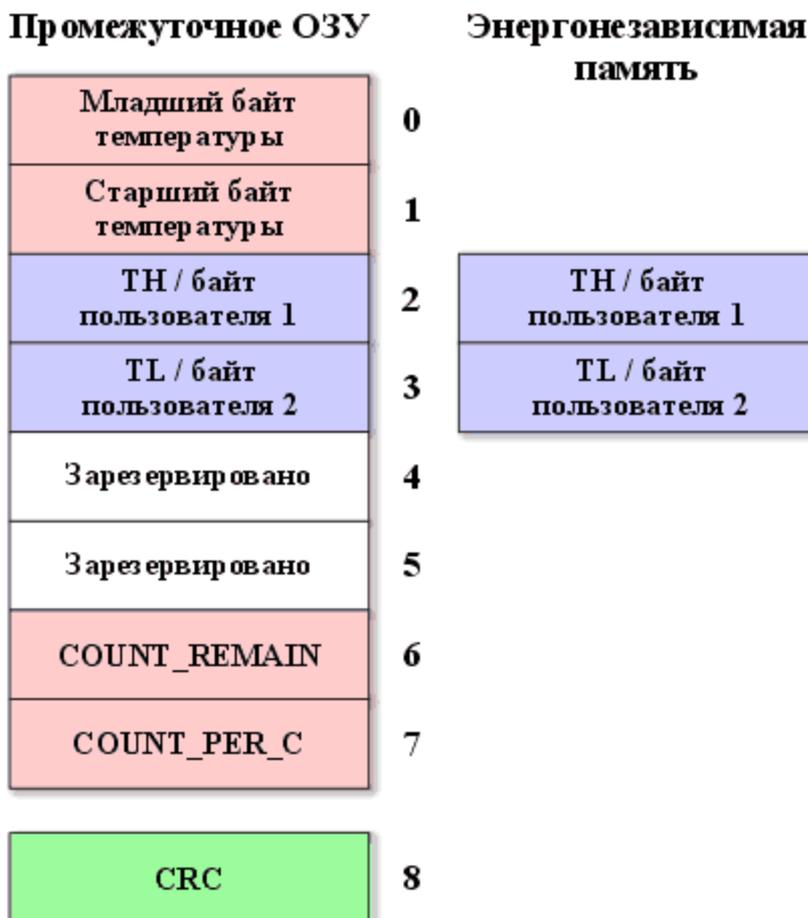


Рис. 3. Карта памяти DS18S20.

Байты ТН и ТL представляют собой температурные пороги, с которыми сравниваются 8 бит каждого измеренного значения температуры (младший бит отбрасывается). С помощью специальной команды можно организовать сигнализацию выхода температуры за пределы этих порогов. Если такая функция не нужна, байты ТН и ТL можно использовать для хранения любых данных пользователя.

Считывание значения измеренной температуры, а также передача команды начала преобразования и других команд производится с помощью 1-проводного интерфейса (1-Wire™) фирмы DALLAS. На основе этого интерфейса фирма DALLAS даже создала сеть, называемую microLAN (или µLAN). Для работы в этой сети выпускается целый ряд устройств, таких как адресуемые ключи, АЦП, термометры, часы реального времени, цифровые потенциометры. Кстати, такой же протокол обмена имеют и цифровые ключи IButton (или Touch Memory), которые сейчас широко используются в системах ограничения доступа.

Протокол, который используется 1-проводным интерфейсом, достаточно прост. В любой момент времени на 1-проводной шине можно выделить устройство-мастер, которым может быть микропроцессор или компьютер, и подчиненное устройство, в нашем случае это микросхема термометра. Так как у нас на шине присутствуют только мастер и всего одно подчиненное устройство, можно опустить всё то, что связано с адресацией устройств. В результате требуется знать лишь протокол передачи байтов, которые могут являться командами или данными.

