

Устройства, подключаемые к микроконтроллерам

Материалы по дисциплине «Микропроцессорные системы»
Специальность 230113 «Компьютерные системы и комплексы»

Составитель: Торгашин Р.Г

ГОБУ СПО ВО Борисоглебский индустриальный техникум

Устройства, подключаемые к МК

В этом тексте перечислены далеко не все устройства и варианты их подключения к MCU. Скорее, это обзор, цель которого – дать представление о наиболее популярных устройствах и методах. Многие главы взяты из открытых источников в сети Интернет, ссылки на которые указаны в сносках.

Переключатели

Для организации простого ввода сигналов в микроконтроллер в системах часто используются тумблеры типа SPST или кнопочные переключатели.

Типы переключателей (основные) в англоязычной системе обозначаются английской аббревиатурой: SPDT, DPDT, SPST и DPST, обозначающей количество полюсов (контактов, которые переключаются) и количество направлений (контактов, к которым подключаются или от которых отключаются).

SPST - Single Pole, Single Throw. Один полюс, одно направление. Один контакт может быть подключен к одному другому контакту или отключен от одного другого контакта. Это тоже одна группа контактов, но не полная.

Переключатель типа SPST - это самый дешевый из имеющихся в настоящее время переключателей.

SPDT - Single Pole, Double Throw. Один полюс, два направления. Это означает, что есть один общий контакт (полюс), который может быть подключен к одному из двух других контактов. В русской схемотехнике это называется одна группа контактов. Подобный тип переключения имеет реле РЭС-10.

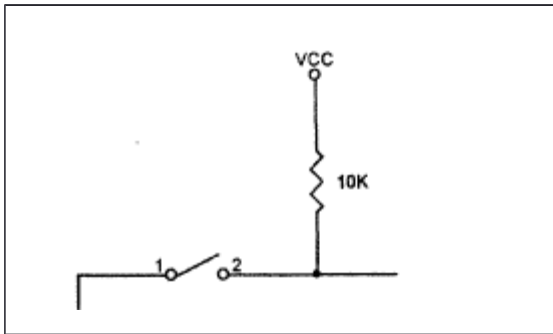
DPDT - Double Pole, Double Throw. Два полюса, два направления. Что означает, что существуют два контакта (полюса), каждый из которых может быть подключен то к одному контакту, то к другому. По русски это называется две группы контактов. Электрически между собой каждая группа никак не соединена. Каждая из них может коммутировать совершенно разные схемы. Общее у них только одно, срабатывают они одновременно. Такой тип переключения имеет реле РЭС-9.

Существуют переключатели, у которых гораздо больше групп контактов, чем две. К примеру реле, которые сами по себе являются электрическими переключателями, с четырьмя группами контактов довольно распространены.

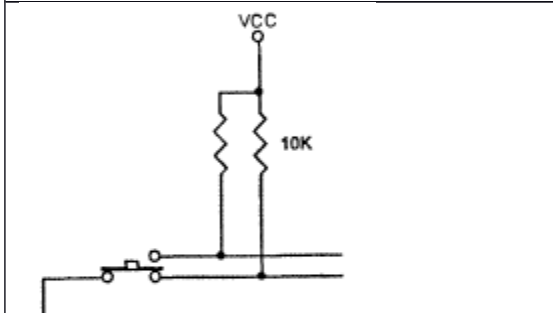
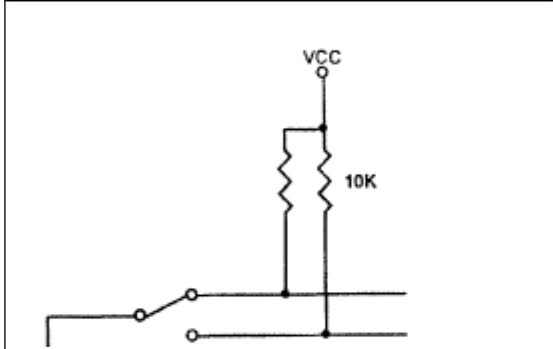
DPST - Double Pole, Single Throw. два полюса, одно направление. Каждый из двух контактов может быть подключен к одному другому контакту или отключен от одного другого контакта. Это считается две группы контактов. Таких групп может быть и больше двух.

Таблица 1 Виды переключателей

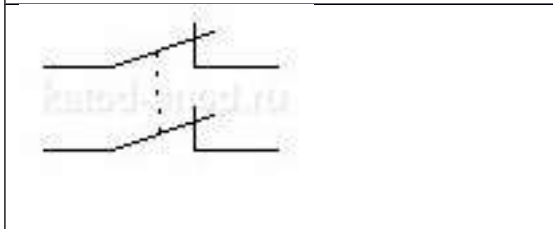
SPST		
		



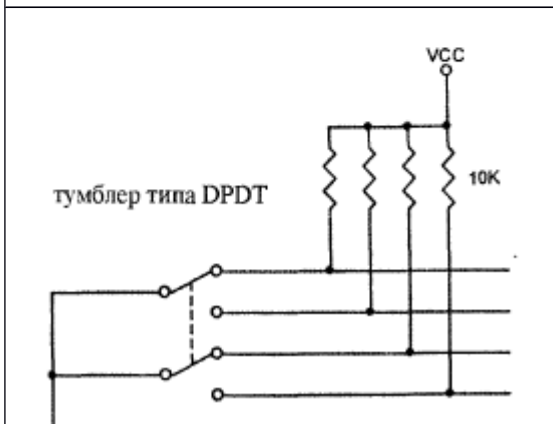
SPDT (SPOT)



DPST



DPDT





В общем случае переключатели могут создавать проблемы при их использовании для ввода некоторых типов данных в силу того, что они подвержены механическому дребезгу контактов. Внутри переключателя находятся металлические контакты, при соединении которых физически возникает явление механического дребезга контактов, которое длится некоторое время. Этот дребезг должен быть компенсирован в системах, в которых количество соединений или синхронизация срабатывания переключателей является важной

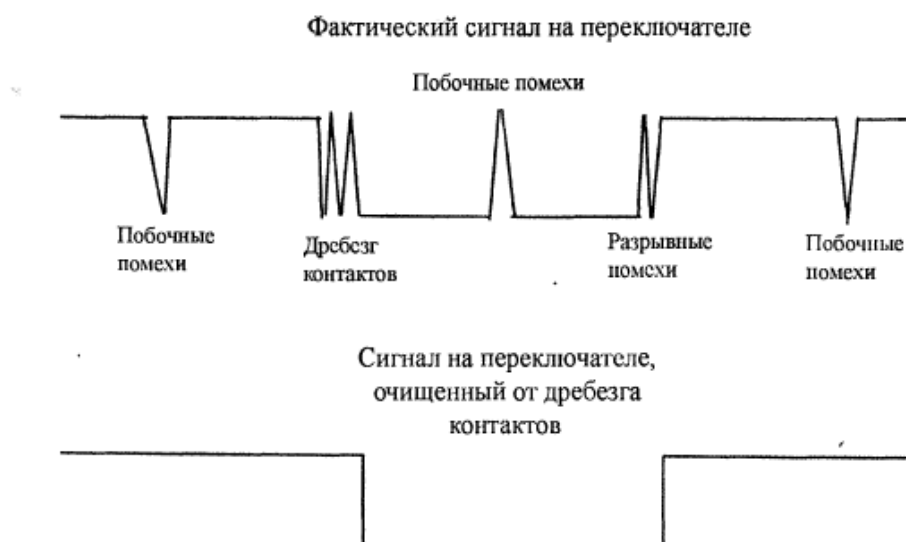


Рисунок 1 Дребезг контактов в SPST

Имеются различные схемы устранения дребезга контактов переключателей. Рис.1 показывает форму электрического сигнала в случае дребезга контактов переключателя. Помехи возникают в сигнальных линиях переключателей в силу того, что обычно является невозможным экранировать переключатель и его соединения с системой от окружающих помех. В большинстве случаев помехи, так же, как и дребезг, должны быть удалены с контактов системы для обеспечения ее надежной работы. Как можно видеть на иллюстрации, переключатели могут оказаться чрезвычайно шумными устройствами. Количество шума зависит от окружающей среды, при этом на рис. 1 представлен экстремальный случай. Дребезг контактов также является переменной величиной. Когда переключатель является новым, дребезг контактов будет минимальным, однако по мере старения переключателя дребезг контактов становится все более серьезным. Помехи, возникающие при разрыве контактов, практически отсутствуют, когда переключатель новый, однако они возникают по мере старения переключателя и становятся все более интенсивными со временем.

Как устранить эти проблемы, чтобы получить чистый сигнал, показанный в нижней части рисунка? Программные временные задержки часто используются, чтобы пропустить проблемные временные области сигналов. Могут также использоваться специальные электронные схемы,

однако они стоят денег, поэтому используются довольно редко. Рис. 2 демонстрирует некоторые электронные схемы, которые применяются для устранения дребезга механических переключателей. Эти схемы весьма эффективны, однако достаточно дороги.

