

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«ВОЛОГОДСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»
(ВоГУ)

Компьютерные технологии

Методические указания к выполнению контрольных работ.
Часть № 2

Направление подготовки: 13.02.03 – ЭЛЕКТРОЭНЕРГЕТИКА И
ЭЛЕКТРОТЕХНИКА

Профиль подготовки: электроснабжение

Квалификация (степень) выпускника: бакалавр

Форма обучения: заочная

Факультет: электроэнергетический

Кафедра: электроснабжения

Вологда

2017

ВВЕДЕНИЕ

Принимая во внимание огромное влияние современных информационных технологий на процесс образования, многие педагоги все с большей готовностью включают их в свою методическую систему. Проникновение современных информационных технологий в сферу образования позволяет педагогам качественно изменить содержание, методы и организационные формы обучения. Целью внедрения этих технологий в образовании являются гуманизация, индивидуализация, интенсификация процесса обучения и повышение качества обучения на всех ступенях образовательной системы.

Под информационными и коммуникационными технологиями предлагается понимать комплекс объектов, действий и правил, связанных с подготовкой, переработкой и доставкой информации при персональной, массовой и производственной коммуникации, а также все технологии и отрасли, интегрально обеспечивающие перечисленные процессы.

В последние годы термин «информационные технологии» часто выступает синонимом термина «компьютерные технологии», так как все информационные технологии в настоящее время так или иначе связаны с применением компьютера. Однако, термин «информационные технологии» намного шире и включает в себя «компьютерные технологии» в качестве составляющей. При этом информационные технологии, основанные на использовании современных компьютерных и сетевых средств, образуют термин «Современные информационные технологии».

2. Микроконтроллеры

Под микроконтроллером в общем случае понимают микропроцессорное устройство, способное выполнять ограниченный набор функций. Первым прототипом микроконтроллера считается разработанный в 1971 году сотрудниками Texas Instruments, запатентовавшими свое изобретение под названием «однокристалльная микро-ЭВМ». Отличительной особенностью этого устройства является размещение непосредственно на кристалле не только вычислительного ядра, но и запоминающего устройства, хранящего инструкции и данные, устройства ввода-вывода, а также набор встроенных периферийных устройств. Первым по настоящему коммерчески успешным считается выпущенный в 1980 году фирмой Intel микроконтроллер i8051.

Наибольшее распространение микроконтроллеры получили во встроенных системах контроля и управления. Главной причиной популярности микроконтроллеров служит тот факт, что они являются практически полностью готовыми вычислительными устройствами, не требующими для своей работы дополнительного оборудования. Кроме того, возможность программировать работу микроконтроллера позволяет реализовывать достаточно сложные электронные устройства, в которых большая часть функционала (иногда до 90%) реализуется программно.

В настоящее время на рынке микроконтроллеров активно работают более 30 разработчиков и изготовителей. Производители предлагают широкий ассортимент микроконтроллеров, отличающихся как техническими характеристиками, так и перечнем встроенных периферийных устройств, благодаря чему разработчики имеют возможность подобрать микроконтроллер, который наиболее подходит для решения конкретной задачи.

Критериями для выбора микроконтроллеров чаще всего служат:

- Быстродействие.
- Габаритные размеры и тип корпуса.
- Энергопотребление, наличие энергосберегающего режима работы, необходимость охлаждения.
- Встроенные периферийные устройства, начиная от EEPROM-памяти и заканчивая LAN или LCD-контроллером.
- Надежность.

Еще одной важной характеристикой, влияющей как на практичность, так и на цену устройства, является способ программирования:

Перепрограммируемые микроконтроллеры с УФ или электрическим стиранием, являются самыми дорогими, но вместе с тем, и наиболее

практичными устройствами для мелкосерийного и экспериментального производства.

Однократно-программируемые микроконтроллеры дешевле перепрограммируемых, однако, программирование возможно только один раз.

Идея разработки нового прогрессивного RISC-ядра зародилась в норвежском городе Тронхейм (Trondheim) в светлых головах двух студентов Norwegian University of Science and Technology (NTNU). Звали изобретателей Альф Боген (Alf-Egil Bogen) и Вегард Воллен (Vegard Wollen). Находясь в очаровательном окружении смеси университетских зданий, вычислительных центров, будущие директора Atmel Norway создали архитектуру, которая стала одной из самых удачных на мировом рынке микроконтроллеров.

В 1995 году Боген и Воллен решили предложить американской корпорации Atmel, известной на тот момент своим "ноу-хау" изготовления чипов с Flash-памятью, выпускать новый 8-битный RISC-микроконтроллер и снабдить его Flash-памятью программ на кристалле. В 1996 году был основан исследовательский центр Atmel в Тронхейме. 150-тысячный Тронхейм усилиями своего университета каждый год порождает до 20-ти новых компаний, специализирующихся в секторах рынка начиная от автоматизации и до передачи и обработки данных. В конце 1996 года был выпущен опытный кристалл AT90S1200, а во второй половине 1997-го корпорация Atmel приступила к серийному производству нового семейства микроконтроллеров, к их рекламной и технической поддержке.

Новое ядро было запатентовано и получило название AVR, которое по прошествии уже нескольких лет стало трактоваться самыми различными способами. Кто-то утверждает, что это не иначе как Advanced Virtual RISC, другие полагают, что не обошлось здесь без Alf Egil Bogen Vegard Wollan RISC. Держателями патента при этом являются: Wollan, Vegard (NO); Bogen, Alf-Egil (NO); Myklebust, Gaute(NO); Bryant, John, D. (US).

Самыми популярными на сегодняшний день являются микроконтроллеры серии AVR фирмы Atmel.

Кроме центрального процессора, памяти и портов ввода/вывода, любой микроконтроллер AVR содержит обязательный набор так называемых периферийных устройств. Периферийные они по отношению к ЦПУ микроконтроллера. Но находятся они так же внутри микросхемы. Перечислим эти устройства:

Встроенные таймеры/счетчики. Микроконтроллеры AVR содержат несколько их видов. Есть восьми и шестнадцатиразрядные таймеры. Их количество меняется от одного до трех.

Генератор сигнала с широтно-импульсной модуляцией (ШИМ)

2.1. Аналоговый компаратор.

Аналогово-цифровой преобразователь (АЦП). АЦП микроконтроллеров AVR имеют либо 6, либо 8 каналов (могут преобразовывать в цифры до 8 входных аналоговых сигналов).