Вначале рассмотрим аппаратную конфигурацию. 1-проводная шина является двунаправленной. На рис. 4 показана аппаратная конфигурация интерфейсной части DS18S20 и мастера шины. У каждого 1-проводного устройства к шине подключен вход приемника и выход передатчика с открытым стоком. Открытый сток позволяет подключать к шине множество устройств, обеспечивая логику «монтажное или». Генератор тока 5мкА обеспечивает на входе 1-проводного устройства низкий логический уровень, когда шина не подключена. Так как линия тактового сигнала отсутствует, обмен является синхронным. Это означает, что в процессе обмена нужно достаточно точно выдерживать требуемые временные соотношения.

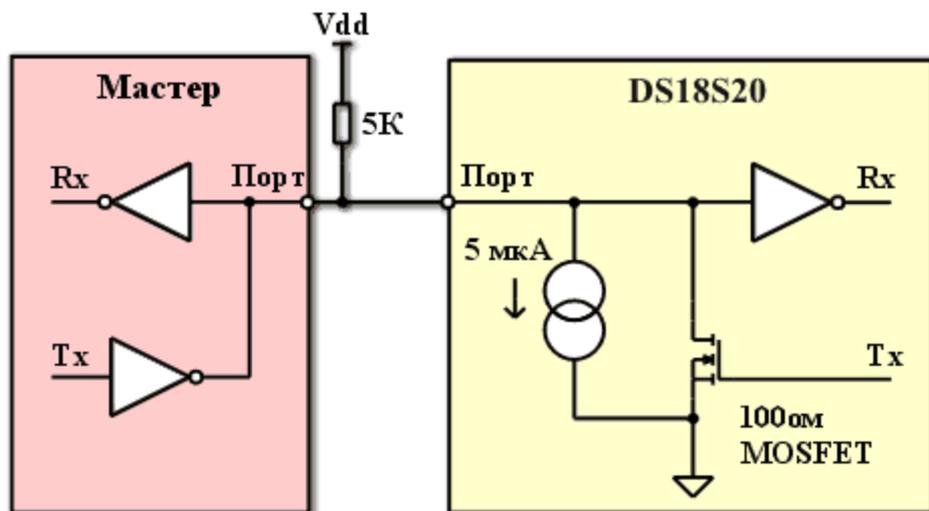


Рис. 4. Аппаратная конфигурация интерфейсной части 1-проводных устройств.

1-проводная шина оперирует с TTL-уровнями, т.е. логическая единица представлена уровнем напряжения около 5В, а логический ноль – напряжением вблизи 0В. В исходном состоянии на линии присутствует уровень логической единицы, который обеспечивается подтягивающим резистором номиналом около 5Ком.

Инициатором обмена по 1-проводной шине всегда выступает мастер. Все пересылки начинаются с процесса инициализации. Инициализация производится в следующей последовательности (рис. 5):

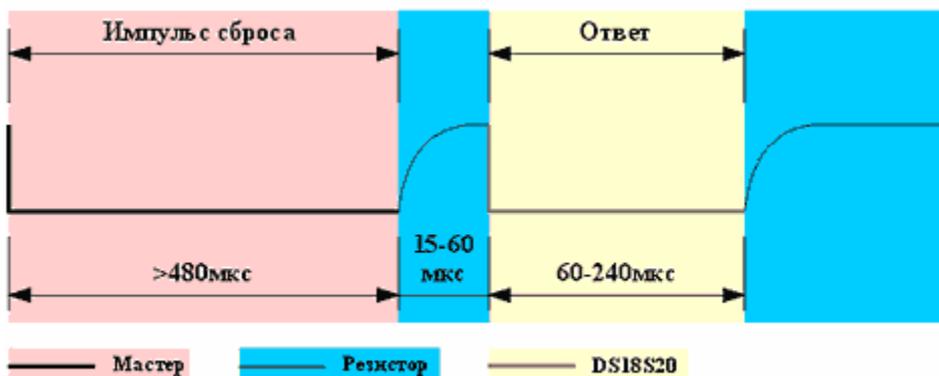


Рис. 5. Инициализация обмена по 1-проводной шине.

- Мастер посылает импульс сброса (reset pulse) - сигнал низкого уровня длительностью не менее 480 мкс.
- За импульсом сброса следует ответ подчиненного устройства (presence pulse) - сигнал низкого уровня длительностью 60 - 240 мкс, который генерируется через 15 - 60 мкс после завершения импульса сброса.

