Лабораторная работа № 5

# Использование динамической индикации

Цель работы:

1. Изучение принципа динамической индикации.
2. Построение многоразрядного индикатора для микроконтроллера

ATmega8535.

Оборудование и программное обеспечение:

1. Лабораторный макет MK8535 в составе: модуль базовый МБ8535 +

модуль сменный МС02.

1. Кабель соединительный для программирования МК.
2. Персональный компьютер.
3. Интегрированная среда программирования CodeVisionAVR.

# Описание схемы лабораторного макета

В сменном модуле МС02 имеются 4 светодиодных семисегмент- ных индикатора с общим анодом. Структурная схема лабораторного макета приведена на рис. 39. Все индикаторы подключены параллельно к порту C. Транзисторные ключи VT1…VT4 служат для зажигания ин- дикаторов. Для этого на базы транзисторов программным способом нужно обеспечить поочерёдную подачу 1. Кнопки S1 и S2 подключены к входам внешних прерываний INT0 и INT1.

*Таблица 8*

Комбинации бит для цифр от 0 до 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Цифра | 0 | | | 1 | | | 2 | | | 3 | | | 4 | | |
| Изображение на индикаторе |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Двоич. зн-е | **01000001** | | | **01110111** | | | **00101001** | | | **00100011** | | | **00010111** | | |
| Шестн. зн-е | **0x41** | | | **0x77** | | | **0x29** | | | **0x23** | | | **0x17** | | |
| Цифра | 5 | | | 6 | | | 7 | | | 8 | | | 9 | | |
| Изображение на индикаторе |  |  |  |  |  |  |  |  |  |  |  |  |  | | |
| Двоич. зн-е | **10000011** | | | **10000001** | | | **01100111** | | | **00000001** | | | **00000011** | | |
| Шестн. Зн-е | **0x83** | | | **0x81** | | | **0x67** | | | **0x01** | | | **0x03** | | |

В модуле МС02 выводы индикаторов подключены к разрядам порта C произвольно, исходя из удобства трассировки печатной платы. Чтобы правильно высветить цифры, нужно разработать для них соответ- ствующие комбинации бит, выводимых в порт С (это и будет программ- ный дешифратор). В табл. 8 приведены комбинации бит для цифр от 0 до 9 с учетом того, что сегменты зажигаются 0 на выходе порта.

Приведенные в таблице коды не зажигают сегмент точки. Для того чтобы отдельно управлять зажиганием точки, нужно выводить 0 в разряд 0 порта C.

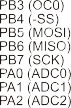
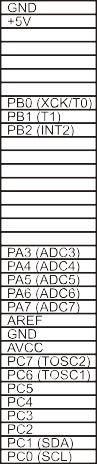
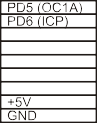


Рис. 39. Структурная схема лабораторного макета. Для модуля МС02 показана только та часть схемы, которая используется в данной работе

# Подготовка к работе

Рассмотрим и выполним пример, где на 4-разрядный десятичный индикатор выводится целое число. Это число хранится в ячейке памяти МК и может быть увеличено или уменьшено по прерыванию с помощью кнопок, подключенных ко входам INT0 и INT1 (см. рис. 39). Пусть по нажатию кнопки S1 происходит увеличение значения счетчика, а по нажатию кнопки S2 – уменьшение. Основная программа обеспечивает непрерывную динамическую индикацию числа.

Создадим проект в CodeVisionAVR и зададим нужный режим ра- боты МК. Линии 4…7 порта A настроить на вывод. Их начальное значе- ние – 0. Линии 0…7 порта C также настроить на вывод. Их начальное значение – 1. Входы INT0 и INT1 настроить на прерывание по спадаю- щему фронту.

Ниже приведен текст программы для МК ATmega8535. В нем опущены те строки, автоматически сформированные **CodeWizardAVR**, которые не относятся непосредственно к данному примеру.