Последовательный приемо-передатчик. Предназначен для организации последовательного канала связи с другими устройствами. Например, с СОМ-портом компьютера.

Специализированный последовательный канал SPI. Может применяться для обмена информацией между разными микроконтроллерами. А так же для программирования памяти программ в последовательном режиме.

2.2. Память программ

AVR имеет Flash-память программ, которая может быть загружена как с помощью обычного программатора, так и с помощью SPI-интерфейса, в том числе непосредственно на целевой плате. Число циклов перезаписи - не менее 1000. Последние версии кристаллов семейства "mega" выпуска 2001-2002 года имеют возможность самопрограммирования. Это означает, что микроконтроллер способен самостоятельно, без какого-либо внешнего программатора, изменять содержимое ячеек памяти программ. То есть, новые AVR могут менять алгоритмы своего функционирования и программы, заложенные в них, и далее работать уже по измененному алгоритму или новой программе. Например, можно написать и сохранить несколько рабочих версий программы для конкретного приложения во внешней энергонезависимой памяти (DataFlash, SEEPROM и т.п.), а затем по мере необходимости или по реакции на какие-нибудь внешние или внутренние логические условия перегружать рабочие программы в тот же самый микроконтроллер AVR, не извлекая его из печатной платы. Для этого весь массив памяти программ делится на две неравные по объему области: блок загрузчика (программа, управляющая перезаписью Flash-памяти программ) и блок для размещения рабочего программного кода, причем свободная память в области загрузчика может быть использована в качестве дополнительного пространства для рабочего кода. Программа - загрузчик создается самим разработчиком и должна быть запрограммирована внешним программатором.

2.3. Память данных

AVR имеет также блок энергонезависимой электрически стираемой памяти данных EEPROM. Этот тип памяти, доступный программе микроконтроллера непосредственно в ходе ее выполнения, удобен для хранения промежуточных данных, различных констант, таблиц перекодировок, калибровочных коэффициентов и т.п. EEPROM также может быть загружена извне как через SPI интерфейс, так и с помощью обычного программатора.

Число циклов перезаписи - не менее 100000. Два программируемых бита секретности позволяют защитить память программ и энергонезависимую память данных EEPROM от несанкционированного считывания. Имеется внутренняя оперативная память SRAM.

Порты ввода/вывода AVR имеют число независимых линий "Вход/Выход" от 3 до 53. Каждый разряд порта может быть запрограммирован на ввод или на вывод информации. Мощные выходные драйверы обеспечивают токовую нагрузочную способность 20 мА на линию порта (втекающий ток) при максимальном значении 40 мА, что позволяет, например, непосредственно подключать к микроконтроллеру светодиоды и биполярные транзисторы. Общая токовая нагрузка на все линии одного порта не должна превышать 80 мА (все значения приведены для напряжения питания 5 В). Интересная архитектурная особенность построения портов ввода/вывода у AVR заключается в том, что для каждого физического вывода существует 3 бита контроля/управления, а не 2, как у распространенных 8-разрядных микроконтроллеров (Intel, Microchip, Motorola и т.д.). Упрощенная структурная схема элемента ввода/вывода AVR - микроконтроллера приведена на рис. 1. Здесь DDRx - бит контроля направления передачи данных и привязки вывода к шине питания (VCC), PORTx - бит привязки вывода к VCC и бит выходных данных, PINx - бит для отображения логического уровня сигнала на физическом выводе микросхемы. Использование только двух бит контроля/управления порождает ряд проблем при операциях типа "чтение-модификация-запись". Например, если имеют место две последовательные операции "чтение-модификация-запись", то первый результат может быть потерян безвозвратно, если вывод порта работает на емкостную нагрузку и требуется некоторое время для стабилизации уровня сигнала на внешнем выводе микросхемы. Архитектура построения портов ввода/вывода AVR с тремя битами контроля/управления позволяет разработчику полностью контролировать процесс ввода/вывода. Если необходимо получить реальное значение сигнала на физическом выводе микроконтроллера - читайте содержимое бита по адресу PINx. Если требуется обновить выходы - прочитайте PORTx защелку и потом модифицируйте данные. Это позволяет избежать необходимости иметь

копию содержимого порта в памяти для безопасности и повышает скорость работы микроконтроллера при работе с внешними устройствами. Особую значимость приобретает данная возможность AVR для реализации систем, работающих в условиях внешних электрических помех.

2.4.Прерывания

Система прерываний - одна из важнейших частей микроконтроллера. Все микроконтроллеры AVR имеют многоуровневую систему прерываний. Прерывание прекращает нормальный ход программы для выполнения приоритетной задачи, определяемой внутренним или внешним событием.

Для каждого такого события разрабатывается отдельная программа, которую называют подпрограммой обработки запроса на прерывание (для краткости - подпрограммой прерывания), и размещается в памяти программ.

При возникновении события, вызывающего прерывание, микроконтроллер сохраняет содержимое счетчика команд, прерывает выполнение центральным процессором текущей программы и переходит к выполнению подпрограммы обработки прерывания.

После выполнения подпрограммы прерывания осуществляется восстановление предварительно сохраненного счетчика команд и процессор возвращается к выполнению прерванной программы.

Для каждого события может быть установлен приоритет. Понятие приоритет означает, что выполняемая подпрограмма прерывания может быть прервана другим событием только при условии, что оно имеет более высокий приоритет, чем текущее. В противном случае центральный процессор перейдет к обработке нового события только после окончания обработки предыдущего.

2.5.Таймеры/счетчики

Микроконтроллеры AVR имеют в своем составе от 1 до 4 таймеров/счетчиков с разрядностью 8 или 16 бит, которые могут работать и как таймеры от внутреннего источника тактовой частоты, и как счетчики внешних событий.

Их можно использовать для точного формирования временных интервалов, подсчета импульсов на выводах микроконтроллера, формирования последовательности импульсов, тактирования приемопередатчика последовательного канала связи. В режиме ШИМ (PWM)

таймер/счетчик может представлять собой широтно-импульсный модулятор и используется для генерирования сигнала с программируемыми частотой и скважностью. Таймеры/счетчики способны вырабатывать запросы прерываний, переключая процессор на их обслуживание по событиям и освобождая его от необходимости периодического опроса состояния таймеров. Поскольку основное применение микроконтроллеры находят в системах реального времени, таймеры/счетчики являются одним из наиболее важных элементов.

