

Направление подготовки: 140400 Электроэнергетика и электротехника

Профиль подготовки: Электрический транспорт

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Дисциплина:

"Информационные и компьютерные технологии в электротехнике"

Часть 2. «Микроконтроллерные информационно-управляющие устройства и системы»

Практические занятия по курсу «Информационные и компьютерные технологии в электротехнике и на транспорте».

Краткие теоретические сведения

Введение

Создание любой микроконтроллерной системы (МКС) требует не только разработки аппаратных и программных средств, но и их проверки и отладки. Существует два способа разработки, проверки и отладки МКС:

1) По старинке – на реальном микроконтроллере. При этом необходимо разработать и изготовить плату МКС, составить программу, откомпилировать, запрограммировать (прошить) программатором микроконтроллер, и дальше, вооружившись измерительными приборами и осциллографом начать изучать работу устройства по внешним выводам микроконтроллера, и это только в том случае, если он заработает. Если удастся запустить в работу МКС и найти ошибку, то необходимо исправить или написать новую программу, откомпилировать её, прошить заново МК и так до тех пор, пока МКС не заработает без ошибок (или количество циклов перезаписи Flash не закончится). Это аппаратный способ отладки. Он требует специальной измерительной аппаратуры, специалистов высокой квалификации, больших ресурсов времени, и сопряжен со многими проблемами.

2) Разработка и отладка МКС виртуально с помощью специальных пакетов программных средств разработки, отладки и моделирования, так называемых систем IDE (Integrated Development Environment). При этом составление программы, проверка ее на синтаксические ошибки, компиляция, разработка схемы и моделирование МКС, поиск ошибок в алгоритме программы выполняется виртуально на компьютере. Это программный способ разработки МКС. Он имеет много преимуществ: позволяет выполнять симуляцию (моделирование) работы устройства в пошаговом режиме; просматривать содержание любых внутренних регистров МК: регистров общего назначения, регистров ввода/вывода, состояния портов, периферийных устройств и т.д; видеть работу всей МКС в динамике, просматривать диаграммы сигналов в любой точке схемы и измерять временные параметры этих сигналов. В случае обнаружения ошибок алгоритма симулятор в специальном окне может указать тип ошибки и рекомендации на ее устранение.

В рамках данного курса мы будем изучать разработку и программирование МКС на примере микроконтроллеров линейки AVR компании Atmel. Почему именно этой линейки?

Во-первых, микроконтроллеры Atmel® AVR® очень просты в использовании. Они повсеместно доступны, выпускаются в больших количествах и поэтому цена их невысока.

Во-вторых, они имеют высокоэффективную современную архитектуру, высокое быстродействие, сочетание простоты и универсальности, надёжность, большое число модификаций для широкого спектра применения, начиная от простейших встроенных устройств бытовой техники и кончая системами управления сложными объектами в промышленности и на транспорте.

В-третьих, из-за наличия многочисленного программного обеспечения для их программирования и прошивки, в том числе свободно распространяемого по лицензии GNU. То есть, при написании программ вы не будете нарушать ни чьих авторских прав.

В качестве инструмента разработки и программирования будем использовать IDE Atmel Studio, а в качестве инструмента моделирования и конструирования IDE Proteus. Совокупность этих двух систем автоматизированной разработки (САПР) позволяет полностью обеспечить современный цикл высокоэффективного создания системы от идеи до серийного производства.

Программный пакет Atmel Studio (AS) и Microchip Studio (MS)

Интегрированная среда разработки (IDE) AS и MS содержат все необходимое для написания, компиляции и отладки кода и позволяют загружать его прямо в флеш-память на чипе микроконтроллера AVR без применения дополнительных программных компонентов. Эти IDE предоставляются бесплатно. Они представляют собой пакет программных средств, предназначенный для разработки, отладки и моделирования программного обеспечения, выполненным на основе AVR микроконтроллеров. Его часто называют виртуальной лабораторией.