Ответ подчиненного устройства даёт мастеру понять, что на шине присутствует термометр и он готов к обмену. После того, как мастер обнаружил ответ, он может передать термометру одну из команд. Передача ведётся путём формирования мастером специальных временных интервалов (time slots). Каждый временной интервал служит для передачи одного бита. Первым передаётся младший бит. Интервал начинается импульсом низкого уровня, длительность которого лежит в пределах 1 - 15 мкс. Поскольку переход из единицы в ноль менее чувствителен к ёмкости шины (он формируется открытым транзистором, в то время как переход из нуля в единицу формируется подтягивающим резистором), именно этот переход используют 1-проводные устройства для синхронизации с мастером. В подчиненном устройстве запускается схема временной задержки, которая определяет момент считывания данных. Номинальное значение задержки равно 30 мкс, однако, оно может колебаться в пределах 15 - 60 мкс. За импульсом низкого уровня следует передаваемый бит. Он должен удерживаться мастером на шине в течение 60 - 120 мкс от начала интервала. Временной интервал завершается переводом шины в состояние высокого уровня на время не менее 1 мкс. Нужно отметить, что ограничение на это время сверху не накладывается. Аналогичным образом формируются временные интервалы для всех передаваемых битов (рис. 6):



Рис. 6. Передача бита по 1-проводной шине.

Первой командой, которую должен передать мастер для DS18S20 после инициализации, является одна из команд функций ПЗУ. Всего DS18S20 имеет 5 команд функций ПЗУ:

- **Read ROM [33h].** Эта команда позволяет прочитать содержимое ПЗУ. В ответ на эту команду DS18S20 передает 8-битный код семейства (10h), затем 48-битный серийный номер, а затем 8-битную CRC для проверки правильности принятой информации.
- **Match ROM [55h].** Эта команда позволяет адресовать на шине конкретный термометр. После этой команды мастер должен передать нужный 64-битный код, и только тот термометр, который имеет такой код, будет «откликаться» до следующего импульса сброса.
- **Skip ROM [CCh].** Эта команда позволяет пропустить процедуру сравнения серийного номера и тем самым сэкономить время в системах, где на шине имеется всего одно устройство.
- **Search ROM [F0h].** Эта довольно сложная в использовании команда позволяет определить серийные номера всех термометров, присутствующих на шине.
- **Alarm Search [ECh].** Эта команда аналогична предыдущей, но «откликаться» будут только те термометры, у которых результат последнего измерения температуры выходит за предустановленные пределы TH и TL.

Поскольку у нас всего одно устройство, наиболее подходящей для нас функцией является функция Skip ROM. Кроме неё ещё может быть полезной функция Read ROM, которая позволяет идентифицировать подключенное на шину устройство по его коду семейства и серийному номеру.

Приняв команду Read ROM, DS18S20 будет готов передать 64-битный код, который мастер должен принять.

При приеме данных от подчиненного устройства временные интервалы для принимаемых битов тоже формирует мастер. Интервал начинается импульсом низкого уровня длительностью 1 - 15 мкс. Затем мастер должен освободить шину, чтобы дать возможность термометру вывести бит данных. По переходу из единицы в ноль DS18S20 выводит на шину бит данных и запускает схему временной задержки, которая определяет, как долго бит данных будет присутствовать на шине. Это время лежит в пределах 15 - 60 мкс. Для того чтобы данные на шине, которая всегда обладает некоторой ёмкостью, гарантированно установились, требуется некоторое время. Поэтому момент считывания данных мастером должен отстоять как можно дальше, но не более чем на 15 мкс от начала временного интервала (Рис 7):



Рис. 7. Чтение бита по 1-проводной шине.

Прием байта начинается с младшего бита. Вначале идет байт кода семейства. За кодом семейства идет 6 байт серийного номера, начиная с младшего. Затем идет байт контрольной суммы (CRC). В вычислении байта контрольной суммы принимают участие первые 7 байт, или 56 передаваемых бит. Для вычисления используется следующий полином:

$$CRC = X^8 + X^5 + X^4 + 1$$

После приема данных мастер должен вычислить контрольную сумму и сравнить получившееся значение с переданной CRC. Если эти значения совпадают, значит, прием данных прошел без ошибок. Можно также вычислить контрольную сумму для всех 64 принятых бит, которая в этом случае должна быть равна нулю. Блок-схема алгоритма вычисления контрольной суммы показана на рис. 8. Алгоритм использует операции сдвига и «исключающего или». Квадратиками показаны биты переменной, которая используется для вычисления CRC. Перед вычислением её необходимо обнулить, а затем на вход алгоритма нужно последовательно подать 56 принятых бит в том порядке, в котором они были приняты. В результате переменная будет содержать значение CRC.