Лучшим и менее дорогостоящим способом устранения проблемы дребезга контактов и шума является использование программного обеспечения.. Когда переключатель находится в устойчивом состоянии логического нуля, отрабатывается выход из функции. Предполагается, что тактовая частота равна 4 МГц, а временная задержка равна - 15 мс. Переключатели имеют максимальное время дребезга контактов, равное 8 мс, поэтому в данном примере используется время задержки, равное 15 мс, гарантирующее пропускание периода дребезга контактов. В общем случае для удаления дребезга контактов приемлемыми являются времена задержки от 10 до 20 мс. Если время задержки будет намного большим 20 мс, работа переключателя замедлится, потому что он должен будет удерживаться в состоянии срабатывания слишком долго.

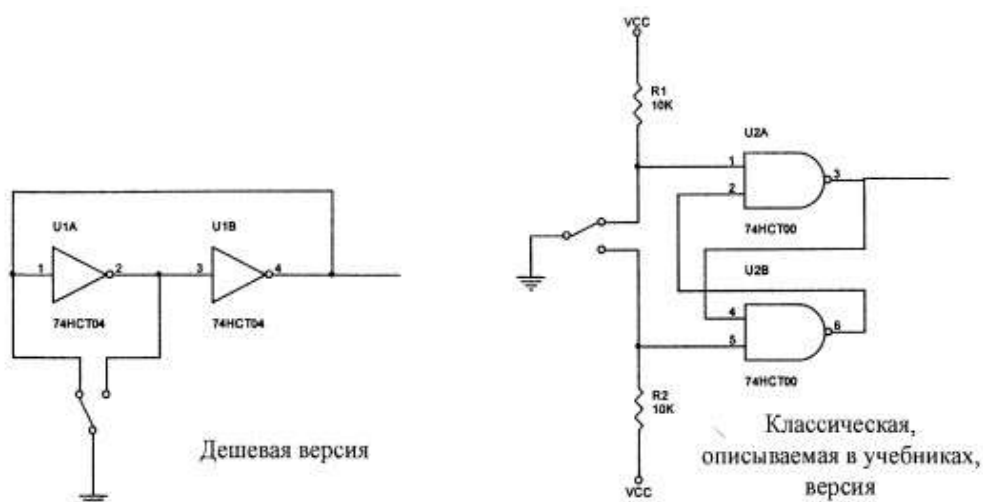


Рисунок 2 Схемы устранения "дребезга" контактов

При программном формировании очищенного от дребезга контакта сигнала наибольшее распространение получили два из них:

1. Путем установки временной задержки — программа, обнаружив замкнутое состояния контакта, игнорирует его состояние на время, заведомо большее длительности дребезга.
2. Методом подсчёта числа совпадающих значений сигнала замкнутости — программа многократно считывает состояние контакта, и, если последовало определённое количество подтверждений замкнутости, (определяется экспериментально и выбирается в пределах от 10 до 100) контакт считается устойчиво замкнутым

Светодиоды

Светодиод – нелинейный полупроводниковый прибор, которому для правильной и надежной работы необходим стабильный ток. Перегрузки по току могут вывести светодиод из строя. **Самый простой вариант схемы питания в таком случае – ограничительный резистор, включенный последовательно.** Расчет номинального сопротивления и мощности резистора для светодиода не очень сложная задача, если правильно понимать физику процесса. Рассмотрим общие принципы такого расчета, а затем разберем несколько конкретных примеров из практики.¹

1 <http://le-diod.ru/podklyuchenie-ustanovka/raschet-soprotivleniya-dlya-svetodioda/>

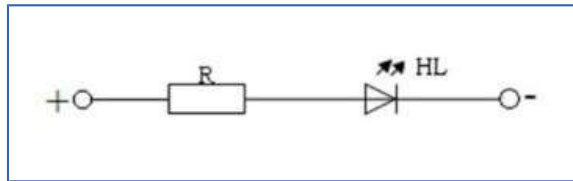


Рисунок 3 Подключение светодиода через токоограничивающий резистор

Между контактами «+» и «-» прикладывается напряжение. Обозначим его буквой U . Ток через резистор и светодиод будет протекать одинаковый, т.к. соединение последовательное. Согласно закону Ома получаем:

$$I = \frac{U}{R + r_{LED}}, \quad (1)$$

где R – сопротивление резистора;

r_{LED} – сопротивление светодиода (дифференциальное).

Отсюда выражаем формулу, по которой можно произвести расчет сопротивления резистора R при заданном токе I :

$$R = \frac{U}{I} - r_{LED}. \quad (2)$$

Разберемся что такое дифференциальное сопротивление светодиода r_{LED} . Для этого нам потребуется его вольтамперная характеристика (ВАХ).

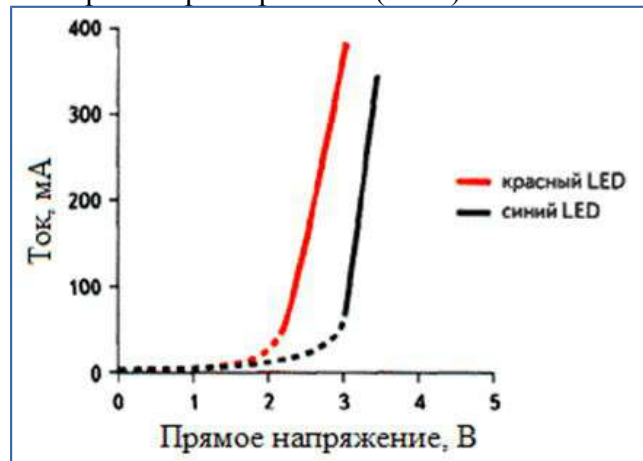


Рисунок 4 ВАХ светодиодов

Как видно из графиков ВАХ светодиодов – нелинейна. Говоря простым языком, его сопротивление постоянному току $r=U/I$ есть переменная величина, которая уменьшается с ростом напряжения. Поэтому вводится понятие дифференциального сопротивления $r_{LED}=dU/dI$, которое характеризует сопротивление диода в отдельно взятой точке кривой ВАХ.

$$R = \frac{U}{I} - \frac{U_{LED}}{I} = \frac{U - U_{LED}}{I}. \quad (3)$$

Практика

Рассмотрим конкретный пример расчета.

Исходные данные: напряжение питания 12В, белый светодиод XPE (CREE) требуется включить на номинальный ток 350 мА согласно схеме, представленной на рисунке 1.

Находим в datasheet значение прямого падения напряжения при токе 350 мА (рисунок 4).

Таблица 2 Параметры светодиода

Characteristics	Unit	Minimum	Typical	Maximum
DC Forward Current - white, royal blue, blue, green	mA			1000
DC Forward Current - red-orange, red	mA			700
DC Forward Current - amber	mA			500
Reverse Voltage	V			5
Forward Voltage (@ 350 mA) - royal blue, blue, white	V		3.2	3.9
Forward voltage (@ 350 mA) - green	V		3.4	3.9
Forward Voltage (@ 350 mA) - red-orange, red, amber	V		2.1	2.8

Типовое значение по таблице — 3,2 вольта. Максимальное значение может достигать 3,9 вольт. То есть в результате производственного процесса может получиться как светодиод с прямым напряжением 3,2 В так и 3,9 В (или любым другим промежуточным значением), но вероятность получения 3,2 вольт наиболее высока (если хотите – это «математическое ожидание» этой величины). По этой причине в расчет обычно берется типовое значение.

Используя формулу (3) и калькулятор получаем:

$$R=(12-3,2)/0,35 \approx 25,1 \text{ Ом.}$$

Ближайшее значение из ряда E24 – 24 Ом. Значение тока при этом сопротивлении получится 367 мА, что на 5% превышает требуемое значение. Если учесть еще и допуск на номинал резистора, который для ряда E24 также 5%, то в худшем случае получается вообще 386 мА. Если такое отклонение не допустимо, то можно добавить в цепь последовательно еще один резистор номиналом 1 Ом. Все эти действия рекомендуется сопровождать реальными измерениями сопротивлений резисторов и получающихся токов, иначе ни о какой точности не может идти и речи. Резистор 24 Ом может иметь погрешность в сторону увеличения до 25,2 Ом, добавив 1 Ом, получим 26, 2 и «перекос» силы тока через светодиод в противоположную сторону.

Предположим, что нам не нужна высокая точность задания тока и резистор 24 Ом нас устраивает.