Программный пакет “Proteus” (PS)

Proteus это IDE (интегрированный пакет программ) компании Labcenter Electronics (Великобритания) для полного цикла автоматизированной разработки электронных устройств. Пакет Proteus VSM состоит из двух взаимосвязанных частей:

ISIS – программы для создания принципиальных схем электронных устройств и их моделирования на основе библиотеки моделей электронных компонентов в формате PSPICE; ISIS позволяет достаточно реалистично моделировать микроконтроллеры AVR, 8051, PIC10, PIC16, PIC18, ARM7, Motorola HC11 и всю схему в которой они работают;

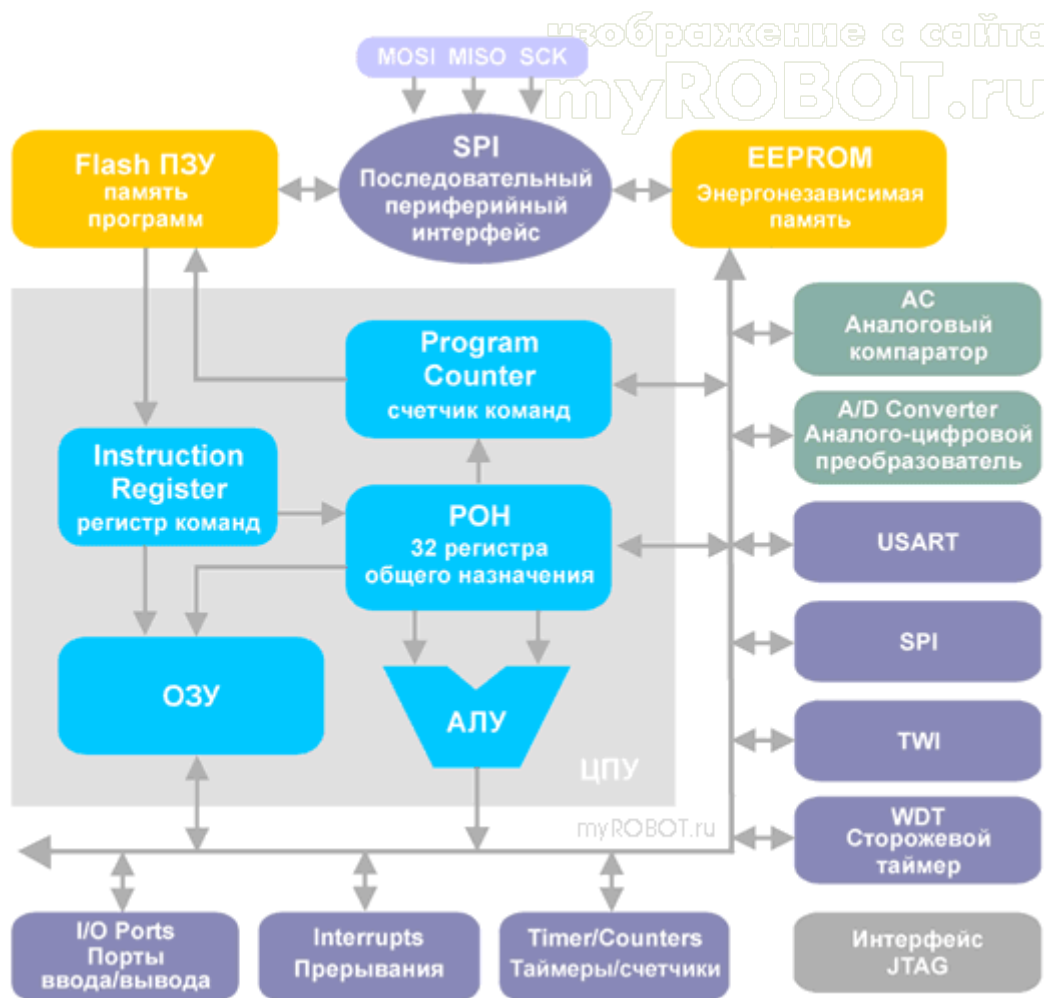
ARES – программ разработки печатных плат и твердотельного 3D-моделирования конструкции электронных устройств на основе библиотеки 3D-моделей компонентов.

В состав восьмой версии входит также среда разработки **VSM Studio**, позволяющая писать, отлаживать и компилировать программы для микроконтроллеров, используемых в проекте.

Proteus поддерживает подключение или создание новых элементов (SPICE) и различных внешних компиляторов для большого количества типов микроконтроллеров.

Знакомство с микроконтроллером AVR

Микроконтроллеры AVR содержат: быстрый RISC-процессор, два типа энергонезависимой памяти (Flash-память программ и память данных EEPROM), оперативную память RAM, порты ввода/вывода и различные периферийные интерфейсные схемы.