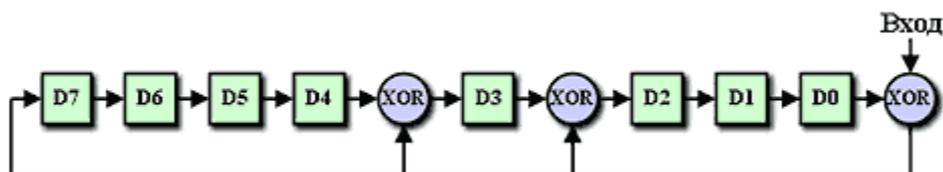


Рис. 8. Блок-схема алгоритма вычисления контрольной суммы.

Такой же алгоритм вычисления контрольной суммы используется и в случае чтения промежуточного ОЗУ, только там считанная из термометра CRC (9-й байт) рассчитана для 8-ми байтов данных.

После обработки одной из команд функций ПЗУ, DS18S20 способен воспринимать еще несколько команд:

- **Write Scratchpad [4Eh].** Эта команда позволяет записать данные в промежуточное ОЗУ DS18S20.
 - **Read Scratchpad [BEh].** Эта команда позволяет считать данные из промежуточного ОЗУ.
 - **Copy Scratchpad [48h].** Эта команда копирует байты TH и TL из промежуточного ОЗУ в энергонезависимую память. Эта операция требует около 10мс.
 - **Convert T [44h].** Эта команда запускает процесс преобразования температуры.
 - **Recall E2 [B8h].** Эта команда действует обратным образом по отношению к команде Copy Scratchpad, т.е. она позволяет считать байты TH и TL из энергонезависимой памяти в промежуточное ОЗУ. При включении питания эта команда выполняется автоматически.
 - **Read Power Supply [B4h].** Эта команда позволяет проверить, использует ли DS18S20 паразитное питание. Дело в том, что DS18S20 можно подключать всего с помощью двух проводов, в этом случае для питания используется линия данных. Особенности этого режима мы здесь рассматривать не будем.
- При использовании DS18S20 только для измерения температуры нужны всего две из этих команд: Convert T и Read Scratchpad.

Последовательность действий при измерении температуры должна быть следующей:

- Пышалаем импульс сброса и принимаем ответ термометра.
- Пышалаем команду Skip ROM [CCh].
- Пышалаем команду Convert T [44h].
- Формируем задержку минимум 750мс.
- Пышалаем импульс сброса и принимаем ответ термометра.
- Пышалаем команду Skip ROM [CCh].
- Пышалаем команду Read Scratchpad [BEh].
- Читаем данные из промежуточного ОЗУ (8 байт) и CRC.
- Проверяем CRC, и если данные считаны верно, вычисляем температуру.

Для подключения DS18S20 к COM-порту компьютера используется адаптер, схема которого приведена на рис. 9, где показано окно помощи программы.