Структура **dig** служит для хранения значений цифр, выводимых на индикатор. Эти значения служат индексами для массива **BitSet[10]**, элементами которого являются 10 8-битовых двоичных комбинаций для изображения цифр от 0 до 9. Так образуется программ- ный дешифратор.

Функция **IntToDigits(int v)** осуществляет выделение 4 десятичных цифр целого числа **v** и запись их в структуру **dig**. Число в данном примере должно быть в диапазоне 0…9999.

Функция **ShowDigits()** осуществляет один цикл динамической индикации числа. Для обеспечения непрерывной индикации её следует вызывать периодически.

Процедуры обработки внешних прерываний обеспечивают увели- чение и уменьшение значения переменной **cnt**, которое и выводится на индикатор.

# #include <mega8535.h> #include <delay.h>

**struct**

**{**

**unsigned char d0; // значение 0 разряда unsigned char d1; // значение 1 разряда unsigned char d2; // значение 2 разряда**

**unsigned char d3; // значение 3 разряда**

**} dig;**

**void IntToDigits(int v)**

**{**

**dig.d0 = v%10; // выделить 0 разряд**

**v /= 10;**

**dig.d1 = v%10; // выделить 1 разряд**

**v /= 10;**

**dig.d2 = v%10; // выделить 2 разряд**

**v /= 10;**

**dig.d3 = v%10; // выделить 3 разряд**

**}**

# void ShowDigits()

**{**

**// таблица дешифратора цифр 0...9**

**static unsigned char BitSet[10] = {0x41,0x77, 0x29,0x23,0x17,0x83,0x81,0x67,0x01,0x03};**

**PORTC = BitSet[dig.d0]; // цифра 0 разряда PORTA = 0x10; // зажечь 0 разряд delay\_ms(1);**

**PORTA = 0x00; // погасить 0 разряд PORTC = BitSet[dig.d1]; // цифра 1 разряда PORTA = 0x20; // зажечь 1 разряд delay\_ms(1);**

**PORTA = 0x00; // погасить 1 разряд PORTC = BitSet[dig.d2]; // цифра 2 разряда PORTA = 0x40; // зажечь 2 разряд delay\_ms(1);**

**PORTA = 0x00; // погасить 2 разряд PORTC = BitSet[dig.d3]; // цифра 3 разряда PORTA = 0x80; // зажечь 3 разряд delay\_ms(1);**

**PORTA = 0x00; // погасить 3 разряд**

**// External Interrupt 0 service routine interrupt [EXT\_INT0] void ext\_int0\_isr(void)**

**{**

# if (cnt < 9999) IntToDigits(++cnt);

**}**

# // External Interrupt 1 service routine interrupt [EXT\_INT1] void ext\_int1\_isr(void)

**{**

# if (cnt > 0) IntToDigits(--cnt);

**}**

# void main(void)

**{**

**PORTA = 0x00; DDRA = 0xF0; // Port A init. PORTC = 0xFF; DDRC = 0xFF; // Port C init.**

**// External Interrupt(s) initialization**

**// INT0: On**

**// INT0 Mode: Falling Edge**

**// INT1: On**

**// INT1 Mode: Falling Edge**

**// INT2: Off GICR |= 0xC0; MCUCR = 0x0A; MCUCSR = 0x00; GIFR = 0xC0;**

**#asm(«sei») // Global enable interrupts**

**IntToDigits(cnt); while (1)**

**{**

# ShowDigits();

**}**

**}**

**}**

# int cnt = 2013;

**Задание**

1. Создать проект и реализовать рассмотренный выше пример программы.
2. Доработать программу так, чтобы избавиться от влияния дребезга контактов кнопок S1 и S2.
3. В процедуре **ShowDigits()** увеличить все задержки времени до

50 мсек (**delay\_ms(50)**), чтобы наблюдать замедленную работу динамической индикации. Проверить, начиная с какой частоты мер- цание индикатора уже не ощущается.

1. Написать программу для вывода на индикатор шестнадцатеричных целых чисел. Для этого понадобится дополнить программный дешифратор (см. табл. 8) изображениями цифр A, B, C, D, E, F.

