

Аналого-цифровой преобразователь микроконтроллеров AVR

Аналого-цифровые преобразователи (АЦП) являются устройствами, которые принимают входные аналоговые сигналы и генерируют соответствующие им цифровые сигналы, пригодные для обработки микропроцессорами и другими цифровыми устройствами. **АЦП** входит во многие современные модели **МК AVR**, он многоканальный. Обычно число каналов равно 8, но в разных моделях оно может варьировать от 4 каналов в младших моделях семейства Tiny, 6 в ATmega8, до 16 каналов в ATmega2560.

Многоканальность означает, что на входе единственного модуля **АЦП** установлен аналоговый мультиплексор, который может подключать этот вход к различным выводам **МК** для осуществления измерений нескольких независимых аналоговых величин с разнесением по времени. Входы мультиплексора могут работать по отдельности (в несимметричном режиме для измерения напряжения относительно "земли") или (в некоторых моделях) объединяться в пары для измерения дифференциальных сигналов. Иногда **АЦП** дополнительно снабжается усилителем напряжения с фиксированными значениями коэффициента усиления 10 и 200.

Сам **АЦП** представляет собой преобразователь последовательного приближения с устройством выборки-хранения и фиксированным числом тактов преобразования, равным 13 (или 14 для дифференциального входа; первое преобразование после включения потребует 25 тактов для инициализации **АЦП**). Тактовая частота формируется аналогично тому, как это делается для таймеров— с помощью специального предделителя тактовой частоты **МК**, который может иметь коэффициенты деления от 1 до 128. Но в отличие от таймеров, выбор тактовой частоты **АЦП** не совсем произволен, т. к. быстродействие аналоговых компонентов ограничено. Поэтому коэффициент деления следует выбирать таким, чтобы при заданном "кварце" тактовая частота **АЦП** укладывалась в рекомендованный диапазон 50-200 кГц (т. е. максимум около 15 тыс. измерений в секунду). Увеличение частоты выборки допустимо, если не требуется достижение наивысшей точности преобразования.

Разрешающая способность **АЦП** в **МК AVR** — 10 двоичных разрядов, чего для большинства типовых применений достаточно. Абсолютная погрешность преобразования зависит от ряда факторов и в идеальном случае не превышает ± 2 младших разрядов, что соответствует общей точности измерения примерно 8 двоичных разрядов. Для достижения этого результата необходимо принимать специальные меры: не только "вгонять" тактовую частоту в рекомендованный диапазон, но и снижать по максимуму интенсивность цифровых шумов. Для этого рекомендуется, как минимум, не использовать оставшиеся выводы того же порта, к которому подключен **АЦП**, для обработки цифровых сигналов, правильно разводите платы, а как

максимум — дополнительно к тому еще и включать специальный режим **ADC Noise Reduction**.

Отметим также, что АЦП может работать в двух режимах: одиночного и непрерывного преобразования. Второй режим целесообразен лишь при максимальной частоте выборки. В остальных случаях его следует избегать, т. к. обойти в этом случае необходимость параллельной обработки цифровых сигналов, как правило, невозможно, а это означает снижение точности преобразования.

Регистры управления АЦП

Ниже приведена таблица с описанием регистра **ADCSR**

	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ATmega8x ATmega163x ATmega323x ATmega128x
Чтение(R)/Запись(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Начальное значение	0	0	0	0	0	0	0	0	
	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ATmega16x ATmega32x ATmega64x
Чтение(R)/Запись(W)	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Начальное значение	0	0	0	0	0	0	0	0	
ATmega8x, ATmega163x, ATmega323x — ADCSR ATmega16x, ATmega32x, ATmega64x/128x — ADCSRA									

Для разрешения работы АЦП необходимо записать лог. 1 в разряд **ADEN** регистра **ADCSR**, а для выключения — лог. 0. Если АЦП будет выключено во время цикла преобразования, то преобразование завершено не будет (в регистре данных АЦП останется результат предыдущего преобразования).

Разряд	Название	Описание
7	ADEN	Разрешение АЦП (1 - включено, 0 - выключено)

Режим непрерывных измерений активизируется установкой бита **ADFR** (бит 5) этого же регистра. В ряде моделей Mega этот бит носит наименование **ADATE**, и управление режимом работы производится сложнее: там добавляются несколько режимов запуска через различные прерывания (в т. ч. прерывание от компаратора, при наступлении различных событий от таймера и т. п.), и выбирать их следует, задавая биты **ADTS** регистра **SFIOR**, а установка бита **ADATE** разрешает запуск АЦП по этим событиям.