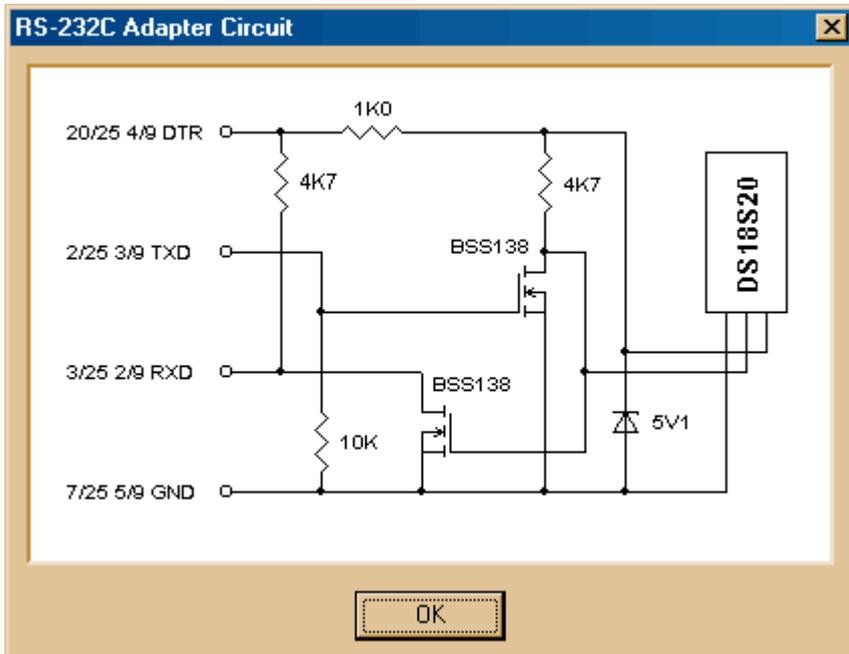


Рис. 9. Окно помощи программы со схемой адаптера.

Схема этого адаптера не так проста, как, например, схема адаптера DS9097 фирмы Dallas. Это связано в первую очередь с тем, что хотелось иметь общую «землю» компьютера и 1-проводной шины. Для питания DS18S20 используется линия DTR последовательного порта. Адаптер обеспечивает на входе RXD порта компьютера лишь однополярные уровни, что, строго говоря, не соответствует спецификации RS-232C. Однако большинство портов с такими уровнями работают нормально. Вместо указанных на схеме n-канальных МОП-транзисторов можно применить близкие по параметрам транзисторы других типов, например, 2N7000. Подойдут также отечественные транзисторы КП501 или КП505. В принципе, можно применить и биполярные транзисторы, добавив в базы ограничительные резисторы. Конструктивно адаптер выполнен в корпусе разъема D-SUB-25 (рис. 10):

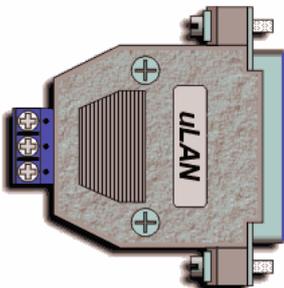


Рис. 10. Внешний вид адаптера.

Разъем паяется прямо на плату, которая входит между рядами контактов. Другая сторона платы выполнена удлиненной и выходит за пределы корпуса разъема. В эту часть платы впаян 3-х контактный винтовой терминал (рис. 11):

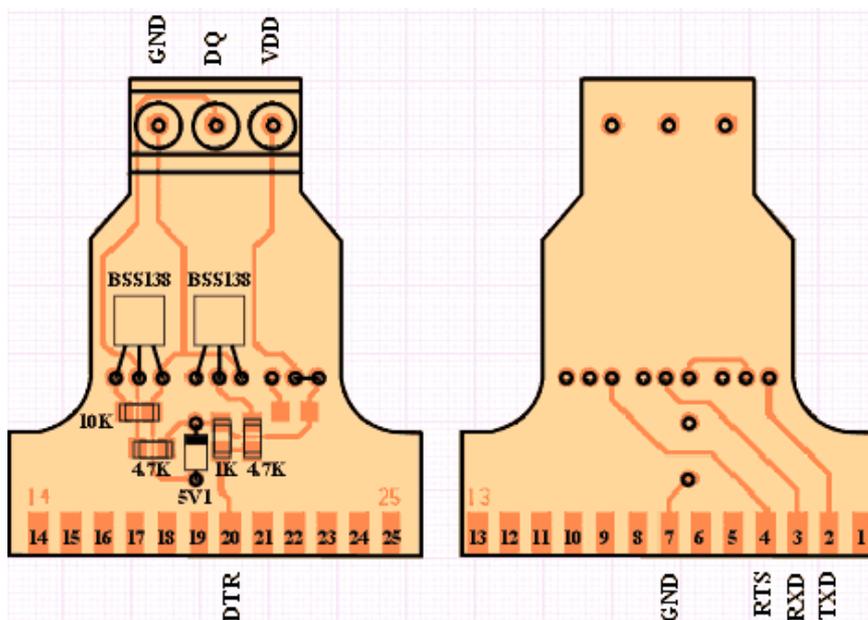


Рис. 11. Рисунки печатной платы адаптера.