Сердцем микроконтроллеров AVR является 8-битное микропроцессорное ядро или центральное процессорное устройство (ЦПУ), построенное на принципах RISC-архитектуры. Основой этого блока служит арифметико-логическое устройство (АЛУ). По системному тактовому сигналу из памяти программ в соответствии с содержимым счетчика команд (Program Counter - PC) выбирается очередная команда и выполняется АЛУ. Во время выбора команды из памяти программ происходит выполнение предыдущей выбранной команды, что и позволяет достичь быстродействия 1 MIPS на 1 МГц.

АЛУ подключено к регистрам общего назначения РОН (General Purpose Registers - GPR). Регистров общего назначения всего 32, они имеют байтовый формат, то есть каждый из них состоит из восьми бит. РОН находятся в начале адресного пространства оперативной памяти, но физически не являются ее частью. Поэтому к ним можно обращаться двумя способами (как к регистрам и как к памяти). Такое решение является особенностью AVR и повышает эффективность работы и производительность микроконтроллера.

Отличие между регистрами и оперативной памятью состоит в том, что с регистрами можно производить любые операции (арифметические, логические, битовые), а в оперативную память можно лишь записывать данные из регистров.

В микроконтроллерах AVR реализована Гарвардская архитектура, в соответствии с которой разделены не только адресные пространства памяти программ и памяти данных, но и шины доступа к ним. Каждая из областей памяти данных (оперативная память и EEPROM) также расположена в своем адресном пространстве.

Память программ Flash ПЗУ (Flash ROM)

Память программ предназначена для хранения последовательности команд, управляющих функционированием микроконтроллера, и имеет 16-ти битную организацию. Все AVR имеют Flash-память программ, которая может быть различного размера - от 1 до 256 КБайт. Ее главное достоинство в том, что она построена на принципе электрической перепрограммируемости, т. е. допускает многократное стирание и запись информации. Программа заносится во Flash-память AVR как с помощью обычного программатора, так и с помощью SPI-интерфейса, в том числе непосредственно на собранной плате. Возможностью внутрисхемного программирования (функция ISP) через коммуникационный интерфейс SPI обладают все микроконтроллеры AVR, кроме Tiny11 и Tiny28.

Все микроконтроллеры семейства Mega имеют возможность самопрограммирования, т. е. самостоятельного изменения содержимого своей памяти программ. Эта особенность позволяет создавать на их основе очень гибкие системы, алгоритм работы которых будет меняться самим микроконтроллером в зависимости от каких-либо внутренних условий или внешних событий.

Гарантированное число циклов перезаписи Flash-памяти у микроконтроллеров AVR второго поколения составляет не менее 10 тыс. циклов при типовом значении 100 тыс. циклов. (В официальной технической документации Atmel Corp. указывается значение 10 тыс. циклов.)

Память данных

Память данных разделена на три части: регистровая память, оперативная память (ОЗУ - оперативное запоминающее устройство или RAM) и энергонезависимая память (ЭСППЗУ или EEPROM).

Регистровая память (РОН и РВВ)

Регистровая память включает 32 регистра общего назначения (РОН или GPR), объединенных в файл, и служебные регистры ввода/вывода (РВВ). И те и другие расположены в адресном пространстве ОЗУ, но не являются его частью. В области регистров ввода/вывода расположены различные служебные регистры (регистры управления микроконтроллером, регистры состояния и т. п.), а также регистры управления периферийными устройствами, входящими в состав микроконтроллера. По сути, управление микроконтроллером заключается в управлении этими регистрами.

Оперативная память (ОЗУ или RAM)

Внутренняя оперативная статическая память Static RAM (SRAM) имеет байтовый формат и используется для оперативного хранения данных.

Размер оперативной памяти может варьироваться у различных чипов от 64 Байт до 4 КБайт. Число циклов чтения и записи в RAM не ограничено, но при отключении питающего напряжения вся информация теряется.

Для некоторых микроконтроллеров возможна организация подключения внешнего статического ОЗУ объемом до 64К.

Энергонезависимая память данных (EEPROM)

Для длительного хранения различной информации, которая может изменяться в процессе функционирования микроконтроллерной системы, используется EEPROM-память. Все AVR имеют блок энергонезависимой электрически перезаписываемой памяти данных EEPROM от 64 Байт до 4 КБайт. Этот тип памяти, доступный программе микроконтроллера непосредственно в ходе ее выполнения, удобен для хранения промежуточных данных, различных констант, коэффициентов, серийных номеров, ключей и т.п. EEPROM может быть загружена извне как через SPI интерфейс, так и с помощью обычного программатора. Число циклов стирание/запись - не менее 100 тыс.