2.6..Сторожевой таймер

Сторожевой таймер (WatchDog Timer) предназначен для предотвращения катастрофических последствий от случайных сбоев программы. Он имеет свой собственный RC-генератор, работающий на частоте 1 МГц. Как и для основного внутреннего RC-генератора, значение 1 МГц является приближенным и зависит прежде всего от величины напряжения питания микроконтроллера и от температуры.

Идея использования сторожевого таймера предельно проста и состоит в регулярном его сбрасывании под управлением программы или внешнего воздействия до того, как закончится его выдержка времени и не произойдет сброс процессора. Если программа работает нормально, то команда сброса сторожевого таймера должна регулярно выполняться, предохраняя процессор от сброса. Если же микропроцессор случайно вышел за пределы программы (например, от сильной помехи по цепи питания) либо зациклился на каком-либо участке программы, команда сброса сторожевого таймера скорее всего не будет выполнена в течение достаточного времени и произойдет полный сброс процессора, инициализирующий все регистры и приводящий систему в рабочее состояние.

2.7.Аналоговый компаратор

Аналоговый компаратор (Analog Comparator) сравнивает напряжения на двух выводах (пинах) микроконтроллера. Результатом сравнения будет логическое значение, которое может быть прочитано из программы.

Выход аналогового компаратора можно включить на прерывание от аналогового компаратора. Пользователь может установить срабатывание прерывания по нарастающему или спадающему фронту или по переключению.

Присутствует у всех современных AVR, кроме Mega8515

2.8. Аналого-цифровой преобразователь

(АЦП) построен по классической схеме последовательных приближений с устройством выборки/хранения (УВХ). Каждый из аналоговых входов может быть соединен со входом УВХ через аналоговый мультиплексор. Устройство выборки/хранения имеет свой собственный усилитель, гарантирующий, что измеряемый аналоговый сигнал будет стабильным в течение всего времени преобразования. Разрядность АЦП составляет 10 бит при нормируемой погрешности +/- 2 разряда. АЦП может работать в двух режимах - однократное преобразование по любому выбранному каналу и последовательный циклический опрос всех каналов. Время преобразования выбирается программно с помощью установки коэффициента деления частоты специального предделителя, входящего в состав блока АЦП. Оно составляет 70...280 мкс для ATmega103 и 65...260 мкс для всех остальных микроконтроллеров, имеющих в своем составе АЦП. Важной особенностью аналого-цифрового преобразователя является функция подавления шума при преобразовании. Пользователь имеет возможность, выполнив короткий ряд программных операций, запустить АЦП в то время, когда центральный процессор находится в одном из режимов пониженного энергопотребления. При этом на точность преобразования не будут оказывать влияние помехи, возникающие при работе процессорного ядра.

2.9. Универсальный последовательный приемопередатчик

Универсальный асинхронный или универсальный синхронно/асинхронный приемопередатчик (Universal Synchronous/Asynchronous Receiver and Transmitter - UART или USART) - удобный и простой последовательный интерфейс для организации информационного канала обмена микроконтроллера с внешним миром. Способен работать в дуплексном режиме (одновременная передача и прием данных).

Он поддерживает протокол стандарта RS-232, что обеспечивает возможность организации связи с персональным компьютером. (Для стыковки МК и компьютера обязательно понадобится схема сопряжения уровней сигналов. Для этого существуют специальные микросхемы, например MAX232.)

2.10. Последовательный периферийный интерфейс SPI

Последовательный периферийный трехпроводный интерфейс SPI (Serial Peripheral Interface) предназначен для организации обмена данными между двумя устройствами. С его помощью может осуществляться обмен данными между микроконтроллером и различными устройствами, такими, как цифровые потенциометры, ЦАП/АЦП, FLASH-ПЗУ и др. С помощью этого интерфейса удобно производить обмен данными между несколькими микроконтроллерами AVR.

Кроме того, через интерфейс SPI может осуществляться программирование микроконтроллера.

2.11. Двухпроводной последовательный интерфейс TWI

Двухпроводной последовательный интерфейс TWI (Two-wire Serial Interface) является полным аналогом базовой версии интерфейса I2C (двухпроводная двунаправленная шина) фирмы Philips. Этот интерфейс позволяет объединить вместе до 128 различных устройств с помощью двунаправленной шины, состоящей из линии тактового сигнала (SCL) и линии данных (SDA).

2.12. Интерфейс JTAG

Интерфейс JTAG был разработан группой ведущих специалистов по проблемам тестирования электронных компонентов (Joint Test Action Group) и был зарегистрирован в качестве промышленного стандарта IEEE Std 1149.1-1990. Четырехпроводной интерфейс JTAG используется для тестирования печатных плат, внутрисхемной отладки, программирования микроконтроллеров.

Многие микроконтроллеры семейства Mega имеют совместимый с IEEE Std 1149.1 интерфейс JTAG или debugWIRE для встроенной отладки. Кроме того, все микроконтроллеры Mega с флэш-памятью емкостью 16 кбайт и более могут программироваться через интерфейс JTAG.

2.13 Внутренний тактовый генератор

Тактовый генератор вырабатывает импульсы для синхронизации работы всех узлов микроконтроллера. Внутренний тактовый генератор AVR может запускаться от нескольких источников опорной частоты (внешний генератор, внешний кварцевый резонатор, внутренняя или внешняя RC-цепочка). Минимальная допустимая частота ничем не ограничена (вплоть до пошагового режима).

Максимальная рабочая частота определяется конкретным типом микроконтроллера и указывается Atmel в его характеристиках, хотя практически любой AVR-микроконтроллер с заявленной рабочей частотой, например, в 10 МГц при комнатной температуре легко может быть "разогнан" до 12 МГц и выше.

Файл регистров быстрого доступа, содержит 32 8-разрядных рабочих регистра общего назначения связанных непосредственно с ALU. За один тактовый цикл из файла регистров выбираются два операнда, выполняется операция и результат вновь возвращается в файл регистров. Шесть из 32 регистров могут быть использованы как три 16-разрядных регистра указателя косвенной адресации адресного пространства данных, обеспечивающие эффективное вычисление адресов. Один из этих указателей адреса используется, также, как указатель адреса для функции непрерывного просмотра таблиц. Эти 16-разрядные дополнительные регистры обозначаются X-регистр, Y-регистр и Z-регистр.