Можно вовсе обойтись без печатной платы, а выполнить монтаж адаптера навесным способом на контактах разъема. К винтовому терминалу подключается микросхема термометра с помощью проводов, длина которых может составлять до нескольких метров. Если датчик термометра используется в комнатных условиях, то никаких мер по его защите применять не нужно, необходимо только заизолировать выводы. Если же предполагается измерять температуру наружного воздуха или каких-то агрессивных сред, датчик термометра необходимо упаковать. Например, можно взять алюминиевый корпус электролитического конденсатора подходящего диаметра и поместить туда датчик, заполнив весь свободный объем теплопроводящей пастой. Сверху такой стакан необходимо загерметизировать.

Описанный адаптер также подходит для считывания электронных ключей IButton и для подключения других однопроводных устройств.

Программа ds1820.exe, работающая под Win95/98/ME/NT, позволяет считывать и отображать показания термометра, а также считывать серийный номер и программировать два пользовательских байта. Эта программа кроме термометра DS18S20 поддерживает и его предшественника DS1820.

Вид главного окна программы показан на рис. 12:

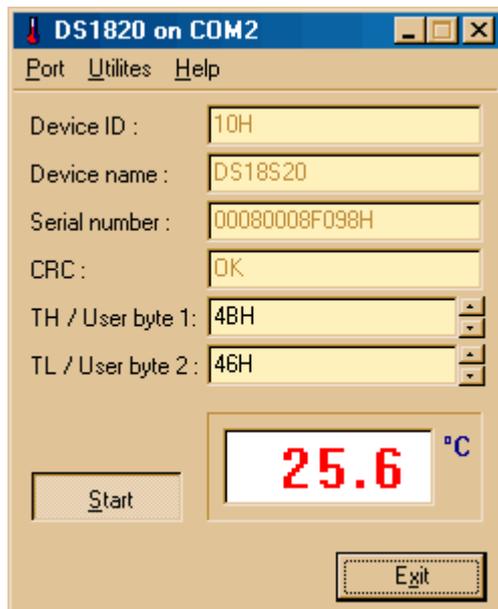


Рис. 12. Вид главного окна программы.

Окно имеет следующие элементы управления:

- Поле Device ID, куда выводится код семейства 1-проводного устройства. Для DS1820 и DS18S20 он равен 10h.
- Поле Device Name, где приводится расшифровка типа устройства.
- Поле Serial number, куда выводится серийный номер, записанный в ПЗУ.
- Поле CRC, где отображается результат проверки контрольной суммы (OK или FAIL).
- Поле TH / User byte 1, куда можно ввести значение в шестнадцатеричной форме, которое будет записано в регистр TH.
- Поле TL / User byte 2, куда можно ввести значение в шестнадцатеричной форме, которое будет записано в регистр TL.
- Кнопка Start запускает процесс измерения температуры. Преобразования выполняются периодически, а измеренная величина выводится в поле температуры. Дискретность представления составляет 0.1°C, что достигается дополнительными вычислениями.
- Кнопка Exit позволяет выйти из программы.

Кроме того, вверху окна имеется меню, состоящее из трех пунктов: Port, Utilites и Help.

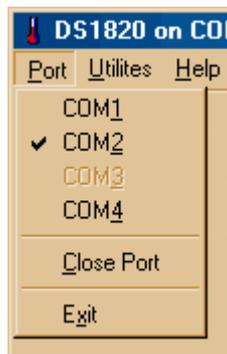


Рис. 13. Меню Port в развёрнутом виде.

Меню Port в развёрнутом виде показано на рис. 13. Это меню позволяет открыть один из четырех портов COM1 – COM4. Кроме того, меню позволяет закрыть порт и выйти из программы. В списке активны только доступные порты (т.е. те, которые физически присутствуют и не заняты в данный момент другими приложениями). Когда порт открывается, проверяется наличие на этом порту адаптера (достаточно соединения TXD – RXD). Если адаптер не обнаружен, выводится соответствующее сообщение (рис. 14):



Рис. 14. Сообщение об ошибке при отсутствии адаптера.