Определим мощность, которая будет рассеиваться на резисторе по формуле (4):

$$P=0,367^2 \times 24 \approx 3,2 \text{ Вт.}$$

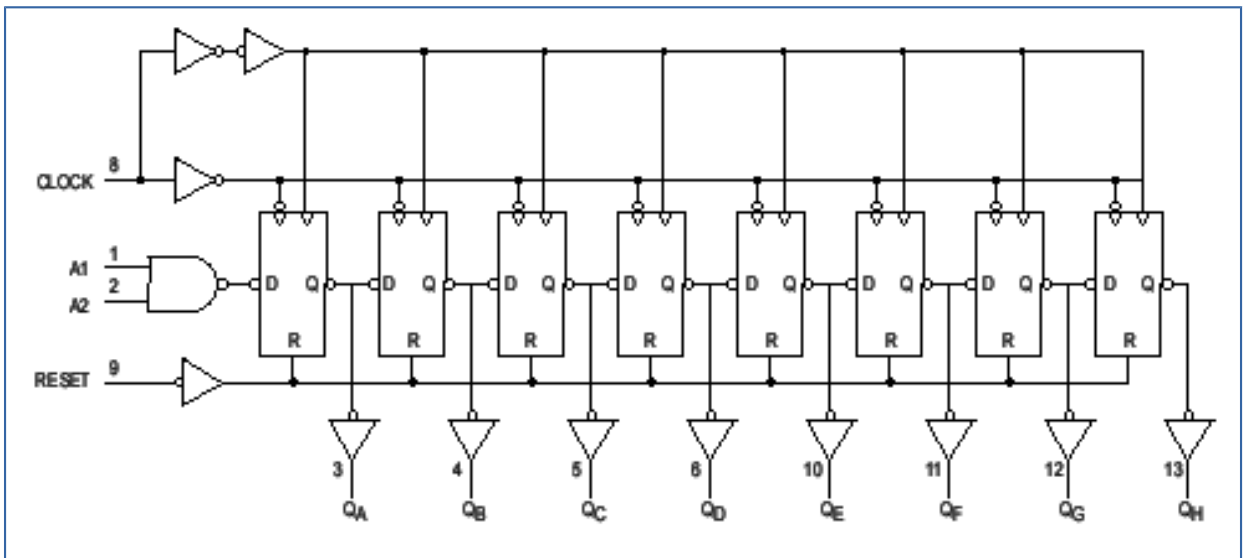
Номинальная мощность рассеяния резистора должна быть с запасом не менее 30%, иначе он будет перегреваться. А если условия отвода тепла затруднены (например, в корпусе плохая конвекция), то запас должен быть еще больше.

В итоге выбираем резистор мощностью 5 Вт с номинальным сопротивлением 24 Ом.

Сдвиговый регистр²

Иногда требуется **ОЧЕНЬ** много выходных портов. Особенно если хотим сделать что-нибудь на светодиодах. Гирлянду например. Тут на помощь из вековых глубин выплывает старая добрая дискретная логика. На этот раз нас выручит сдвиговый регистр. Возьму, для примера, **74НС164** он же, для любителей совковых трешевых микросхем в неубиваемом каменном корпусе, наш **КМ555ИР8**.

² <http://easyelectronics.ru/sdvigovyj-registr.html>



У него есть 8 выходов и четыре входа. R-сброс, C-тактовый, A1 и A2 вход. На самом деле, внутри они заведены через логический элемент 2И-НЕ и идут на D триггеры. D — это такой тип триггера, который по тактовому импульсу схватывает и отправляет на выход то, что у него на входе. Тут они цепью стоят, передавая бит от одного к другому и нет принципиальной разницы сколько их тут будет, восемь штук или восемь миллиардов. Но чем больше, тем дольше по этой эстафете гнать данные до конца. Поэтому мы смело можем эти регистры соединять последовательно.

От МК, как видно, требуется только четыре выхода. Одним (RESET) мы сбрасываем состояние регистра. Из второго (Data) побитно выводится байт, а тактовый CLC обеспечивает продвижение битов по регистру. Самых регистров тут три. Они сцеплены последовательно. Когда переполняется первый, то биты из него переходят во второй, потом в третий. Итого, 24 вывода.

Наполнять регистр просто:

- 1) Поднимаем и держим **RESET** в 1
- 2) Выдаем первый (старший) бит на **Data**.
- 3) Опускаем в 0 и поднимаем в 1 тактовый выход. На восходящем фронте происходит занос в регистр и сдвиг всей цепочки на один шаг.
- 4) Повторить со второго пункта пока все биты не выдадим.

А для сброса достаточно уронить **Reset** в ноль на пару микросекунд.

Все просто :)

P.S

Кружок на входе регистра означает, что вход инверсный. Т.е. подал ноль — сработало. Треугольник на входе показывает по какому фронту произойдет срабатывание. Запомнить просто: $_/_$ — это, типа, импульс. А треугольник, как стрелочка, указывает на нужный фронт. $\rightarrow_/_$ передний (восходящий фронт) и $_/_<$ задний (нисходящий фронт)

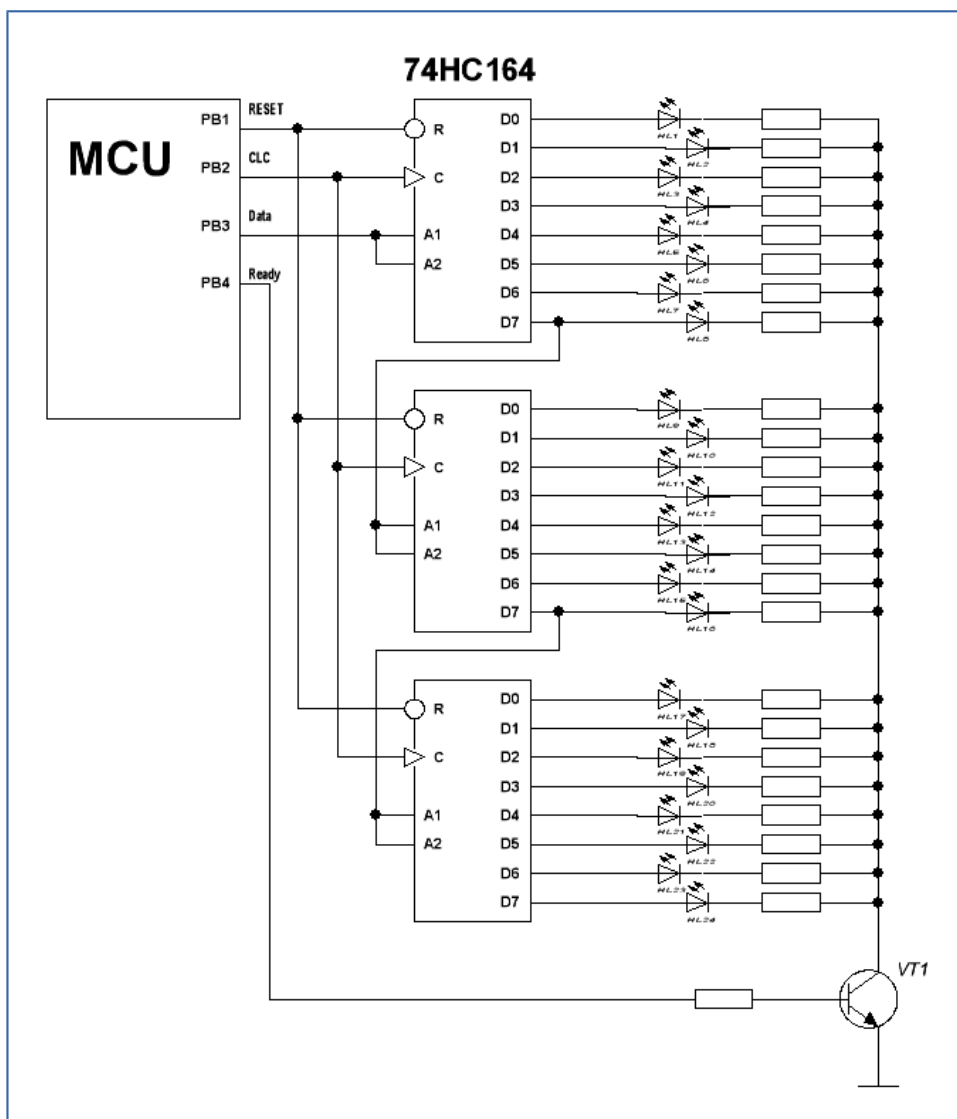


Рисунок 5 Использование сдвигового регистра для управления светодиодами (элементами семисегментного индикатора)

Семисегментники³

Для отображения цифровой информации в системах на базе микроконтроллеров используются светодиодные семисегментные индикаторы. Они просты в управлении, имеют высокую яркость, широкий диапазон рабочих температур и низкую стоимость. К недостаткам светодиодных индикаторов относятся – высокое энергопотребление, отсутствие управляющего контроллера и скудные возможности по выводу буквенной информации.

Светодиодный семисегментный индикатор представляет собой группу светодиодов расположенных в определенном порядке и объединенных конструктивно. Зажигая одновременно несколько светодиодов можно формировать на индикаторе символы цифр. Индикаторы различаются по типу соединения светодиодов – общий анод, общий катод, по количеству отображаемых разрядов – одноразрядные, двух разрядные и т.д. и по цвету – красные, зеленые, желтые и т.д.