Периферия

Периферия микроконтроллеров AVR включает: порты (от 3 до 48 линий ввода и вывода), поддержку внешних прерываний, таймеры-счетчики, сторожевой таймер, аналоговые компараторы, 10-разрядный 8-канальный АЦП, интерфейсы UART, JTAG и SPI, устройство сброса по понижению питания, широтно-импульсные модуляторы.

Порты ввода/вывода (I/O)

Порты ввода/вывода AVR имеют число независимых линий "вход/выход" от 3 до 53. Каждая линия порта может быть запрограммирована на вход или на выход. 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).

Архитектурная особенность построения портов ввода/вывода у AVR заключается в том, что для каждого физического вывода (пина) существует 3 бита контроля/управления, а не 2, как у распространенных 8-разрядных микроконтроллеров (Intel, Microchip, Motorola и т.д.). Это позволяет избежать необходимости иметь копию содержимого порта в памяти для безопасности и повышает скорость работы микроконтроллера при работе с внешними устройствами, особенно в условиях внешних электрических помех.

Прерывания (INTERRUPTS)

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.

Таймеры/счетчики (TIMER/COUNTERS)

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.

Сторожевой таймер (WDT)

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 заиклился 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.

Аналоговый компаратор (AC)

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

Аналого-цифровой преобразователь (A/D CONVERTER)

Аналого-цифровой преобразователь (АЦП) служит для получения числового значения

напряжения, поданного на его вход. Этот результат сохраняется в регистре данных АЦП. Какой из выводов (пинов) микроконтроллера будет являться входом АЦП, определяется числом, занесенным в соответствующий регистр.

Универсальный последовательный приемопередатчик (UART или USART)

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 или RS-485, 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 или MAX485 .)

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.

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).

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.

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.

Система реального времени (RTC)

RTC реализована во всех микроконтроллерах Mega и в двух кристаллах "classic" - AT90(L)S8535. Таймер/счетчик RTC имеет отдельный предделитель, который может быть программным способом подключен или к источнику основной тактовой частоты, или к дополнительному асинхронному источнику опорной частоты (кварцевый резонатор или внешний синхросигнал). Для этой цели зарезервированы два вывода микросхемы. Внутренний осциллятор оптимизирован для работы с внешним "часовым" кварцевым резонатором 32,768 кГц.

Питание

AVR функционируют при напряжениях питания от 1,8 до 6,0 Вольт. Ток потребления в активном режиме зависит от величины напряжения питания и частоты, на которой работает микроконтроллер, и составляет менее 1 мА для 500 кГц, 5 ... 6 мА для 5 МГц и 8 ... 9 мА для частоты 12 МГц.

AVR могут быть переведены программным путем в один из трех режимов пониженного энергопотребления.

Режим холостого хода (IDLE). Прекращает работу только процессор и фиксируется содержимое памяти данных, а внутренний генератор синхросигналов, таймеры, система прерываний и сторожевой таймер продолжают функционировать. Ток потребления не превышает 2,5 мА на частоте 12 МГц.