Меню Help содержит рисунок принципиальной схемы адаптера и сведения о разработчике программы.

Для хранения установок программа использует ini-файл, который создается автоматически в том же директории, где расположен exe-файл. В ini-файле содержится информация о положении окна программы и номер используемого COM-порта:

[General]
Left=427
Top=295
COM port=2

Если в существующем ini-файле указан номер COM-порта, который на момент запуска программы занят или отсутствует, выводится специальное сообщение (рис. 15):



Рис. 15. Сообщение об ошибке при недоступности порта.

Меню Utilites (рис. 16) содержит три пункта:

- Read ROM – считывание содержимого ПЗУ (код семейства, серийный номер), а также считывание TH и TL.
- Start Conv. – дублирует кнопку Start основного окна.
- Write User Bytes – записывает значения TH и TL, которые введены в соответствующих полях.

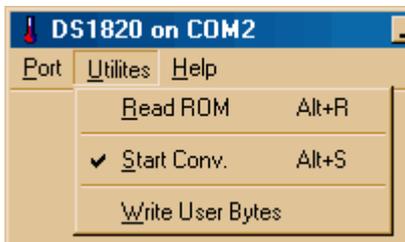


Рис. 16. Меню Utilites в развёрнутом виде.

Обмен, который производится по 1-проводной шине, требует довольно точного соблюдения временных соотношений микросекундного диапазона. Работа под Windows, точно сформировать такие интервалы программным способом невозможно. Поэтому необходимые временные интервалы формируются аппаратно микросхемой приемопередатчика последовательного порта, который используется не совсем обычно. Для генерации импульса сброса и приема ответа порт настраивается на скорость 9600 бод, длина символа 8 бит, и передается число F0h. Это приводит к формированию

импульса сброса низкого уровня (с учетом инверсии адаптера) длительностью примерно 520 мкс (стартовый бит + 4 передаваемых бита). За ним следует импульс высокого уровня такой же длительности (4 передаваемых бита + стоповый бит), в течение которого ожидается ответ термометра. Если термометр не подключен, то порт примет число F0h не искаженным. Но если термометр сформировал импульс ответа, то принятое число будет содержать большее количество единиц, чем четыре. Таким образом определяется наличие ответа.

Для генерации интервалов приема и передачи битов, порт настраивается на скорость 115200 бод. Стартовый бит начинает временной интервал, затем идут 8 единичных или нулевых бит, в зависимости от значения бита, который нужно передать. Заканчивается последовательность стоповым битом, который на некоторое время устанавливает на 1-проводной шине высокий уровень. Вид этой последовательности полностью соответствует требуемому виду временного интервала передачи бита, за исключением того, что обмен является более медленным, чем способна обеспечить 1-проводная шина: на один бит затрачивается примерно 87 мкс.

Прием со стороны передатчика последовательного порта ничем не отличается от передачи единичного бита. Проанализировав принятый последовательным портом символ можно установить, какой бит передал термометр. В случае нулевого бита один или несколько битов принятого портом байта будут единичными (с учетом инверсии адаптера).

Для работы с COM-портом программа пользуется функциями API через специальную «оберточную» динамическую библиотеку comapi32.dll. Однако скорость обмена получается гораздо ниже расчетной из-за того, что переключение скорости COM-порта (вызов функции SetCommState с изменённым значением поля BaudRate структуры DCB) идёт удивительно долго (порядка 200 миллисекунд!). Это весьма неприятное свойство API.

Программа [DS1820.zip](#) вместе с исходным текстом на Delphi 5 [source20.zip](#) распространяется бесплатно.

При использовании термометра необходимо учитывать, что корпус DS18S20 сделан из материала с относительно плохой теплопроводностью. Поэтому утечка тепла через выводы и подключенные к ним провода может быть весьма ощутимой. Для уменьшения ошибки измерения следует использовать провода малого сечения.

Кроме измерения температуры компонентов внутри корпуса компьютера, этот термометр можно использовать и для других целей, например, как комнатный или как наружный термометр.