ALU поддерживает арифметические и логические операции между регистрами или между константой и регистром. Выполняются в ALU и операции с отдельными регистрами.

В дополнение к операциям с регистрами, регистровый файл может использоваться и для обычной адресации памяти. Это объясняется тем, что файл регистров располагается по 32 самыми младшими адресами пространства данных, и к ним можно обращаться как к обычным ячейкам памяти. Пространство памяти I/O содержит 64 адреса периферийных функций CPU таких как: регистры управления, таймеры/счетчики, аналого-цифровые преобразователи и другие I/O функции. К памяти I/O можно обращаться непосредственно или как к ячейкам пространства памяти соответствующим адресам регистра файлов \$20 - \$5F. В микроконтроллерах AVR использованы принципы Гарвардской архитектуры - отдельные память и шины для программ и данных. При работе с памятью программ используется одноуровневый конвейер - в то время, как одна команда выполняется, следующая команда выбирается из памяти программ, Такой прием позволяет выполнять команду в каждом тактовом цикле. Памятью

программ является внутрисистемно программируемая Flash память. За малым исключением AVR команды имеют формат одного 16-разрядного слова, в связи с чем каждый адрес памяти программ содержит одну 16-разрядную команду. В процессе обработки прерываний и вызовов подпрограмм адрес возврата счетчика команд (PC) сохраняется в стеке. Стек размещается в SRAM данных и, следовательно размер стека ограничен только общим размером SRAM и уровнем ее использования. Все пользовательские программы в подпрограммах возврата (прежде, чем подпрограммы или прерывания будут выполняться) должны инициализировать указатель стека (SP). 16-разрядный указатель стека, с возможностью чтения/записи располагается в пространстве I/O. AVR архитектура поддерживает пять различных режимов адресации 4000 байт SRAM данных. Гибкий модуль обработки прерываний имеет в пространстве I/O свой управляющий регистр с дополнительным битом разрешения глобального прерывания в регистре статуса. Все прерывания имеют свои векторы прерывания в таблице векторов прерывания, располагаемой в начале памяти программ. Приоритеты прерываний соответствуют положению векторов прерываний - прерывание с наименьшим адресом вектора имеет наивысший приоритет. По разнообразию и количеству реализованных инструкций AVR больше похожи на CISC, чем на RISC процессоры. Например, у PIC-контроллеров система команд насчитывает до 75 различных инструкций, а у MCS51 она составляет 111. В целом, прогрессивная RISC архитектура AVR в сочетании с наличием регистрового файла и расширенной системы команд позволяет в короткие сроки создавать работоспособные программы с эффективным кодом как по компактности реализации, так и по скорости выполнения.

У AVR-контроллеров "с рождения" есть две особенности, которые отличают это семейство от остальных МК. Во-первых, система команд и архитектура ядра AVR разрабатывались совместно с фирмой-разработчиком компиляторов с языков программирования высокого уровня IAR Systems. В результате появилась возможность писать AVR-программы на языке C без большой потери в производительности по сравнению с программами, написанными на языке ассемблера. Подробнее этот вопрос мы обсудим в главе 5.

Во-вторых, одним из существенных преимуществ AVR стало применение конвейера. В результате для AVR не существует понятия машинного цикла: большинство команд выполняется за один такт. Для сравнения отметим, что пользующиеся большой популярностью МК семейства PIC выполняют команду за 4 такта, а классические 8051 — вообще за 12 тактов (хотя есть и современные модели x51 с машинным циклом в один такт).

Правда, при этом пришлось немного пожертвовать простотой системы команд, особенно заметной в сравнении с x51, где, например, любые

операции пересылки данных внутри контроллера, независимо от способа адресации, выполняются единственной командой `mov` в различных вариантах, в то время как в AVR почти для каждого способа своя команда, к тому же иногда с ограниченной областью действия. Есть некоторые сложности и в области операций с битами. Тем не менее это не приводит к заметным трудностям при изучении AVR-ассемблера: наоборот, тексты программ получаются короче и больше напоминают программу на языке высокого уровня. Следует также учесть, что из общего числа команд от 90 до 130, в зависимости от модели, только 50-60 уникальных, остальные взаимозаменяемые. И, наконец, этот недостаток полностью нивелируется при использовании языка C, фактически уравнивающего разные архитектуры с точки зрения особенностей программирования.

3. Микроконтроллеры AVR

Огромное преимущество AVR-архитектуры — наличие 32 оперативных регистров, не совсем равноправных, но позволяющих в ряде случаев вообще не обращаться к оперативной памяти и не использовать стек (что в принципе невозможно в том же семействе x51), более того, в младших моделях AVR стек вообще недоступен для программиста. Потому структура ассемблерных программ для AVR стала подозрительно напоминать программы на языке высокого уровня, где операторы работают не с ячейками памяти и регистрами, а с абстрактными переменными и константами.

Еще одна особенность AVR со схемотехнической точки зрения — все выводы в них могут пребывать в трех состояниях (вход — отключено — выход) и электрически представляют собой КМОП-структуры (т. е. имеет место симметрия выходных сигналов и высокое сопротивление для входных). В общем случае это значительно удобнее портов X51 (двустабильных и TTL-совместимых) и предполагает лучшую помехозащищенность (по крайней мере, от помех по шине "земли").

Суммировав мнения из различных источников и опираясь на собственный опыт, автор пришел примерно к такому подразделению областей применения трех самых распространенных семейств контроллеров.

Контроллеры классической архитектуры x51 (первые микросхемы семейства 8051 были выпущены еще в начале 1980-х) лучше всего подходят для общего изучения предмета. Отметим, что кроме Atmel, x51-совместимые изделия выпускают еще порядка десятка фирм, включая такие гиганты, как Philips и Siemens, есть и отечественные аналоги (серии 1816, 1830 и др.), что делает эту архитектуру наиболее универсальной.

Семейство AVR рекомендуется для начинающих электронщиков-практиков, в силу простоты и универсальности устройства, преемственности структуры для различных типов контроллеров, простоты схемотехники и программирования (в данном случае под "программированием" понимается процесс записи программ в микросхему).

PIC фирмы Microchip идеально подходят для проектирования несложных устройств, особенно предназначенных для тиражирования.