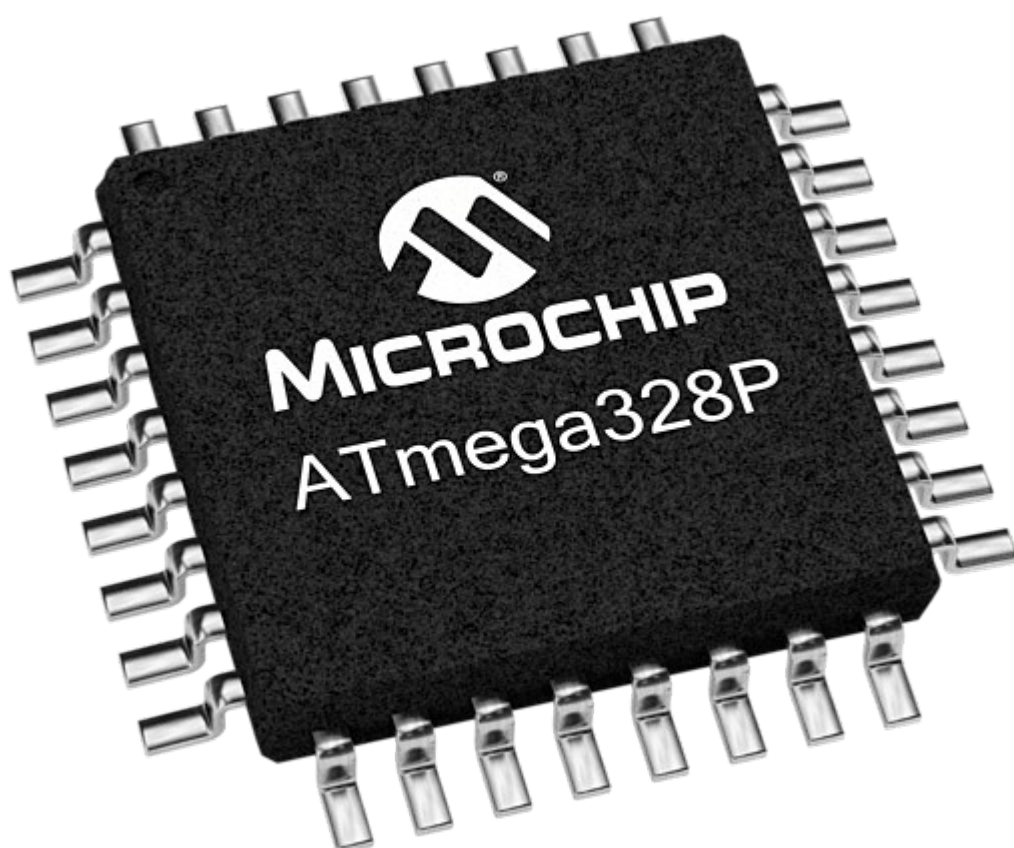
Стоповый режим (POWER DOWN). Сохраняется содержимое регистрового файла, но останавливается внутренний генератор синхросигналов, и, следовательно, останавливаются все функции, пока не поступит сигнал внешнего прерывания или аппаратного сброса. При включенном сторожевом таймере ток потребления в этом режиме составляет около 80 мкА, а при выключенном - менее 1 мкА. (Все приведенные значения справедливы для напряжения питания 5 В).

Экономичный режим (POWER SAVE). Продолжает работать только генератор таймера, что обеспечивает сохранность временной базы. Все остальные функции отключены.