³ <http://chipenable.ru/index.php/programming-avr/item/58-uchebnyy-kurs-semisegmentnyy-indikator-dinamicheskaya-indikatsiya.html>

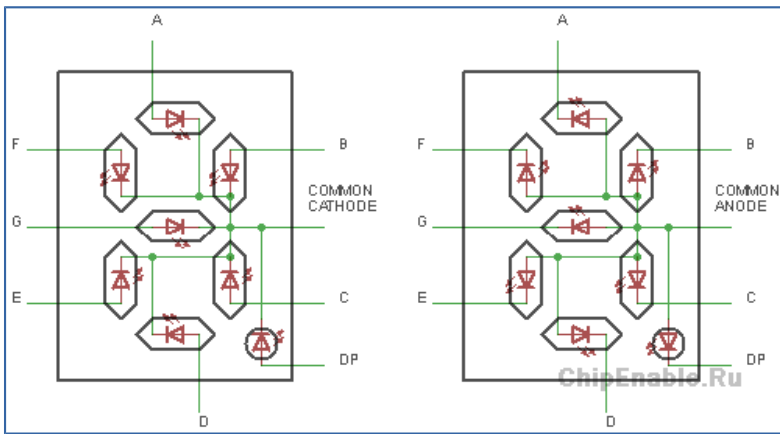


Рисунок 6 Структура семисегментников с общим катодом и с общим анодом



Рисунок 7 Блок семисегментников

Семи сегментным индикатором можно управлять статически или динамически. При **статическом управлении** разряды индикатора подключены к микроконтроллеру независимо друг от друга и информация на них выводится постоянно. Этот способ управления проще динамического, но без использования дополнительных элементов, как-то сдвиговые регистры, подключить многоразрядный семисегментный индикатор к микроконтроллеру будет проблематично - может не хватить выводов.

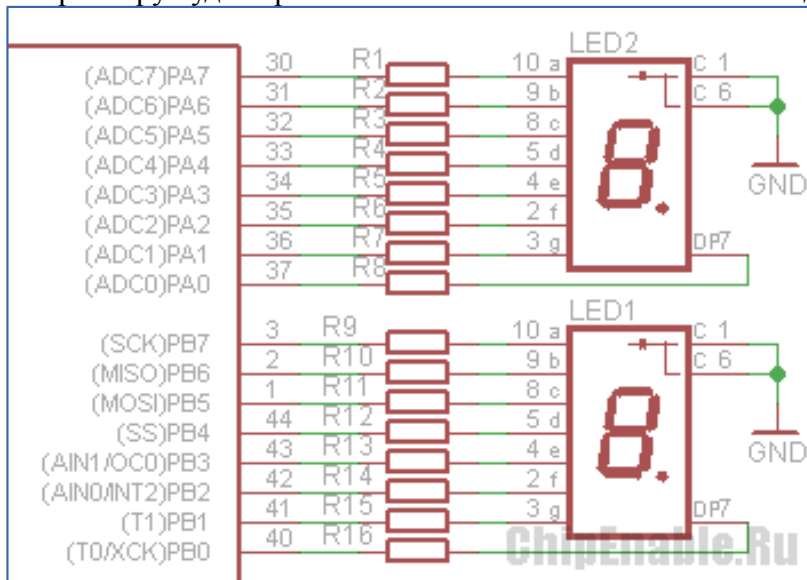


Рисунок 8 Подключение семисегментников "напрямую" для статического управления

Подключение семисегментника через сдвиговый регистр⁴ является более удобным. В этом случае двоичный код, описывающий работу светодиодов семисегментника

4 <http://cxem.net/mc/mc235.php>

заносится в сдвиговый регистр. Пропадает необходимость постоянно держать на выводах MCU код символа. Кроме того, для подключения любого числа индикаторов достаточно трех выводов MCU. А при некотором навыке можно обойтись одним!

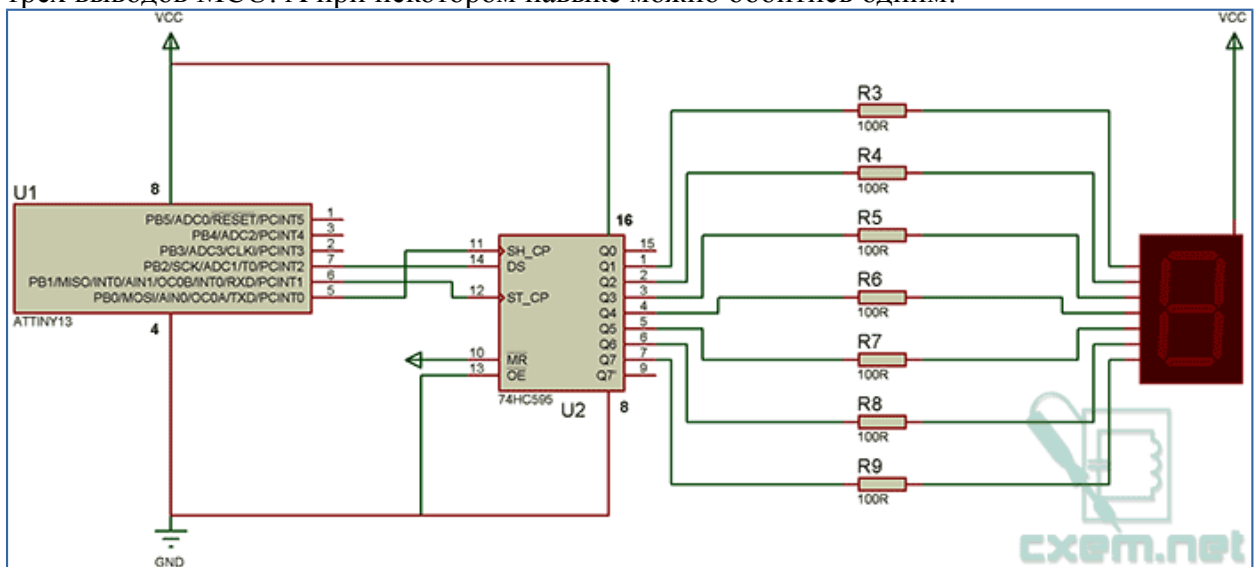


Рисунок 9 Простой способ использования регистра

В качестве примера рассмотрим работу статического светодиодного цифрового индикатора, управляемого по SPI

Каждым семисегментным индикатором управляет отдельная микросхема типа *SN74HC595D*, содержащая:

- 8-разрядный сдвиговый регистр;
- параллельный регистр-защелку;
- управляемый выходной буфер.

Динамическое управление (динамическая индикация) подразумевает поочередное зажигание разрядов индикатора с частотой, не воспринимаемой человеческим глазом. Схема подключения индикатора в этом случае на порядок экономичнее благодаря тому, что одинаковые сегменты разрядов индикатора объединены.