# ВНУТРЕННИЙ АЦП В МИКРОКОНТРОЛЛЕРАХ AVR

**Общие сведения**

Аналого-цифровой преобразователь (АЦП) – это электронный узел, на вход которого поступает аналоговый сигнал, а на выходе фор- мируется цифровой код, пропорциональный уровню входного сигнала.

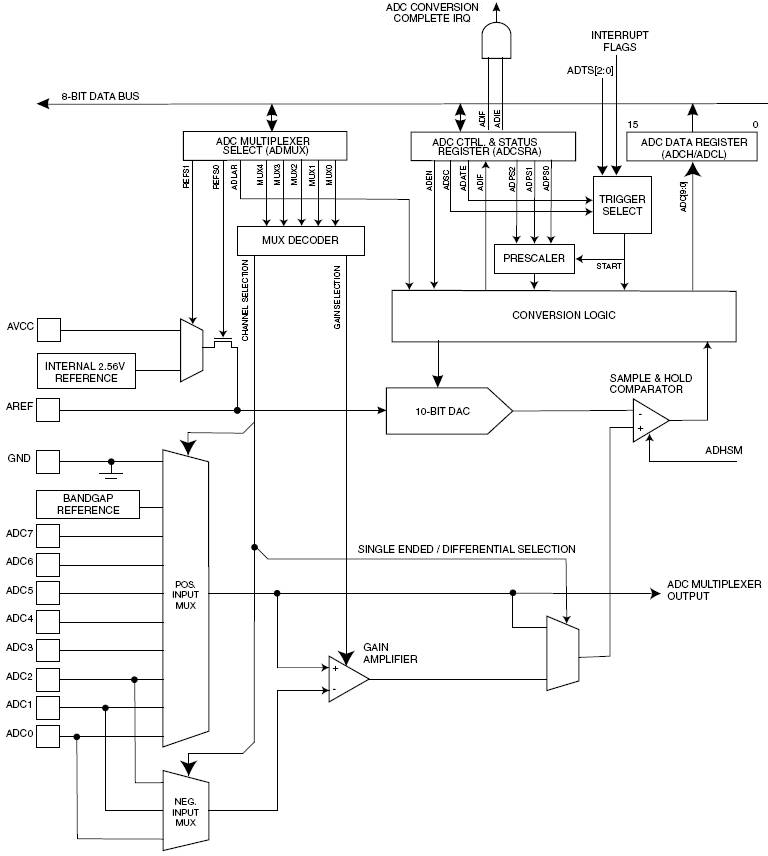


Рис. 40. Структурная схема АЦП

В состав современных микроконтроллеров, как правило, входит модуль АЦП. МК ATmega8535 содержит 10-разрядный АЦП последова- тельного приближения (рис. 40). Основные параметры этого АЦП:

Абсолютная погрешность ±2 МЗР

Интегральная нелинейность ±0.5 МЗР

Быстродействие до 15 тыс. выборок/сек.

На входе модуля АЦП имеется 8-канальный аналоговый мульти- плексор, позволяющий подключиться к одному из выводов порта A (в этом случае выводы порта A выполняют альтернативную функцию аналоговых входов). Входы мультиплексора являются несимметричными, т.е. входное напряжение измеряется относительно 0 В (GND). Кроме того, входы мультиплексора могут использоваться и попарно как диф- ференциальные. Имеется также возможность 20- и 200-кратного предва- рительного усиления входного сигнала.

В качестве источника опорного напряжения для АЦП может использоваться как напряжение питания микроконтроллера, так и внут- ренний либо внешний источник опорного напряжения.

Длительность цикла преобразования АЦП составляет 13 тактов.

В процессе работы АЦП может функционировать в двух режимах:

1. Режим одиночного преобразования, когда запуск каждого преобразо- вания инициируется пользователем.
2. Режим непрерывного преобразования, когда запуск преобразований выполняется непрерывно через определенные интервалы времени.