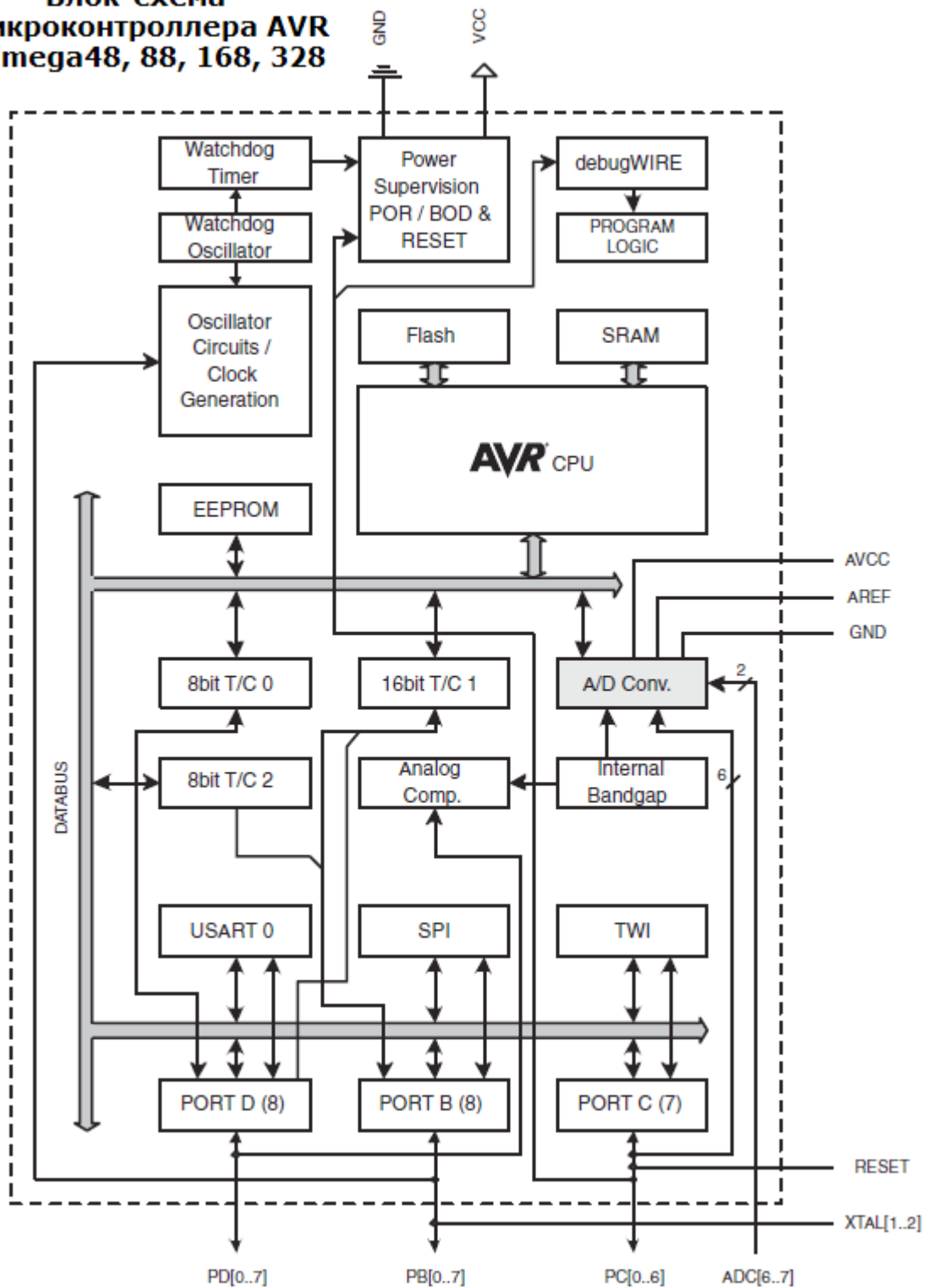
Сброс при снижении напряжения питания (BOD)

Схема BOD (Brown-Out Detection) отслеживает напряжение источника питания. Если схема включена, то при снижении питания ниже некоторого значения она переводит микроконтроллер в состояние сброса. Когда напряжение питания вновь увеличится до порогового значения, запускается таймер задержки сброса. После микроконтроллера формирования задержки внутренний сигнал сброса снимается и происходит запуск.

В дальнейшем конкретно будем рассматривать микроконтроллер Atmega 328P, как наиболее универсальный для рассматриваемого нами круга задач.