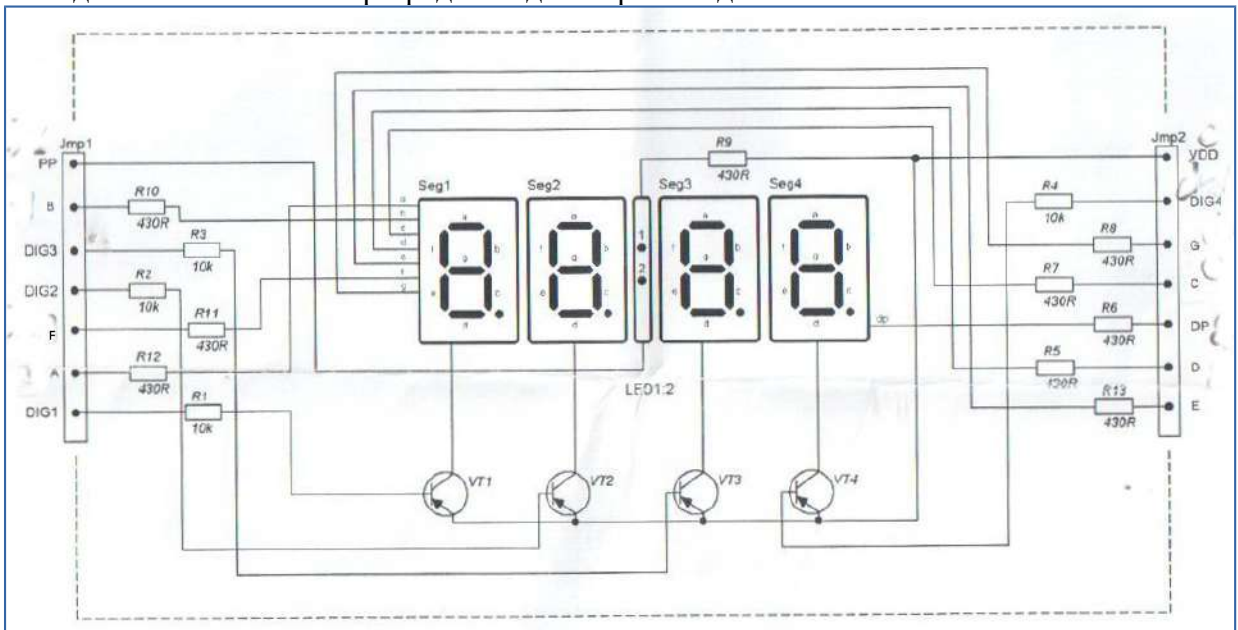


Рисунок 12 Схема блока 4-х индикаторов с динамическим управлением

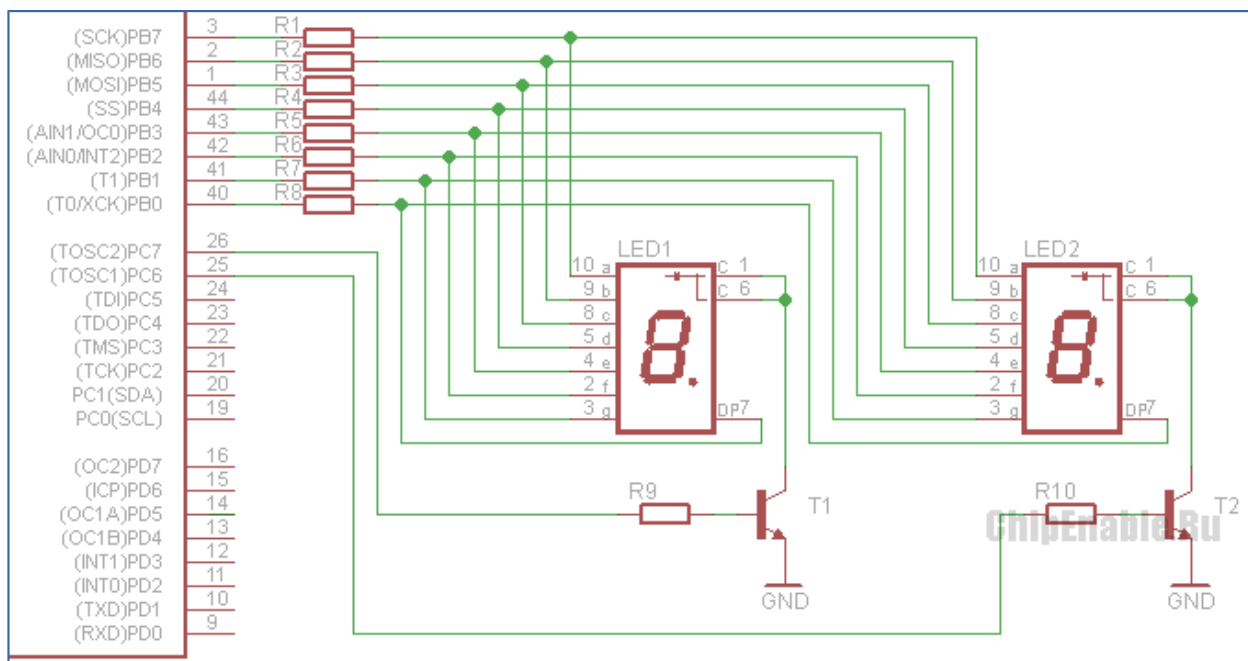


Рисунок 13 Подключение 2-х индикаторов в режиме динамического управления к MCU

Продвинутый уровень: [Управление семисегментными индикаторами по одному проводу](#)

Аналоговый термодатчик

Аналоговый термодатчик это устройство, напряжение на выходе которого зависит от температуры датчика. Эти устройства могут использоваться для измерения температуры окружающей среды и т.п.

Такой датчик можно подключить к входу АЦП

Рассмотрим, например, датчик TC1047A
Напряжение питания 5В
Измеряемая температура -40...+125 °С

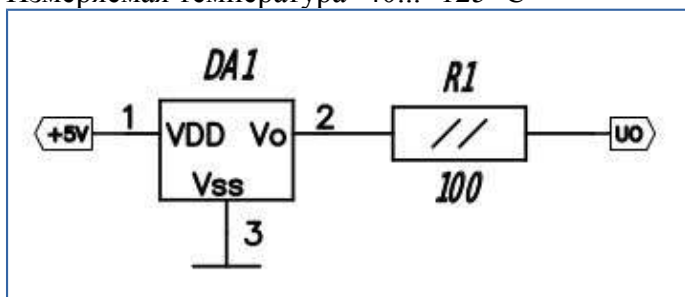


Рисунок 14 Подключение и внешний вид датчика TC1047A

Точность преобразования температуры 2 °С

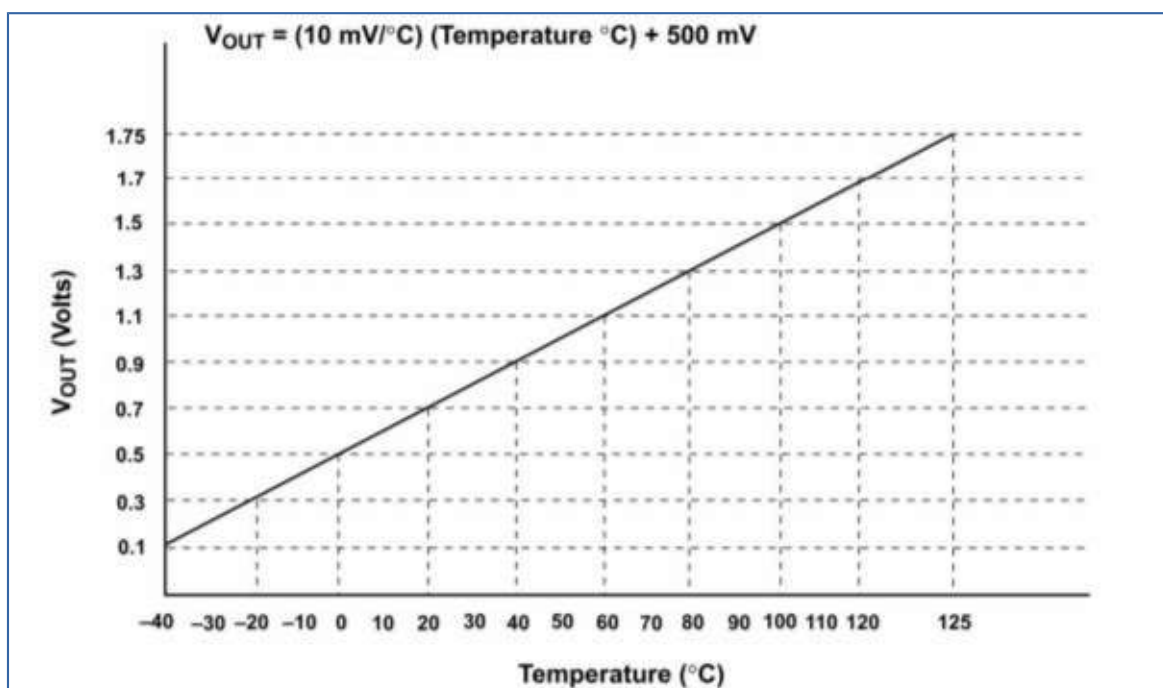


Рисунок 15 Зависимость напряжения от температуры

Цифровой термодатчик

Как известно, параметры полупроводникового р-п перехода в биполярных транзисторах и диодах имеют температурную зависимость. Если прямосмещенный переход соединить с генератором постоянного тока, то его выходное напряжение будет находиться в прямо пропорциональной зависимости от температуры. Калибровка таких датчиков проводится только по двум точкам.

Полупроводниковые датчики имеют встроенную схему преобразования сигнала в цифровую форму. Благодаря тому, что чувствительный элемент и схема преобразования сигнала находятся на одном кристалле, погрешность измеряемой температуры подобных датчиков достаточно мала и составляет от 0,5 до 2°С. Кроме того, при считывании дополнительных регистров и их арифметической обработке, можно достичь высокого разрешения с ценой младшего разряда 0.0625°С.

Многие датчики с цифровым выходным сигналом могут быть использованы в качестве термостатов с пользовательской регулировкой контрольных точек с гистерезисом. Встроенная энергонезависимая EEPROM память сохраняет максимальные и минимальные пределы, заданные пользователем, превышение которых приводит к включению логических выходов микросхемы. Третий логический выход будет оставаться открытым до

снижения температуры до низкого уровня (программируемый гистерезис). Термодатчики выдают информацию о параметрах температуры в виде 9-, 10-, 11-, или 12-битного кода.

Некоторые микросхемы датчиков температуры имеют несколько адресных выводов, благодаря чему к шине может быть подключено до 16 различных устройств. Обмен данными (чтение/запись) может производиться по различным интерфейсам: однопроводному 1-Wire, двухпроводному 2-Wire, SPI, трехпроводному и др.

Основные области применения:

- управление вентиляторами вычислительных систем
- подстройка тактовой частоты для компенсации влияния повышения температуры критичных к разогреву процессоров
- аналитическое и научное оборудование
- автоматическое регулирование температуры окружающей среды
- управление кондиционерами зданий и сооружений
- компенсация влияния температуры на стабильность времязадающих элементов, критичных к стабильности частоты, в частности измерительной аппаратуры, аппаратуры радиосвязи.

Рассмотрим, для примера датчик DS1621

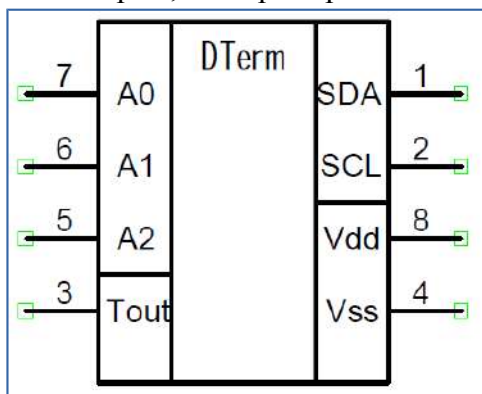


Рисунок 16 Датчик DS1621

Микросхема термодатчика DS1621 может измерять температуру в диапазоне от -5 до +125°C. Измерение температуры производится с шагом в 0,5°C. Управление и считывание температуры производится по шине I²C. Микросхема имеет адрес 1001 A2 A1 A0, где A_x - биты которые задаются внешним сигналом на выводах A2 A1 A0. Если на вывод A2 подать напряжение высокого логического уровня, на A1 низкое, A0 высокое, то микросхема будет иметь адрес 10011010. Это позволяет задавать ей до восьми различных адресов. Данный способ адресации используется, когда на шину I²C устанавливается более 1 такого устройства. По умолчанию адрес микросхемы 10010000.