Разряд	Название	Описание
5	ADFR(ADATE)	Выбор режима работы АЦП

Так как нулевые значения всех битов **ADTS** (по умолчанию) означают режим непрерывного преобразования, то в случае, когда вы их значения не трогали, функции битов **ADATE** и **ADFR** в других моделях будут совпадать.

ADTS2	ADTS1	ADTS0	Источник стартового сигнала
0	0	0	Режим непрерывного преобразования
0	0	1	Прерывание от аналогового компаратора
0	1	0	Внешнее прерывание INT0
0	1	1	Прерывание по событию "Совпадение" таймера/счетчика T0
1	0	0	Прерывание по переполнению таймера/счетчика T0
1	0	1	Прерывание по событию "Совпадение" таймера/счетчика T1
1	1	0	Прерывание по переполнению таймера/счетчика T1

1	1	1	Прерывание по событию "Захват" таймера/счетчика T1
---	---	---	--

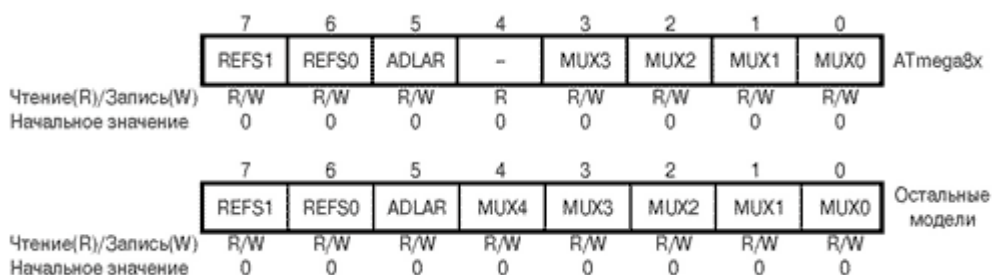
Если выбран режим запуска не от внешнего источника, то преобразование запускается установкой бита **ADSC** (бит 6). При непрерывном режиме установка этого бита запустит первое преобразование, затем они будут автоматически повторяться. В режиме однократного преобразования, а также независимо от установленного режима при запуске через прерывания (в тех моделях, где это возможно) установка бита **ADSC** просто запускает одно преобразование. При наступлении прерывания, запускающего преобразование, бит **ADSC** устанавливается аппаратно. Отметим, что преобразование начинается по фронту первого тактового импульса (тактового сигнала АЦП, а не самого контроллера!) после установки **ADSC**. По окончании любого преобразования (и в одиночном, и в непрерывном режиме) устанавливается бит **ADIF** (бит 4. флаг прерывания). Разрешение прерывания АЦП осуществляется установкой бита **ADIE** (бит 3) все того же регистра **ADCSR/ADCSRA**.

Разряд	Название	Описание
6	ADSC	Запуск преобразования (1 - начать преобразование)
4	ADIF	Флаг прерывания
3	ADIE	Разрешение прерывания

Для работы с АЦП необходимо еще установить его тактовую частоту. Это делается тремя младшими битами регистра **ADCSR/ADCSRA** под названием **ADPS0...2**. Коэффициент деления частоты тактового генератора МК устанавливается по степеням двойки, все нули в этих трех битах соответствуют коэффициенту 2, все единицы — 128. Оптимальная частота преобразования лежит в диапазоне 50-200 кГц, так что, например, для тактовой частоты МК, равной 4 МГц, коэффициент может иметь значение только 32 (состояние битов **ADPS0...2** = 101, частота 125 кГц) или 64 (состояние битов **ADPS0...2** = 110, частота 62,5 кГц). При тактовой частоте 16 МГц в допустимый диапазон укладывается только коэффициент 128.

ADPS2	ADPS1	ADPS0	Коэффициент деления
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Ниже приведена таблица с описанием регистра **ADMUX**.