Эта классификация во многом субъективна, и автор не будет оспаривать другие точки зрения: различные семейства МК постепенно сближаются по параметрам, становятся полностью взаимозаменяемыми и, как и во всей современной электронике, выбор того или иного семейства часто носит характер "религиозного".

К тому же три упомянутых семейства МК — лишь наиболее распространенные среди универсальных контроллеров, но далеко не самые массовые вообще. Общее количество существующих семейств микроконтроллеров оценивается приблизительно в 100 с лишним, причем ежегодно появляются все новые и новые. Каждое из этих семейств может включать десятки разных моделей. При этом первое место среди производителей 8-разрядных МК традиционно принадлежит фирме Motorola, в основном за счет контроллеров для мобильных устройств. Компания Microchip со своим семейством PIC занимает третье место, а Atmel — лишь шестое. При этом, кроме 8-разрядных МК AVR, Atmel выпускает еще несколько разновидностей МК, к которым относятся не только упомянутые наследники 8051, но и ARM-процессоры и специализированные МК для различных применений. Тем не менее эта формальная статистика еще ни о чем не говорит — так, среди МК со встроенной flash-памятью Atmel принадлежит уже треть мирового рынка.

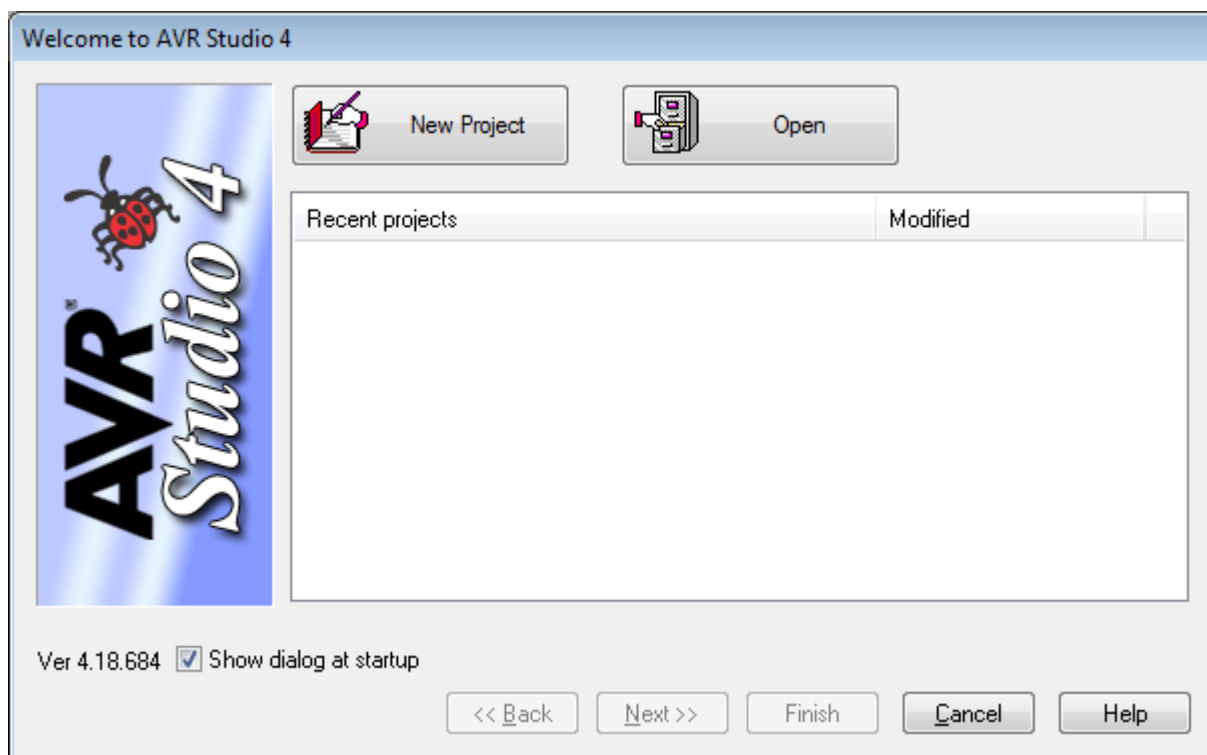
4. AVR Studio. Создание первого проекта

AVR studio программа предназначенная для разработки и отладки программ для микроконтроллеров AVR.

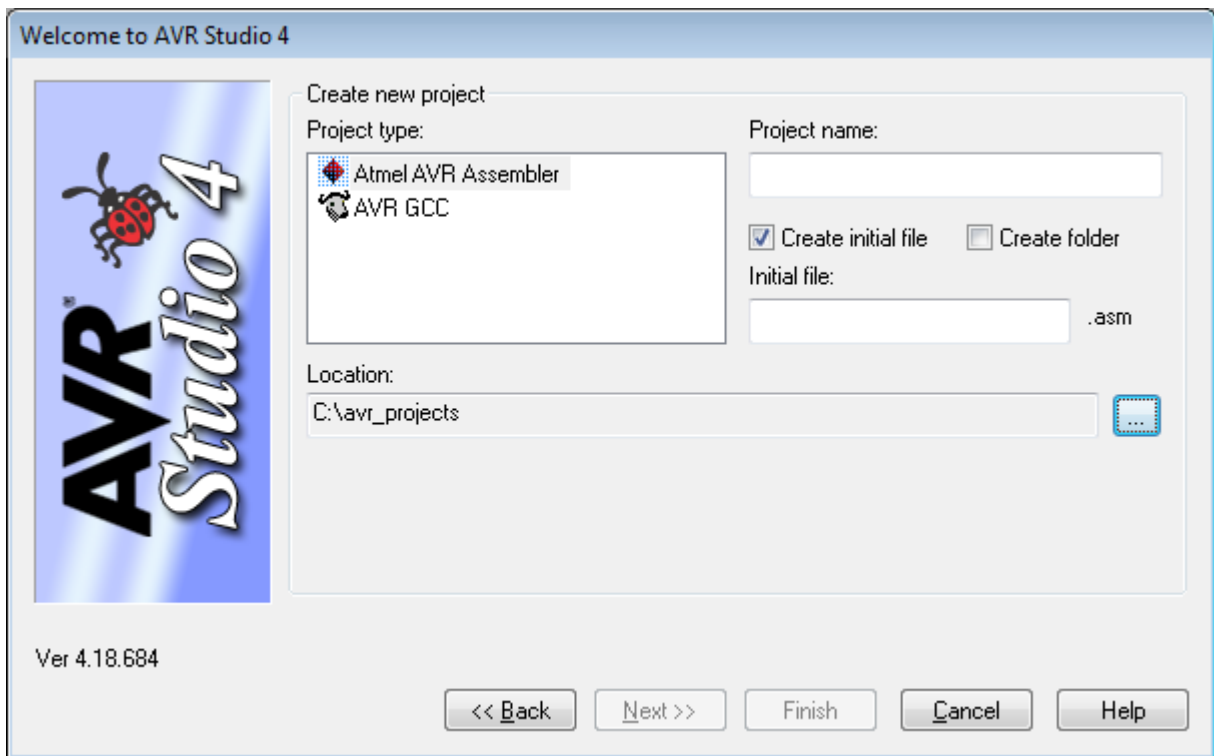
Создайте на любом диске каталог. Имя каталога маленькими буквами на английском языке, например avr_project.