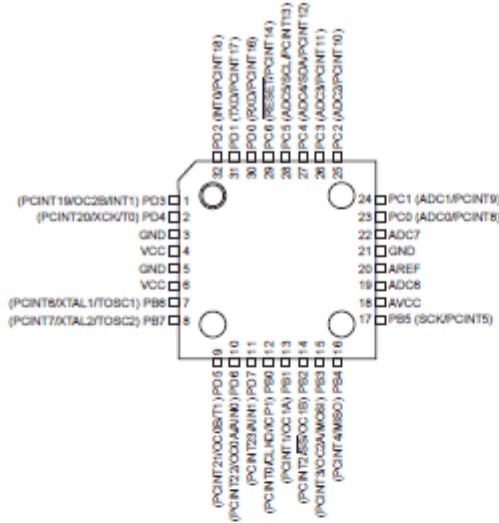


**Блок-схема
микроконтроллера AVR
ATmega48, 88, 168, 328**

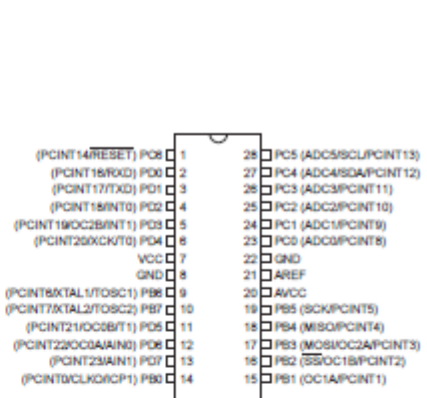


Pinout ATmega48A/PA/88A/PA/168A/PA/328/P

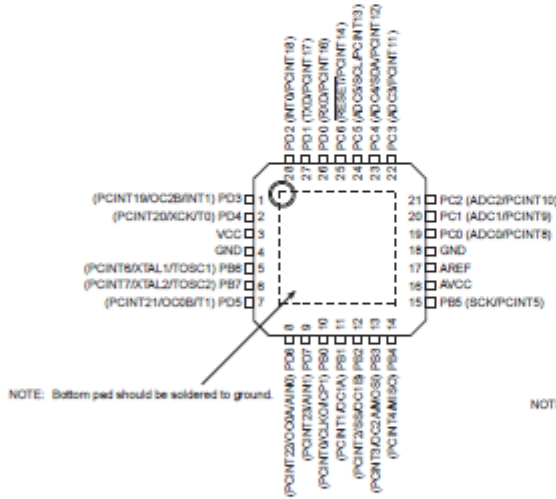
32 TQFP Top View



28 PDIP

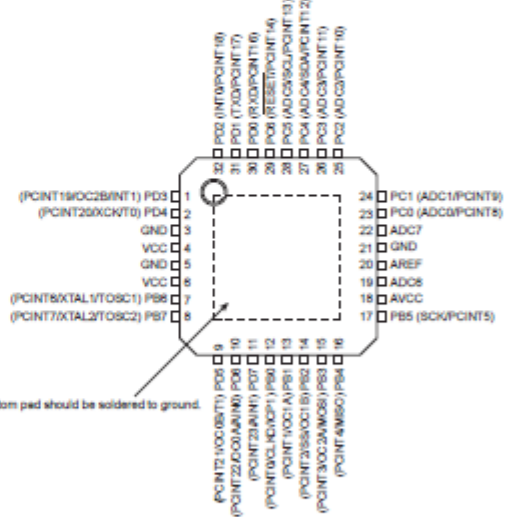


28 MLF Top View



NOTE: Bottom pad should be soldered to ground.

32 MLF Top View



NOTE: Bottom pad should be soldered to ground.

Table 1-1. 32UFBGA - Pinout ATmega48A/48PA/88A/88PA/168A/168PA

	1	2	3	4	5	6
A	PD2	PD1	PC6	PC4	PC2	PC1
B	PD3	PD4	PD0	PC5	PC3	PC0
C	GND	GND			ADC7	GND
D	VDD	VDD			AREF	ADC6
E	PB6	PD6	PB0	PB2	AVDD	PB5
F	PB7	PD5	PD7	PB1	PB3	PB4

Тема: Изучение работы центрального процессора микроконтроллера с помощью интегрированной среды разработки AVR STUDIO

Цель: научиться использовать для написания программ интегрированную среду разработки (ИСП) AVR Studio. Изучить работу основных команд центрального процессора с помощью симулятора.

Задание: разработать и промоделировать исполнение основных команд центрального процессора в среде программирования AVR Studio.

Общие сведения

Процессор микроконтроллера выполняет три основные операции, обеспечивающие выполнение программы:

- 1) формирует адрес очередной команды;
- 2) выбирает команду из памяти;
- 3) организует выполнение команды.

Основные узлы процессора:

счетчик команд или программный счетчик (PC);

дешифратор (декодер) команд;

арифметико-логическое устройство (ALU);

блок регистров общего назначения POH (GPR);

регистр состояния МК (SREG);

регистр-указатель стека (SP).

При работе МП под управлением программы реализуется алгоритмический принцип следования (не рассматриваются многопроцессорные системы). Программный счетчик является указателем на подлежащую выполнению команду и содержит адрес этой команды в памяти программ. Он работает в 3-х режимах:

- 1) сброс (при пуске и перезапуске МК очищается PC и выполнение программы начинается с команды, расположенной в памяти по нулевому адресу);
- 2) инкремент адресов (значение PC увеличивается, указывая на команду программы, следующую за выполняемой);
- 3) установка PC в соответствии с кодом команды условного или безусловного перехода.

Установку программного счетчика для выбора следующей команды реализует дешифратор команд (логика программирования). Дешифратор, также, по коду команды вырабатывает управляющие

сигналы, обеспечивающие её выполнение. Для полного выполнения командного цикла может потребоваться несколько тактов синхронизации, а некоторые команды выполняются за несколько командных циклов.