Выборка источника опорного напряжения производится битами **REFS1...0** регистра **ADMUX** (старшие биты 7 и 6), причем их нулевое значение (по умолчанию) соответствует внешнему источнику. Напряжение этого внешнего источника может лежать в пределах от 2 В до напряжения питания аналоговой части **AVcc** (а оно, в свою очередь, не должно отличаться от питания цифровой части более чем на 0,3 В в большую или меньшую сторону). Можно выбрать в качестве опорного и питание самой аналоговой части, причем двояким способом: либо просто соединить выводы **AREF** и **AVcc** микросхемы, либо установить биты **REFS1...0** в состояние 01 (тогда соединение осуществляется внутренними схемами, но заметим, что внешний опорный источник при этом должен быть отключен).

Предусмотрен и встроенный источник (задается **REFS1...0** в состоянии 11, при этом к выводу **AREF** рекомендуется подключать фильтрующий конденсатор), имеющий номинальное напряжение 2,56В с большим разбросом от 2,4 до 2,7 В.

REFS1	REFS0	Источник опорного напряжения
0	0	Внешний ИОН, подключенный к выводу AREF, внутренний ИОН отключен
0	1	Напряжение питания AVcc*
1	0	Зарезервировано
1	1	Внутренний ИОН напряжением 2,56V, подключенный к выводу AREF*
*Если к выводу AREF подключен источник напряжения, данные варианты использоваться не могут		

Результат преобразования АЦП оказывается в регистрах **ADCH:ADCL**. Поскольку результат 10-разрядный, то по умолчанию старшие 6 битов в регистре **ADCH** оказываются равными нулю. Чтение этих регистров производится, начиная с младшего **ADCL**, после чего регистр **ADCH** блокируется, пока не будет прочитан. Следовательно, даже если момент между чтением регистров попал на фронт 14 (15) такта АЦП, когда данные в них должны меняться, значения прочитанной пары будут соответствовать друг другу, пусть и результат этого преобразования пропадет. В противоположном порядке читать эти регистры не рекомендуется. Но бит **ADLAR** (бит 5 регистра **ADMUX**) предоставляет интересную возможность: если его установить в 1, то результат преобразования в регистрах **ADCH:ADCL** выравнивается влево: бит 9 результата окажется в старшем бите **ADCH**, а незначащими будут младшие 6 битов регистра **ADCL**. В этом случае, если хватает 8-разрядного разрешения результата, можно прочесть только значение **ADCH**.

Выбор каналов и режимов их взаимодействия в АЦП производится битами **MUX0...3** в регистре **ADMUX**. Их значения выбирают нужный канал в обычном (недифференциальном) режиме, когда измеряемое напряжение отсчитывается от "земли". Последние два значения этих битов для семейства Mega (11110 и 11111 в большинстве моделей или 1110 и 1111 для ATmega8) выбирают режимы, когда вход АЦП подсоединяется к опорному источнику компаратора (1,22 В) или к "земле" соответственно, что может использоваться для автокалибровки устройства.

Управление входным мультиплексором в моделях Atmega8x

MUX3-MUX0	Несимметричный вход
0000	ADC0
0001	ADC1
0010	ADC2
0011	ADC3
0100	ADC4*
0101	ADC5*
0110	ADC6**
0111	ADC7**
1000-1101	Зарезервировано
1110	1,22V
1111	0V(GND)
*8-ми разрядное преобразование **Имеются только в корпусах TQFP-32 и MLF-32.	

Остальные комбинации разрядов **MUX** предназначены для установки различных дифференциальных режимов — в тех моделях, где они присутствуют, в других случаях эти биты зарезервированы (как в моделях Atmega8, ATmega163 и др.). В дифференциальном режиме АЦП измеряет напряжение между двумя выбранными выводами (например, между **ADC0** и **ADC1**), причем не все выводы могут быть в таком режиме задействованы. В том числе дифференциальные входы АЦП можно подключать к одному и тому же входу для коррекции нуля. Дело в том, что в ряде моделей на входе АЦП имеется встроенный усилитель, с коэффициентом 1x, 10x и 200x (коэффициент выбирается теми же битами **MUX0...4**), и такой режим используется для его калибровки — в дальнейшем значение выхода при соединенных входах можно просто вычесть.