Таблица 3 Назначение выводов датчика DS1621

Вывод	Название	Направление	Описание
1	SDA	IN/OUT	Шина данных I ² C
2	SCL	OUT	Тактовая шина I ² C
3	Tout	OUT	Выход термостата
4	Vss	-	Общий
5	A2	IN	Вывод задающий 2 бит 7-разрядного адреса
6	A1	IN	Вывод задающий 1 бит 7-разрядного адреса
7	A0	IN	Вывод задающий 0 бит 7-разрядного адреса
8	Vdd	-	Питание

Значение температуры представляется в виде 9-разрядного шестнадцатеричного числа эквивалентного значению температуры в градусах Цельсия. При считывании датчика передается два байта данных.

Первый байт содержит целое значение измеренной температуры.

Второй байт содержит дробную часть: 1 в старшем бите означает, что к первому байту нужно добавить +0,5 градусов Цельсия

Таблица 4 Выходные значения датчика при определенной температуре

Температура	Выходное значение (Binary)	Выходное значение (Hex)
+125°C	01111011 00000000	7B00h
+25°C	00011001 00000000	1900h
+0,5°C	00000000 10000000	0080h
+0°C	00000000 00000000	0000h
-0,5°C	11111111 10000000	FF80h
-25°C	11100111 00000000	E700h
-55°C	11001001 00000000	C900h

Отрицательные значения температур представляются в дополнительном коде, только целое значение (первый байт), дробная часть всегда отображается в прямом коде. Если температура меньше 0 (MSB = 1). При преобразовании из шестнадцатеричного числа в двоично-десятичное

Таблица 5 Команды управления термодатчиком

Команды для измерения температуры	
Команда	Описание
0xAA	Считывание температуры измеренной в предыдущем цикле, считываются два байта данных.
0xA8	
0xA9	
0xEE	Инициализация начала измерения температуры
0x22	Остановка измерения температуры
Команды управления термостатом	
Команда	Описание
0xA1	Открывает доступ для чтения/записи регистра TH
0xA2	Открывает доступ для чтения/записи регистра TL
0xAC	Открывает доступ для чтения/записи регистра конфигурации

Один цикл преобразования длится примерно 1 секунду. Значение регистров содержащих вычисленное значение температуры доступны для считывания в любой момент времени, обновляются в конце каждого цикла преобразования температуры. Для работы с термодатчиком необходимо подать команду 0xEE.

RTC – Часы реального времени

Часы реального времени (RTC) — электронная схема, предназначенная для учёта хронометрических данных (текущее время, дата, день недели и др.), представляет собой систему из автономного источника питания и учитывающего устройства.

Несмотря на то, что часы реального времени могут быть реализованы без использования специализированных устройств, применение специализированной схемы для RTC позволяет добиться более низкого энергопотребления, освобождения центрального процессора для критичных по времени задач, обеспечить более высокую точность.

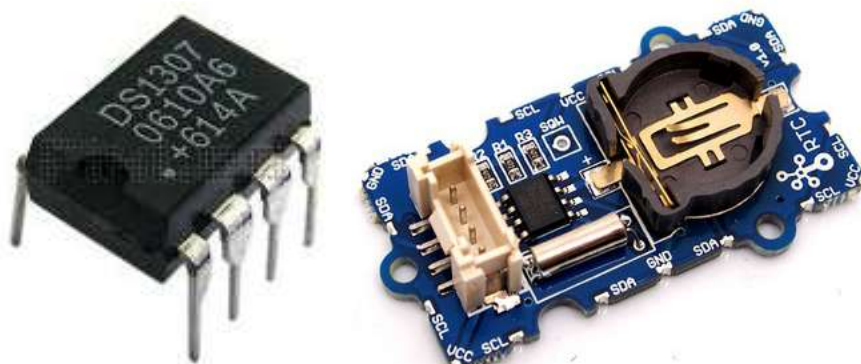


Рисунок 17 Микросхема RTC и плата с RTC и источником питания

RTC часто используют независимые от основной системы источники питания, так, чтобы продолжать работать даже при выключении основной системы. В старых системах для этого использовалась подключаемая к чипу кассета с пальчиковыми элементами на 1 год, позже — встроенная в чип литиевая батарейка на 10 лет, в новых может

использоваться ионистор. В компьютерах тот же самый источник питания может использоваться и для обеспечения работы CMOS-памяти (для хранения настроек BIOS)

Большинство RTC использует кварцевый резонатор, но некоторые используют частоту питающей сети. В большинстве случаев используется кварцевый резонатор на частоте 32768 Гц. Та же частота используется в кварцевых часах. Такая частота обеспечивает 215 циклов в секунду, что очень удобно для простых двоичных счётчиков

Рассмотрим, для примера RTC DS1307⁵

Особенности:

- Очень маленькое энергопотребление. Производитель обещает 10 лет работы часов от одной стандартной батарейки **CR2032**
- 56 байт памяти для хранения пользовательских данных.
- Программируемый вывод для тактирования внешних устройств. Может выдавать 1 Гц, 4.096 кГц, 8.192 кГц и 32.768 кГц.
- 24-х часовой и 12-ти часовой режим



Рисунок 18 RTC DS1307

X1, X2 – Выводы для подключения кварцевого резонатора на частоту 32.768 кГц

VBAT – Вывод для подключения 3-х вольтовой батареи резервного питания

GND – Земля

SDA – линия данных шины i2c

SCL – линия тактовых импульсов шины i2c

SQW/OUT – выходной сигнал для тактирования внешних устройств

VCC – питание 5 вольт

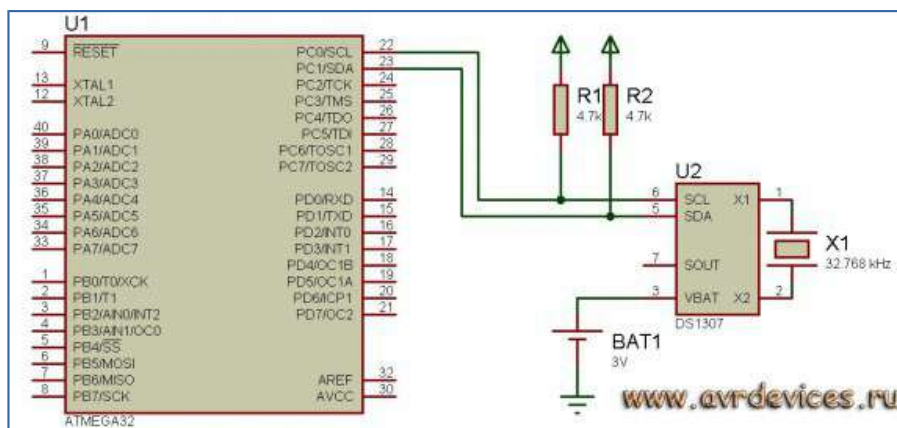


Рисунок 19 Подключение RTC к MCU

Организация памяти часов

5 <http://avrdevices.ru/chasi-realynogo-vremeni-ds1307/>

Данная микросхема наделена 64 байтами памяти. Первые восемь байт – рабочие. В них хранится время, дата, день недели. Остальные выделены под нужды пользователя. В них можно хранить, например, какие-нибудь настройки или еще что-нибудь. Естественно, когда резервное питание пропадает, вся информация в этой памяти разрушается. Вся работа с часами (чтение и установка времени/даты) сводится к тому, чтобы читать и записывать нужные ячейки памяти.