# Функционирование модуля АЦП

Для задания режимов работы и управления АЦП в МК

ATmega8535 используются следующие регистры:

**ADCSRA** – регистр управления и состояния **SFIOR** – регистр специальных функций **ADMUX** – регистр управления мультиплексором

**ADCL** и **ADCH** – младший и старший регистр результата

Чтобы воспользоваться АЦП, нужно выполнить следующие дей-

1. Выбрать входной канал и источник опорного напряжения, записав соответствующие разряды регистра **ADMUX**.
2. Выбрать условия запуска и частоту преобразования, записав соответ- ствующие разряды регистров **ADCSRA** и **SFIOR**.
3. Запустить АЦП и по окончании преобразования забрать результат из регистров **ADCL** и **ADCH**.

Формат регистра ADCSRA приведен на рис. 41, а краткое описа- ние функций его разрядов приведено в табл. 9.

7 6 5 4 3 2 1 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ADEN** | **ADSC** | **ADATE** | **ADIF** | **ADIE** | **ADPS2** | **ADPS1** | **ADPS0** |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Рис. 41. Формат регистра ADCSRA микроконтроллера ATmega8535

*Таблица* 9

Разряды регистра ADCSRA микроконтроллера ATmega8535

|  |  |  |
| --- | --- | --- |
| Разряд | Название | Описание |
| 7 | ADEN | Разрешение АЦП  (1 – вкл., 0 – выкл.) |
| 6 | ADSC | Запуск преобразования (1 – пуск) |
| 5 | ADATE | Режим запуска АЦП |
| 4 | ADIF | Флаг «преобразование завершено» |
| 3 | ADIE | Разрешение прерывания от АЦП |
| 2…0 | ADPS2…ADPS0 | Выбор частоты преобразования |

Формат регистра ADMUX приведен на рис. 42, а краткое описа- ние функций его разрядов приведено в табл. 10.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **REFS1** | **REFS0** | **ADLAR** | **MUX4** | **MUX3** | **MUX2** | **MUX1** | **MUX0** |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Рис. 42. Формат регистра ADMUX микроконтроллера ATmega8535

ствия:

*Таблица 10*

Разряды регистра ADMUX микроконтроллера ATmega8535

|  |  |  |
| --- | --- | --- |
| Разряд | Название | Описание |
| 7, 6 | REFS1, REFS0 | Выбор источника опорного напряжения |
| 5 | ADLAR | Выравнивание результата преобразования |
| 4…0 | MUX4…MUX0 | Выбор входного канала |

Подробные сведения о назначении разрядов регистра ADMUX можно найти в техническом описании ATmega8535. Например, значение 0x40 в ADMUX означает использование напряжения аналогового пита- ния AVCC в качестве опорного, результат преобразования будет вы- ровнен вправо, а вход АЦП подключен к выводу 0 порта A.

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

Режим запуска АЦП определяется состоянием разряда ADATE. Если он установлен в 1, АЦП работает в режиме непрерывного преобра- зования. В этом режиме запуск каждого следующего преобразования осуществляется автоматически после окончания текущего. Если же раз- ряд ADATE сброшен в 0, АЦП работает в режиме одиночного преобра- зования и запуск каждого преобразования осуществляется по команде пользователя.

В микроконтроллере ATmega8535 запуск АЦП возможен не только по команде пользователя, но и по прерыванию от некоторых перифе- рийных устройств, имеющихся в составе микроконтроллера. Для выбора режима запуска используется разряд ADATE регистра ADCSRA и раз- ряды ADTS2…ADTS0 регистра SFIOR.

Формат регистра SFIOR приведен на рис. 43.