Установите программу на ПК и запустите.

Вы увидите приветственное сообщение:

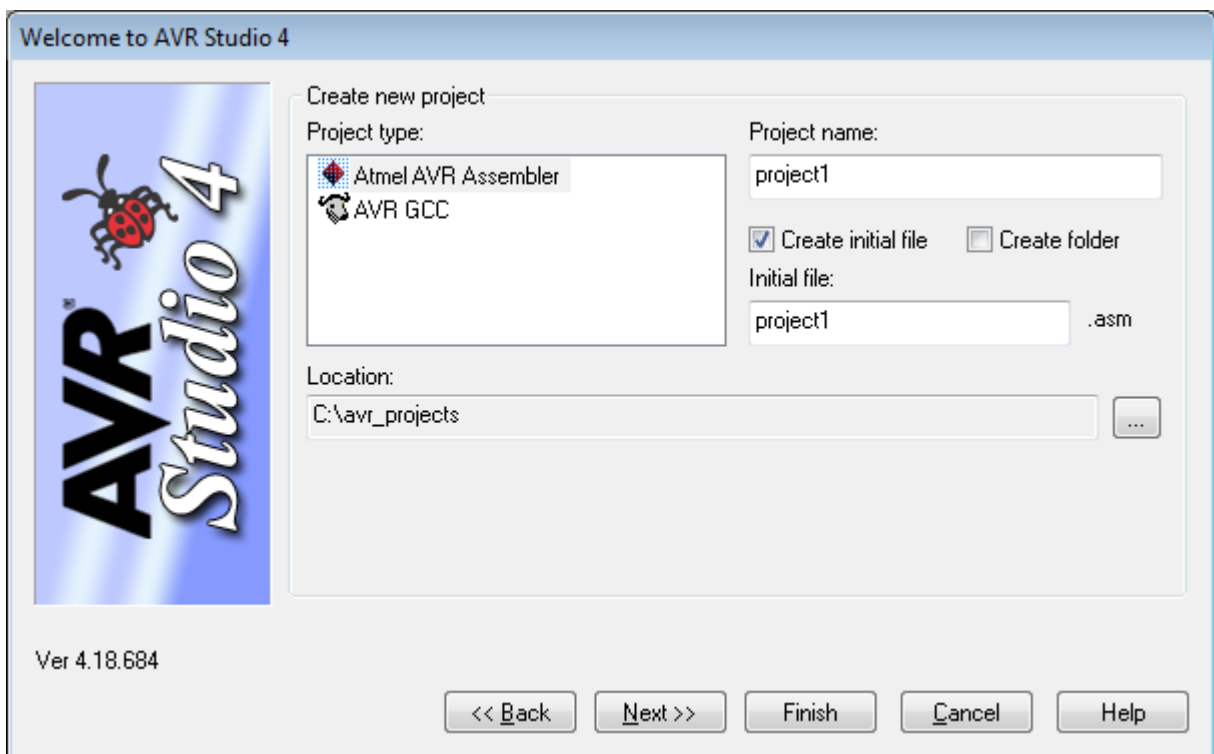


Выберите New Project

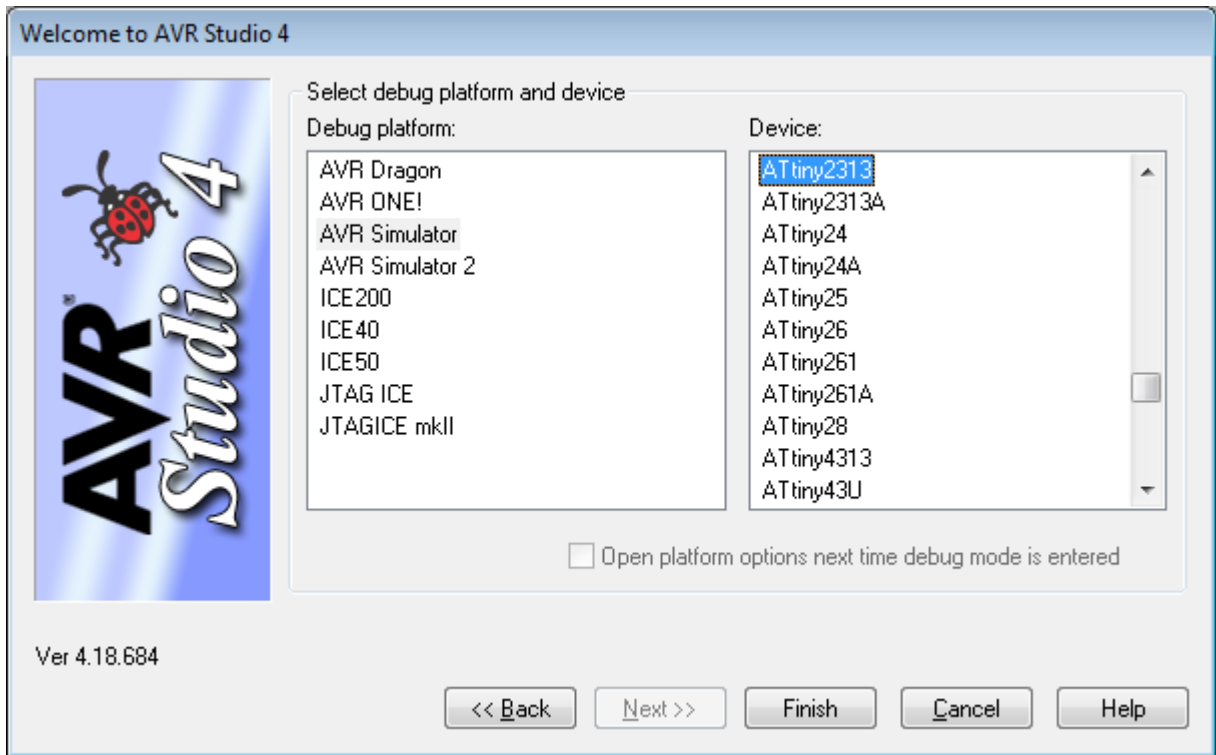


Выберите Atmel AVR Assembler, справа от поля Location нажмите на кнопку «...» и укажите ваш каталог кнопкой Select.