Таблица 6 Значение битов памяти микросхемы RTC DS1307

АДРЕСА	БИТЫ							
	7	6	5	4	3	2	1	0
0x00	CLOCK HALT	ДЕСЯТКИ СЕКУНД			СЕКУНДЫ			
0x01	—	ДЕСЯТКИ МИНУТ			МИНУТЫ			
0x02	—	24/12	АМ/РМ ДЕСЯТКИ ЧАСОВ	ДЕСЯТКИ ЧАСОВ	ЧАСЫ			
0x03	—	—	—	—	—	ДЕНЬ НЕДЕЛИ		
0x04	—	—	ДЕСЯТКИ ДАТЫ		ЕДИНИЦЫ ДАТЫ			
0x05	—	—	—	Десяток месяца	ЕДИНИЦЫ МЕСЯЦА			
0x06	—	ДЕСЯТКИ ЛЕТ			ГОД			
0x07	Output	—	—	SQWE	—	—	RS1	RS0

Все числа в памяти хранятся в двоично-десятичном формате. Это значит что в одном байте может храниться сразу две цифры. Например число 0x23 – содержит в себе цифру 2 и цифру 3. На каждую цифру выделяется по 4 бита. Зачем так сделано? Для удобства и экономии памяти. Кроме времени и даты в памяти хранятся несколько бит настроек:

- **Clock Halt** – управляет часами. Когда бит установлен то часы стоят. Чтобы запустить ход часов необходимо записать в этот бит 0. После подключения батареи резервного питания, этот бит уставлен и часы не считают время! Об этом нужно помнить.
- **24/12** – этот бит выбора режима часов. Когда этот бит равен единице то используется 12-ти часовой режим. В противном случае 24-х часовой. Если используется 12-ти часовой режим то пятый бит показывает АМ или РМ сейчас. Если бит равен 1 то значит РМ. В 24-х часовом режиме этот бит используется для хранения десятков часов совместно с битом 4.
- **Output** – управляет состоянием ноги SQW/OUT. Бит установлен – на ноге лог 1. Сброшен – на ноге 0. Для управления таким образом, бит **SQWE** должен быть сброшен.
- **SQWE** – когда бит установлен, на ноге SQW/OUT появляются прямоугольные импульсы.
- **RS1, RS0** – этими битами задается частота импульсов. Зависимость частоты от комбинации бит находится в таблице ниже:

Таблица 7 Настройка частоты битами RS

RS1	RS0	Частота
0	0	1 Гц
0	1	4.096 кГц
1	0	8.192 кГц
1	1	32.768 кГц

Драйверы (аппаратные устройства)

Как видно из вышесказанного, не сложно подключить и использовать «простые» устройства вроде переключателей или единичных светодиодов. Подключение более «сложных» устройств: шаговых двигателей, LCD матриц, клавиатур, блоков семисегментных индикаторов и т.п. напрямую к контроллеру часто связано с большим расходом выводов и усложняет код управляющей программы.

Например, для подключения двух семисегментных индикаторов к микроконтроллеру потребуется 18 выводов. И при подключении их к микроконтроллеру через сдвиговый регистр можно обойтись 3 выводами, а при определенном навыке – одним! Более того, при наращивании числа семисегментников, число выводов не меняется.

Устройство, которое служит посредником между микропроцессорной системой и периферийным оборудованием называют **драйвером**. Хотя задачи, которые решает аппаратный драйвер, сходны с задачами программных драйверов – путать их не следует! В приведенном выше примере сдвиговый регистр можно считать драйвером блока семисегментников. Все же обычно драйверы имеют более сложную структуру.

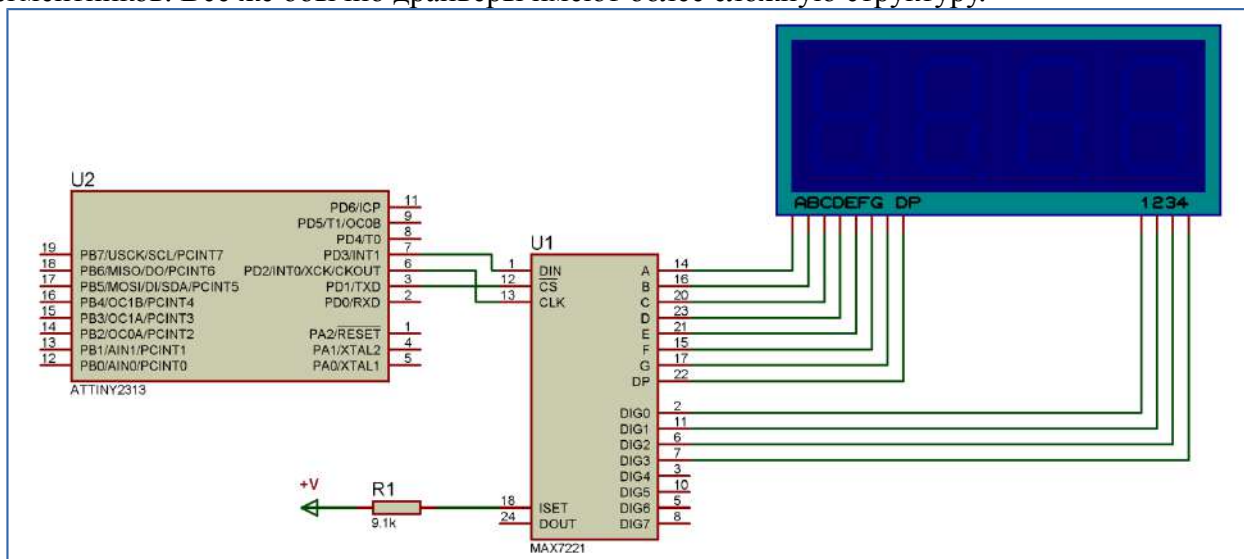


Рисунок 20 Использование драйвера индикаторов MAX7221 для подключения блока из 4 семисегментников⁶

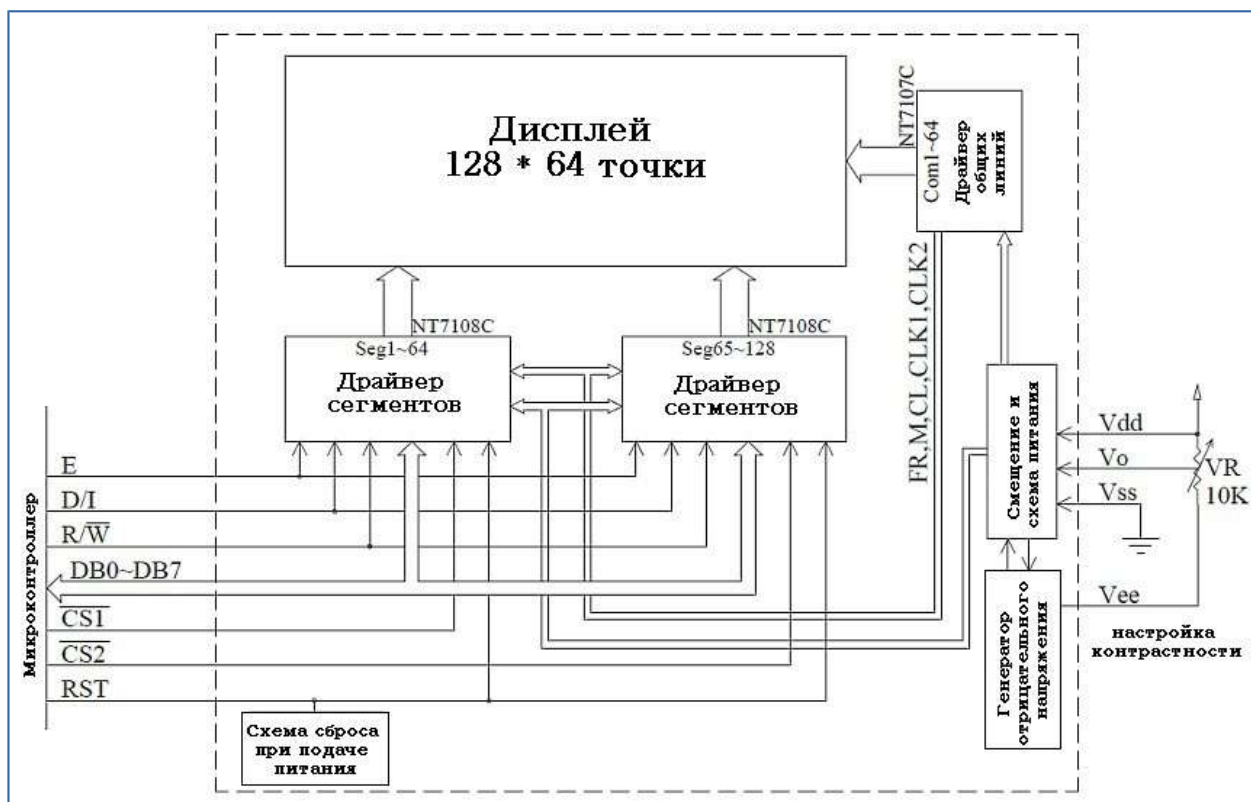


Рисунок 21 Блок-схема модуля графического LCD экрана Winstar WDG0151-TMI.⁷

Подробно параметры использования таких драйверов можно найти в соответствующих DataSheet

Шаговый двигатель⁸

Шаговый двигатель является традиционным исполнительным устройством многих электронных приборов и систем. Шаговый двигатель (далее ШД) представляет собой бесколлекторный двигатель постоянного тока с фиксированными положениями вала. ШД предназначен в первую очередь для точного позиционирования вала без применения систем обратной связи. ШД широко применяются в автомобилях, бытовой радиоаппаратуре, компьютерах. На производстве основное место использования ШД – это автоматизированное технологическое оборудование.