После завершения преобразования (при установке в «1» флага **ADIF** регистра **ADCSR**) его результат сохраняется в регистре данных АЦП. Поскольку АЦП имеет 10 разрядов, этот регистр физически размещен в двух регистрах ввода/вывода **ADCH:ADCL**, доступных только для чтения. По умолчанию результат преобразования выравнивается вправо (старшие 6 разрядов регистра **ADCH** — незначащие). Однако он может выравниваться и влево (младшие 6 разрядов регистра **ADCL** — незначащие). Для управления выравниванием результата преобразования служит разряд **ADLAR** регистра **ADMUX**. Если этот разряд установлен в «1», результат преобразования выравнивается по левой границе 16-разрядного слова, если сброшен в «0» — по правой границе.

Обращение к регистрам **ADCH** и **ADCL** для получения результата преобразования должно выполняться в определенной последовательности: сначала необходимо прочитать регистр **ADCL**, а затем **ADCH**. Это требование связано с тем, что после обращения к регистру **ADCL** процессор блокирует доступ к регистрам данных со стороны АЦП до тех пор, пока не будет прочитан регистр **ADCH**. Благодаря этому можно быть уверенным, что при чтении регистров в них будут находиться составляющие одного и того же результата. Соответственно, если очередное преобразование завершится до обращения к регистру **ADCH**, результат преобразования будет потерян. С другой стороны, если результат преобразования выравнивается влево и достаточно точности 8-разрядного значения, для получения результата можно прочитать только содержимое регистра **ADCH**.

Для недифференциального режима АЦП, когда напряжение отсчитывается от "земли", результат преобразования определяется формулой:

$$Ka = 1024U_{вх}/U_{ref}$$

где Ka — значение выходного кода АЦП, $U_{вх}$ и U_{ref} — входное и опорное напряжения.

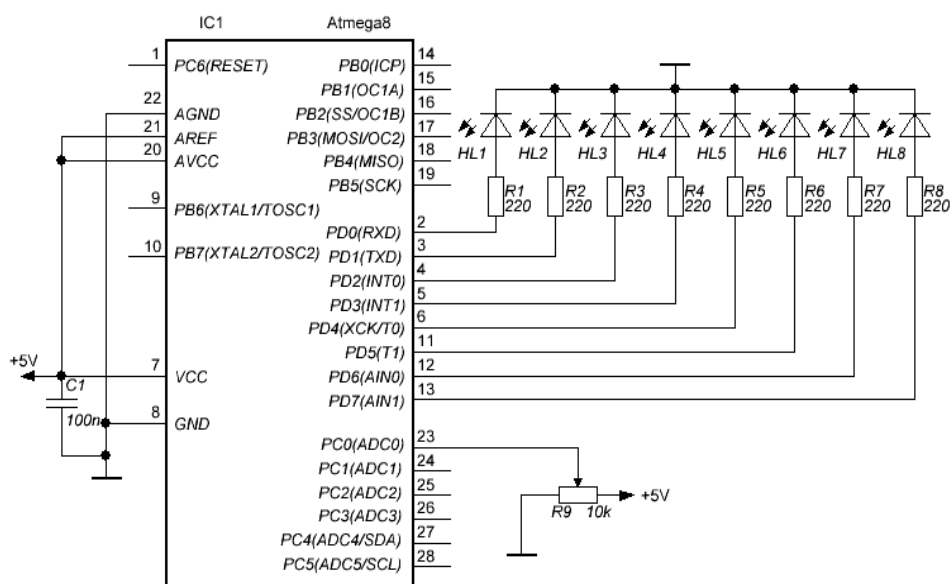
Дифференциальному измерению соответствует такая формула:

$$Ka = 512(U_{pos} - U_{neg})/U_{ref},$$

где U_{pos} и U_{neg} — напряжения на положительном и отрицательном входах соответственно. Если напряжение на отрицательном входе больше, чем на положительном, то результат в дифференциальном режиме становится отрицательным и выражается в дополнительном коде от \$200 (-512) до \$3FF (-1). Реальная точность преобразования в дифференциальном режиме равна 8 разрядам.