*Таблица 11*

Источник сигнала для запуска преобразования ATmega8535

|  |  |  |  |
| --- | --- | --- | --- |
| 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 Т0 |
| 1 | 0 | 0 | Прерывание по переполнению таймера/счетчика Т0 |
| 1 | 0 | 1 | Прерывание по событию «Совпадение В» таймера/счетчика Т1 |
| 1 | 1 | 0 | Прерывание по переполнению таймера/счетчика Т1 |
| 1 | 1 | 1 | Прерывание по событию «Захват»  таймера/счетчика Т1 |

Запуск каждого преобразования в режиме одиночного преобразо- вания, а также запуск первого преобразования в режиме непрерывного преобразования осуществляется установкой в 1 разряда ADSC регистра АDCSRA. Этот разряд остается равным 1 в течение всего времени пре- образования и конце будет сброшен аппаратно.

Запуск преобразования по прерыванию осуществляется при уста- новке в 1 флага выбранного прерывания. Разряд ADSC регистра ADCSRA при этом аппаратно устанавливается в 1. Запуск преобразова- ния в этих режимах также может быть осуществлен установкой в 1 раз- ряда ADSС регистра ADCSRA.

Длительность цикла составляет 13 тактов при использовании несимметричного входа и 13 либо 14 тактов при использовании диффе- ренциального входа; выборка и запоминание входного сигнала осу-

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | ществляется в течение 1,5 и 2,5 тактов соответственно. Через 13 (14) |
| **ADTS2** | **ADTS1** | **ADTS0** | **ADHSM** | **ACME** | **PUD** | **PSR2** | **PSR10** | тактов преобразование завершается, разряд ADSC аппаратно сбрасыва- |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | ется в 0 (в режиме одиночного преобразования) и результат преобразо- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | вания сохраняется в регистре данных АЦП. Одновременно устанавлива- |

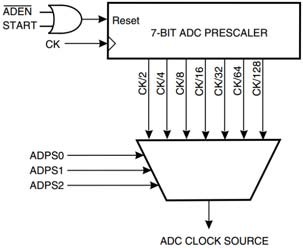
Рис. 43. Формат регистра SFIOR микроконтроллера ATmega8535

Когда разряд ADATE регистра ADCSRA установлен в 1, источ- ник сигнала запуска АЦП определяется содержимым разрядов ADTS2…ADTS0 согласно табл. 11.

ется флаг прерывания ADIF регистра ADСSRA и генерируется запрос на прерывание. Как и флаги остальных прерываний, флаг ADIF сбрасыва- ется аппаратно при запуске подпрограммы обработки прерывания от АЦП или программно, записью в него лог. 1. Разрешение прерывания

осуществляется установкой в 1 разряда ADIE регистра АDСSRA при установленном флаге I регистра SREG.

Для формирования тактовой частоты модуля АЦП в нем имеется отдельный предделитель (рис. 44). Коэффициент деления предделителя и, соответственно, длительность преобразования определяется состоя- нием разрядов ADPS2...ADPS0 регистра ADCSRA (табл. 12).

*Таблица 12*

Задание коэффициента деления предделителя АЦП

|  |  |  |  |
| --- | --- | --- | --- |
| ADPS2 | ADPS1 | ADPS0 | Коэф.  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 |

Рис. 44. Предделитель АЦП

микроконтроллера, а затем вычитать полученное значение из результата последующих преобразований. Таким образом, ошибка смещения может быть снижена до величины, меньшей 1 МЗР.

Состояние разрядов MUX4...MUX0 можно изменить в любой момент, однако, если это будет сделано во время цикла преобразова- ния, смена канала произойдет только после завершения преобразова- ния. Благодаря этому в режиме непрерывного преобразования можно легко осуществлять последовательное преобразование сигналов не- скольких каналов.

Отдельно следует сказать о каналах с дифференциальным входом. После смены таких каналов первое измерение следует производить не ранее чем через 125 мкс после выбора канала. Указанное время требует- ся для установления значения коэффициента усиления предусилителя. Соответственно значения, измеренные до истечения этого срока, не могут считаться достоверными.

Как уже было отмечено, модуль АЦП может использовать раз- личные источники опорного напряжения (ИОН). Выбор конкретного источника опорного напряжения осуществляется с помощью разрядов REFS1:REFS0 регистра ADMUX (табл. 13).