Шаговый двигатель, в общем случае, – это двигатель, имеющий ротор, статор и несколько обмоток. Вал ШД совершает поворот на фиксированный угол (шаг) при кратковременной подаче напряжения на обмотки в правильной последовательности. Обычно обмоток 2 или 4. Для того чтобы заставить ШД вращаться, необходимо поочередно подавать рабочее напряжение на обмотки двигателя в нужной полярности. Для ШД с двумя обмотками (биполярного ШД) необходимо менять очередность и полярность подаваемых импульсов. ШД с 4 обмотками (униполярный ШД) проще в управлении – для вращения вала достаточно поочередно подавать напряжение на каждую из четырёх обмоток. Вал двигателя при этом будет каждый раз поворачиваться на фиксированный шаг. Величина этого шага обусловлена конструкцией двигателя и

⁷<http://www.rlocman.ru/shem/schematics.html?di=113044>

⁸ http://electroprivod.ru/motor_step.htm

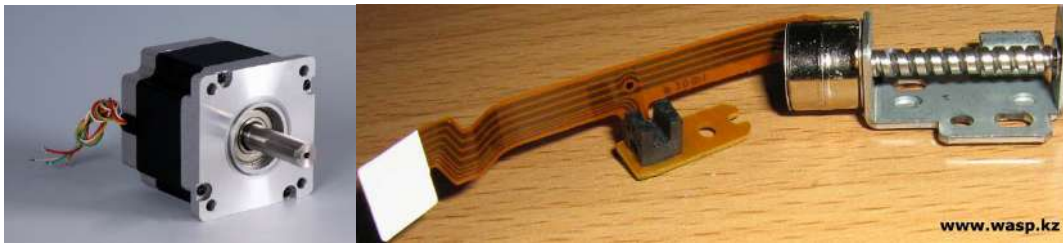


Рисунок 22 Примеры шаговых двигателей

Для нашего примера используем ШД привода головки записи-чтения от 5,25" флоппи-дисковда TEAC.

Это униполярный двигатель с 5 выводами: 4 обмотки и общий провод подачи питания 12 В. ШД имеет шаг 1,8 градуса. Соответственно, для того чтобы вал совершил полный оборот, необходимо подать 200 импульсов. Для небольших униполярных двигателей от CD-привода, принтера или флоппи-дисковда в качестве драйвера можно применить микросхему ULN2003A.

В экспериментальной схеме (рис. 23) используется процессор PIC16F84 от Microchip, однако провести эксперимент с использованием данной программы можно практически на любом процессоре семейства PIC12...PIC18.

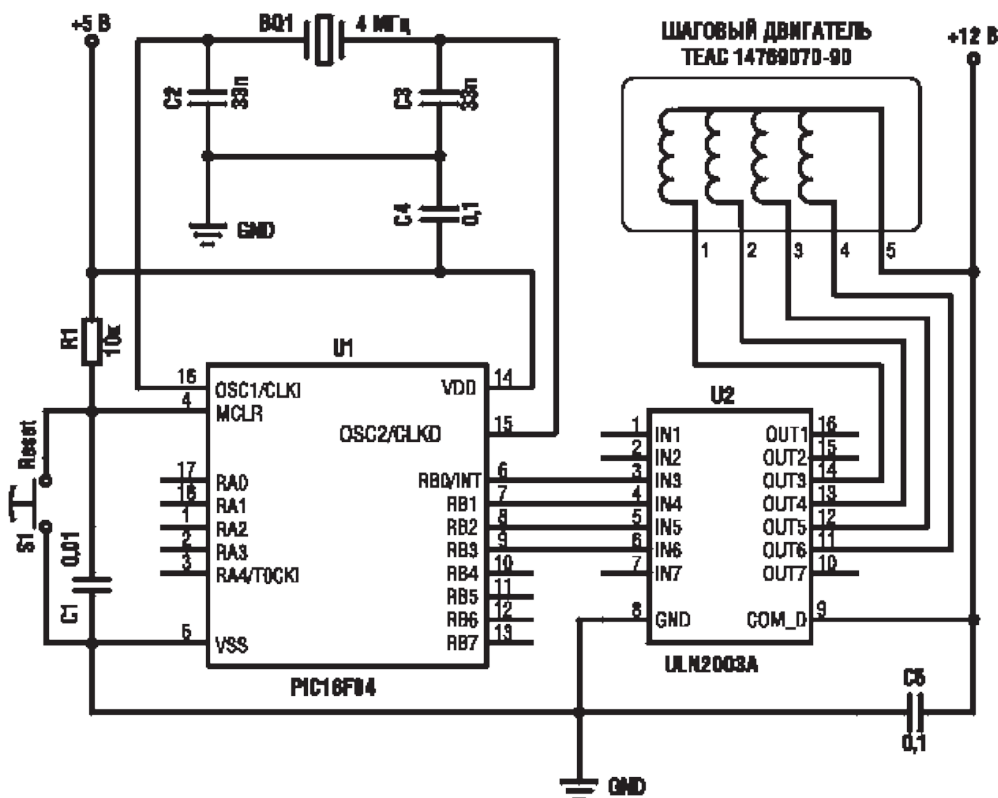


Рисунок 23 Схема подключения шагового двигателя к MCU через драйвер ULN2003A

Программа реализует вращение вала двигателя с заданной скоростью в одном направлении. Используется так называемый «полношаговый режим». Диаграмма подачи импульсов на обмотки приведена на рис. 24. Это простейший способ управления ШД со всеми присущему ему недостатками: повышенная шумность, колебания вала при перемещении от шага к шагу, особенно заметные при наличии массивной нагрузки на валу, малое количество шагов на оборот. В реальных системах используются более

сложные алгоритмы управления ШД. Это режимы плавного разгона, микрошага, использование обратной связи по току и другие. Также может быть реализован «полушаговый» режим управления. Здесь двигатель совершает в два раза больше шагов на оборот, т.е. теперь вал занимает и промежуточные положения между фиксированными углами поворота в 1,8 градуса. Это достигается одновременной подачей напряжения на две соседние обмотки на чётных шагах (рис. 25). Полушаговый режим повышает плавность хода и снижает шум двигателя. Всего программой подаётся 200 импульсов на обмотки двигателя, при этом вал поворачивается на угол 180 градусов. Затем следует пауза в 2 секунды и происходит поворот на тот же угол, но в противоположном направлении.

Следует учитывать, что предельная скорость вращения ограничена конструкцией используемого ШД и не превышает 150...200 оборотов в минуту в зависимости от режима управления

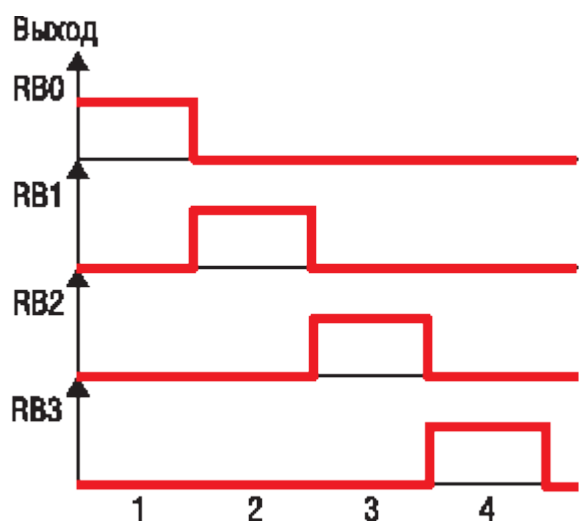


Рисунок 24 Диаграмма подачи импульсов на обмотки в полношаговом режиме

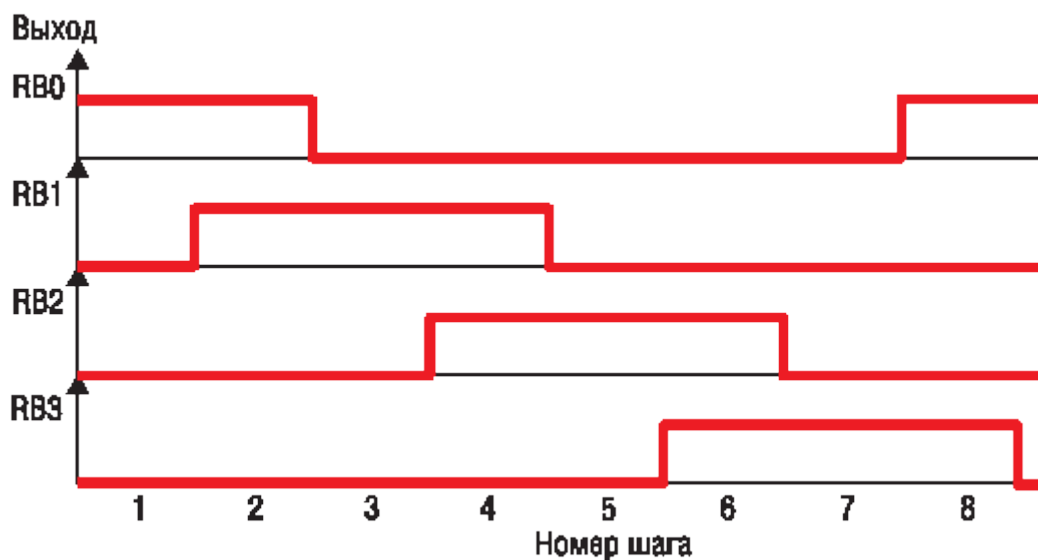


Рисунок 25 Диаграмма подачи импульсов на обмотки в полушаговом режиме использования