В поле Project name введите имя своего проекта.



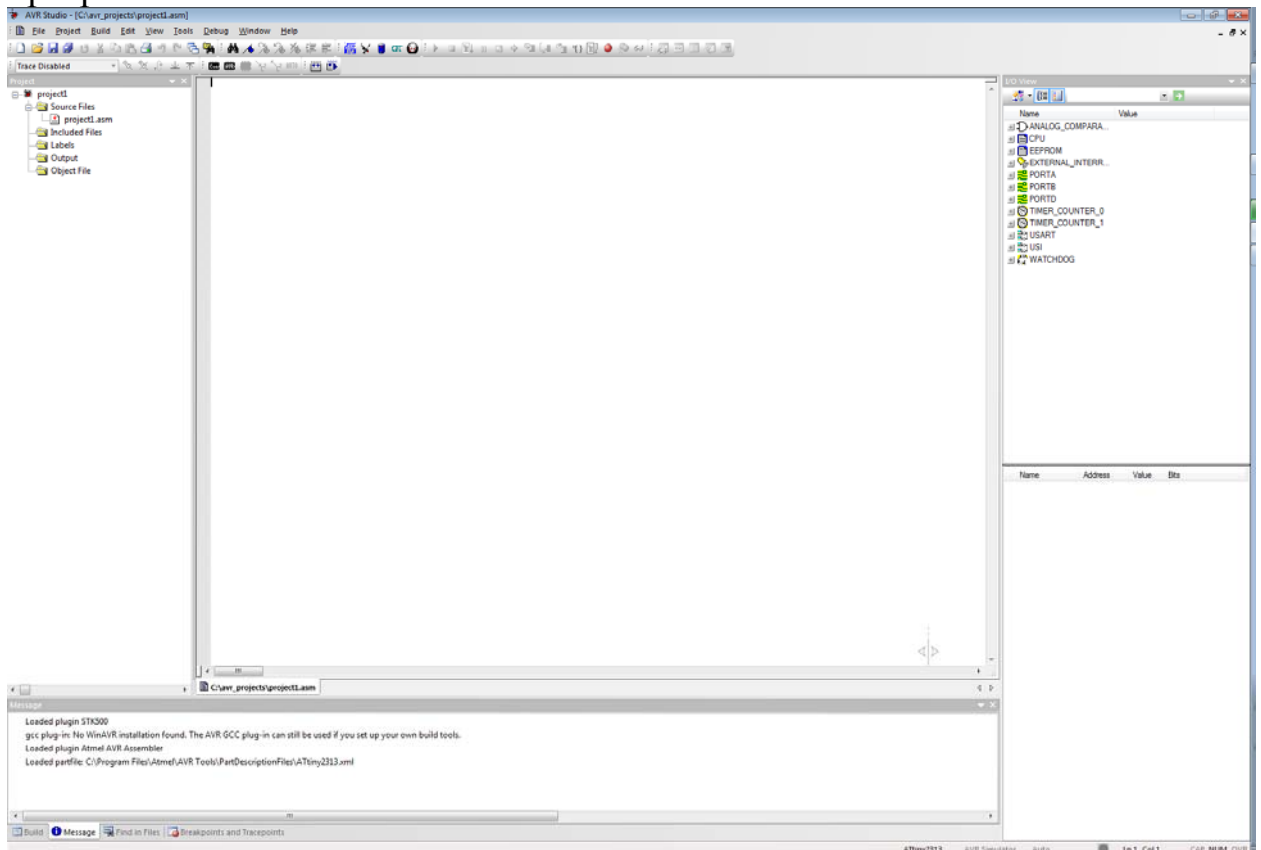
Нажмите кнопку Next



Слева выберите AVR simulator, а справа тип микроконтроллера Atiny 2313.

После этого нажмите Finish

На этом создание проекта завершено, можно приступать к написанию программ.



Интерфейс программы состоит из нескольких окон.

Слева – файлы входящие в проект.

По центру – окно ввода программ.

Внизу – отладочная информация.

5. Создание программы

Что бы научиться работать с программой, создадим нашу первую программу и отдадим её.

Листинг программы

```
.def a=r16
.def b=r17
.org $000
rjmp program1
program1:

ldi a,10
ldi b,1
add a,b
sub b,a
```

Разберем подробно каждую команду:

```
.def a=r16
```

Сопоставим переменную a регистру R16

```
.def b=r17
```

Сопоставим переменную b регистру R17

```
.org $000
```

Перехватываем старт микроконтроллера.

```
rjmp program1
```

После перехвата передаем управление программе program1.

```
program1:
```

Начало программы (всегда заканчивается двоеточием).

```
ldi a,10
```

Присвоим переменной a значение 10.

```
ldi b,1
```

Присвоим переменной b значение 1.

```
add a,b
```

Сложим переменные a и b, результат сложения поместим в a.

```
sub b,a
```

Сложим переменные b, a. Результат сложения поместим в b.

Прежде чем приступить к отладке программы, выполним расчеты вручную. В результате выполнения двух последних операций мы должны получить в регистрах R16 – 11, а в регистре R17 -10.

Запустим компиляцию нажав на кнопку F7. В статусном окне вы должны увидеть результаты и найденные ошибки. Если ошибок нет, появится сообщение 0 errors, 0 warnings.

Если компиляция прошла успешно, нажмите на кнопку Play (синий треугольник) и слева разверните раздел Registers 16-32.

Отладка программы осуществляется последовательным нажатием кнопки F11.