Наибольшая точность преобразования достигается, если тактовая частота модуля АЦП находится в диапазоне 50...200 кГц. Соответственно коэффициент деления предделителя рекомендуется выбирать таким, что- бы тактовая частота модуля АЦП находилась в указанном диапазоне. Если же точности преобразования меньше 10 разрядов достаточно, можно использовать более высокую частоту, увеличивая тем самым частоту вы- борки. Для этой цели предназначен разряд ADCHM регистра SFIOR. При установке этого разряда в 1 скорость преобразования АЦП увеличивается. Однако при этом увеличивается и потребляемая мощность микро- контроллера.

Выводы микроконтроллера, подключенные к входу АЦП, опреде- ляются состоянием разрядов MUX4...MUX0 регистра ADMUX. Для

Выбор источника опорного напряжения

# Результат преобразования

|  |  |  |
| --- | --- | --- |
| 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 2,56 В |

*Таблица 13*

каналов с дифференциальным входом указанные разряды определяют также коэффициент предварительного усиления входного сигнала.

Следует отметить, что предварительный усилитель, используе- мый каналами с дифференциальным входом, имеет встроенную схему коррекции смещения. Оставшаяся после коррекции величина смешения может быть учтена программным способом. Для этого следует оба входа дифференциального усилителя подключить к одному и тому же выводу

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 раз- мешен в двух регистрах ввода/вывода ADCH:ADCL, доступных только для чтения. Эти регистры расположены по адресам $05:$04 и при вклю- чении микроконтроллера содержат значение $0000.

По умолчанию результат преобразования выравнивается вправо (старше 6 разрядов регистра ADCH – незначащие). Однако он может выравниваться и влево (младшие 6 разрядов регистра ADCL – незнача- щие). Для управления выравниванием результата преобразования слу- жит 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 последовательно- сти: сначалa необходимо прочитать регистр ADCL, а затем ADCH. Это требование связано с тем, что после обращения к регистру ADCL про- цессор 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 со- ставляющие одного и того же результата. Соответственно, если очеред- ное преобразование завершится до обращения к регистру ADCH, ре- зультат преобразования будет потерян. С другой стороны, если резуль- тат преобразования выравнивается влево и достаточно точности 8-разрядного значения, для получения результата можно прочитать только содержимое регистра ADCH.

Для каналов с несимметричным входом результат преобразования определяется выражением

Результат преобразования представляется в этом случае в допол- нительном коде, а его значение лежит в диапазоне от 0x200 (–512) до 0x1FF (+511).

# Повышение точности преобразования

Для сведения к минимуму электромагнитных помех, наводимых ядром процессора, имеется дополнительный «спящий» режим – ADC Noise Reduction (режим снижения шумов АЦП). В этом режиме из всех периферийных устройств функционируют только АЦП и сторожевой таймер.

Для использования АЦП в этом режиме необходимо убедиться, что АЦП включен и не занят преобразованием, затем переключить АЦП в режим одиночного преобразования и разрешить прерывание от АЦП, после чего перевести микроконтроллер в режим ADC Noise Reduction. Сразу же после остановки процессора начнется цикл преобразования.

При завершении преобразования будет сгенерировано прерыва- ние от АЦП, которое переведет микроконтроллер в рабочий режим, и начнется выполнение подпрограммы обработки этого прерывания.

*ADC*  *VIN* 1024 ,

*VREF*

где *VIN –* значение входного напряжения, a *VREF* – величина выбранного опорного напряжения. Например, ADC = 0x000 представляет 0 вольт на входе, а ADC = 0x3FF соответствует напряжению на входе, равному *VREF*, минус 1 МЗР.

Для каналов с дифференциальным входом результат преобразо- вания определяется выражением

*ADC*  (*VPOS*  *VNEG* )  *K*  512 ,

*VREF*

где *VPOS* – величина напряжения на положительном входе, *VNEG* – вели- чина напряжения на отрицательном входе, а *К –* выбранный коэффици- ент усиления предусилителя.