Делаем светодиодный индикатор напряжения

Для практического изучения АЦП напишем программу светодиодного индикатора напряжения. Как и в прошлых примерах будем использовать микроконтроллер Atmega8. Восемь индикаторов подключаем к порту D контроллера, это будет линейная шкала уровня сигнала от 0 до 5V. Входом АЦП у нас будет вывод PC0(ADC0), к которому через переменный резистор сопротивлением 10 кОм подается напряжение. Схема устройства представлена ниже:



К точности АЦП в этом устройстве предъявляются наименьшие требования. Источником опорного напряжения служит напряжение питания микроконтроллера - 5 В, для этого вывод AREF соединяем с выводом Vcc микроконтроллера, также поступаем с выводами питания

аналоговой части **AVcc** и **AGND**, подключаем их к плюсу и минусу соответственно, в программе битами **REFS1** и **REFS0** задаем источник **ИОН**.

Режим индикации работает следующим образом: после окончания преобразования, которое работает в непрерывном режиме, считываем биты **ADCH** и **ADCL**. Это значение потом сравниваем с предварительно рассчитанными константами. Если значение **ADC** больше константы загорается один светодиод, если значение **ADC** больше второй константы загораются уже два светодиода и т.д.

Константы высчитываются так: так как АЦП 10-ти битный, число 1024 раскладываем на 8 равных частей, а по формуле уже вычисляем эти значения в Вольтах.

```
128...0,625V
256...1.25V
384...1.875V
512...2.5V
640...3.125V
768...3.75V
896...4.375V
1020...5V (приблизительно)
```

Полный код программы показан ниже. Частота тактового генератора контроллера 8MHz.

```
/**/ Использование АЦП. Светодиодная шкала ***/

#include <avr/io.h>
#include <util/delay.h>

int main (void)
{
  DDRD = 0xFF;
  PORTD = 0x00;

  /**/ Настройка АЦП ***/
  ADCSRA |= (1 << ADEN) // Включение АЦП
            |(1 << ADPS1)|(1 << ADPS0); // делитель преобразователя на 8
  ADMUX |= (0 << REFS1)|(0 << REFS0) // внешний ИОН
            |(0 << MUX0)|(0 << MUX1)|(0 << MUX2)|(0 << MUX3); // вход PC0
```

```

while(1)
{
unsigned int u;
ADCSRA |= (1 << ADSC); // Начинаем преобразование
while ((ADCSRA&(1 << ADIF))== 0); // Ждем флага окончания
преобразования

u = (ADCL|ADCH << 8); // Считываем ADC

if (u > 128) // 0.625V
PORTD = 0b00000001;
else
PORTD = 0b00000000;

if (u > 256) // 1.25V
    PORTD = 0b00000011;
if (u > 384) // 1.875V
    PORTD = 0b00000111;
if (u > 512) // 2.5V
    PORTD = 0b00001111;
if (u > 640) // 3.125V
    PORTD = 0b00011111;
if (u > 768) // 3.75V
    PORTD = 0b00111111;
if (u > 896) // 4.375V
    PORTD = 0b01111111;
if (u > 1020) // 5V
    PORTD = 0b11111111;

_delay_ms(30);
}
}

```

Bit	Low	High
7	<input type="checkbox"/> BODLEVEL Brown out detector trigger level	<input type="checkbox"/> RSTDISBL Disable reset
6	<input type="checkbox"/> BODEN Brown out detector enable	<input type="checkbox"/> WTDON Enable watchdog
5	<input checked="" type="checkbox"/> SUT1 Select start-up time	<input checked="" type="checkbox"/> SPIEN Enable Serial programming and Data Downloading
4	<input type="checkbox"/> SUTO Select start-up time	<input type="checkbox"/> CKOPT Oscillator Options
3	<input checked="" type="checkbox"/> CKSEL3 Select Clock Source	<input type="checkbox"/> EESAVE EEPROM memory is preserved through chip erase
2	<input type="checkbox"/> CKSEL2 Select Clock Source	<input checked="" type="checkbox"/> BOOTSZ1 Select Boot Size
1	<input checked="" type="checkbox"/> CKSEL1 Select Clock Source	<input checked="" type="checkbox"/> BOOTSZ0 Select Boot Size
0	<input checked="" type="checkbox"/> CKSELO Select Clock Source	<input type="checkbox"/> BOOTRST Select Reset Vector

Задания для выполнения:

1. Составить конспект по данной лекции.
2. Изучить принцип работы АЦП на основе микроконтроллера.
3. Составить программу и смоделировать в среде Proteus светодиодный индикатор входного напряжения.

Результат и архив с проектом отправить на e-mail: rasov@rambler.ru с темой АЦП_ФИО