Пронаблюдайте над значениями, которые заносятся в регистры и проверьте правильность работы программы. Не забывайте, что числа в регистрах отображаются в шестнадцатеричной системе счисления.

6. Команды перехода

В отличие от высокоуровневых языков, таких как C++ или Delphi, в языке ассемблера нет переходов по условиям «Если». Для этого применяется так называемый статусный регистр SREG.

Изменим нашу первую программу следующим образом:

```
.def a=r16
.def b=r17
.org $000
rjmp program1
program1:

ldi a,1
ldi b,3
cp a,b (команда сравнения переменных a,b)
```

Произведем компиляцию и запустим программу. В окне I/O view откроем вкладку I/O Atiny 2313 и затем вкладку CPU. Разверните статусный регистр и проведите отладку с помощью кнопки F11. Запишите флаги, которые будут установлены.

Теперь повторите ту же операцию для тех случаев, когда переменные будут равны или когда переменная a будет больше чем переменная b. Запишите результаты.

Не сложно заметить, что в каждом случае появление флагов обусловлено тем или иным результатом сравнения.

Напишем программу, которая учитывает установку флагов:

```
def a=r16
.def b=r17
.org $000
rjmp program1
program1:

ldi a,10
ldi b,1
cp a,b
brcs proc2
ldi a,5
rjmp program1
proc2:
```

```
ldi a,6  
rjmp program1
```

В этой программе, в случае когда a больше чем b , в переменную a будет всегда заноситься значение 5. Если a будет меньше чем b , то соответственно в переменную a будет заноситься значение 6.

Переход осуществляется с помощью команды `brcs`, которая срабатывает, когда устанавливается флаг C .

Самостоятельно напишите программу по сортировке 3х чисел по возрастанию.

7. ЗАДАНИЕ НА КОНТРОЛЬНУЮ РАБОТУ

Вариант выдается на установочной лекции.

Требуется: разработать полную электрическую схему (выбрать источник питания, кварцевый резонатор, семисегментный индикатор и другие элементы схемы) с использованием микроконтроллера Atmel ATmega 163 и написать программу для разработанного устройства.

Разработанное устройство должно выполнять следующую задачу:

Устройство контроля проверяет наличие аварийного сигнала на аналоговом датчике аварии (датчике нулевой последовательности) (задействовать 2 пина). Если аварийный сигнал выше 3В, это означает наличие аварии. Максимальное значение – 5В.

После того, как получен аварийный сигнал с датчика, проводится повторная проверка, через время указанное в таблице задания. Если спустя указанное время, сигнал с датчика аварии поступает, то нужно произвести опрос указанных в задании присоединений. Вывести на экран семисегментного индикатора номер поврежденного кабеля и единицу на выход подключенный к системе звуковой сигнализации.

Отключить звуковую сигнализацию можно введя код разблокировки, код набирается с помощью простой матричной клавиатуры 3x3. Если код введен правильно – отключить звуковую индикацию, на экран вывести 8, перейти на начало программы.

Если аналоговый сигнал больше 1В и меньше 3В – на экране семисегментного индикатора должна гореть черточка.

АЦП реализовать по своему выбору (либо на аналоговой базе), либо с помощью микроконтроллера.

Если обрыва нет, то на индикаторе должен отображаться 0. Схему выполнить на листах формата А4 или А3.

№	№ используемых присоединений	Время ожидания (секунды)	Код разблокировки
1	1,2,3,4,5	1	3523
2	1,2,3,4,7	2	1523
3	2,3,4,5,6	3	5512
4	2,4,5,6,7	4	1252
5	3,4,5,6,7	5	7912
6	1,3,5,6,7	6	6154

7	2,3,4,6,9	7	1313
8	5,6,7,8,9	8	1597
9	2,3,5,6,9	9	5122
10	4,5,6,7,8	10	1876
11	1,2,5,6,7	11	1125
12	1,3,4,8,9	12	1136
13	1,2,5,6,9	13	7561
14	2,4,5,6,9	14	1179
15	1,5,6,8,9	15	5564
16	3,5,7,8,9	0.1	1378
17	1,2,3,8,9	0.2	1222
18	1,2,3,5,7	0.3	1976
19	1,3,5,6,7	0.4	4312
20	1,2,3,4,7	0.5	4972
21	1,3,5,6,8	0.6	4697
22	1,2,3,6,7	0.7	9795
23	1,3,4,7,8	0.8	1975
24	1,3,4,8,9	0.9	4657
25	1,2,5,7,8	0.15	4692
26	1,5,6,8,9	0.11	4972
27	3,4,5,8,9	0.16	2271
28	3,4,5,7,9	0.12	1975
29	2,4,5,8,9	0.14	6657
30	2,3,4,6,9	0.11	1359

Если номер больше указанного в таблице, выбирается вариант соотв. порядковому номеру с вычитанием 30 из него.

Микроконтроллер работает с частотой 4мгц. Для упрощения написания кода, каждой команде соответствует 2 такта. В качестве преобразователя аналогового сигнала можно использовать схему на усилителях или дополнительный микроконтроллер с АЦП.

Программа должна быть написана только на языке Assembler специфики AVR.

Содержание проекта:

1. Введение
2. Выбор и обоснование выбора микроконтроллера
3. Разработка источника питания микроконтроллера
4. Выбор семисегментного индикатора
5. Выбор АЦП
6. Алгоритмы работы схемы
7. Алгоритмы работы программы

8. Текст программы
9. Заключение
10. Приложение (чертежи)

11.

Требования к выполнению контрольной работы.

Контрольная работа выполняется в печатном виде на листах А4 согласно требованиям ВУЗа. Для успешной сдачи работы, допускается не более 15% ошибок в листинге программы в каждом из заданий.

Работа сдается на проверку до сессии, назначается время защиты, если работа проверена и дан допуск к защите.

Список учебников:

1. Голубцов М.С. Микроконтроллеры AVR от простого к сложному. Год издания: 2003 Солон-Пресс , с. 288
2. Белов А. В. Конструирование устройств на микроконтроллерах.
3. Белов А. В. Микроконтроллеры AVR в радиолюбительской практике.
4. Гребнев В. В. Микроконтроллеры семейства AVR фирмы Atmel.
5. Евстифеев А. В. Микроконтроллеры AVR семейств Tiny и Mega фирмы Atmel.