Арифметико-логическое устройство (Arithmetic Logic Unit) служит исключительно для выполнения арифметических операций (сложение, вычитание, логические И и ИЛИ, сравнение, сдвиг разрядов, установка регистра состояния в соответствии с результатом операции). Данные, с которыми напрямую работает АЛУ, содержатся в РОН. Результат операции (кроме операций сравнения) помещается в РОН и, дополнительно, в регистре состояния МК SREG устанавливаются признаки результата (флаги нуля, переноса, переполнения и др.). Назначение каждого из битов (флагов) регистра SREG иллюстрирует рисунок 1.

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	$\leftrightarrow$	$\leftrightarrow$	$\leftrightarrow$	$\leftrightarrow$	$\leftrightarrow$	$\leftrightarrow$

- H:** $Rd3 \cdot Rr3 + Rr3 \cdot \overline{R3} + \overline{R3} \cdot Rd3$
Set if there was a carry from bit 3; cleared otherwise
- S:** $N \oplus V$, For signed tests.
- V:** $Rd7 \cdot Rr7 \cdot \overline{R7} + \overline{Rd7} \cdot \overline{Rr7} \cdot R7$
Set if two's complement overflow resulted from the operation; cleared otherwise.
- N:** $R7$
Set if MSB of the result is set; cleared otherwise.
- Z:** $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is \$00; cleared otherwise.
- C:** $Rd7 \cdot Rr7 + Rr7 \cdot \overline{R7} + \overline{R7} \cdot Rd7$
Set if there was carry from the MSB of the result; cleared otherwise.

Рисунок 1 – Флаги регистра состояния процессора и их назначение.

AVR Studio 4 – профессиональная интегрированная среда разработки (Integrated Development Environment – IDE), предназначенная для написания и отладки прикладных программ для AVR микропроцессоров в среде Windows 9x/NT/2000. AVR Studio 4 содержит ассемблер и симулятор.

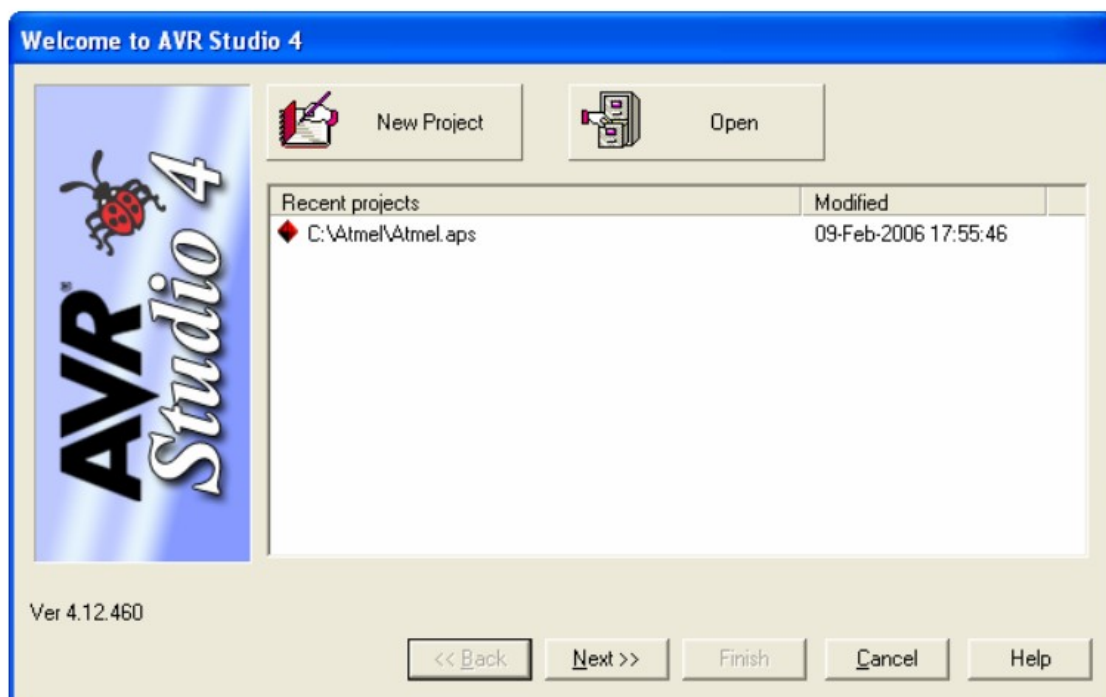
Начало работы

При программировании в среде AVR Studio необходимо выполнить стандартную последовательность действий:

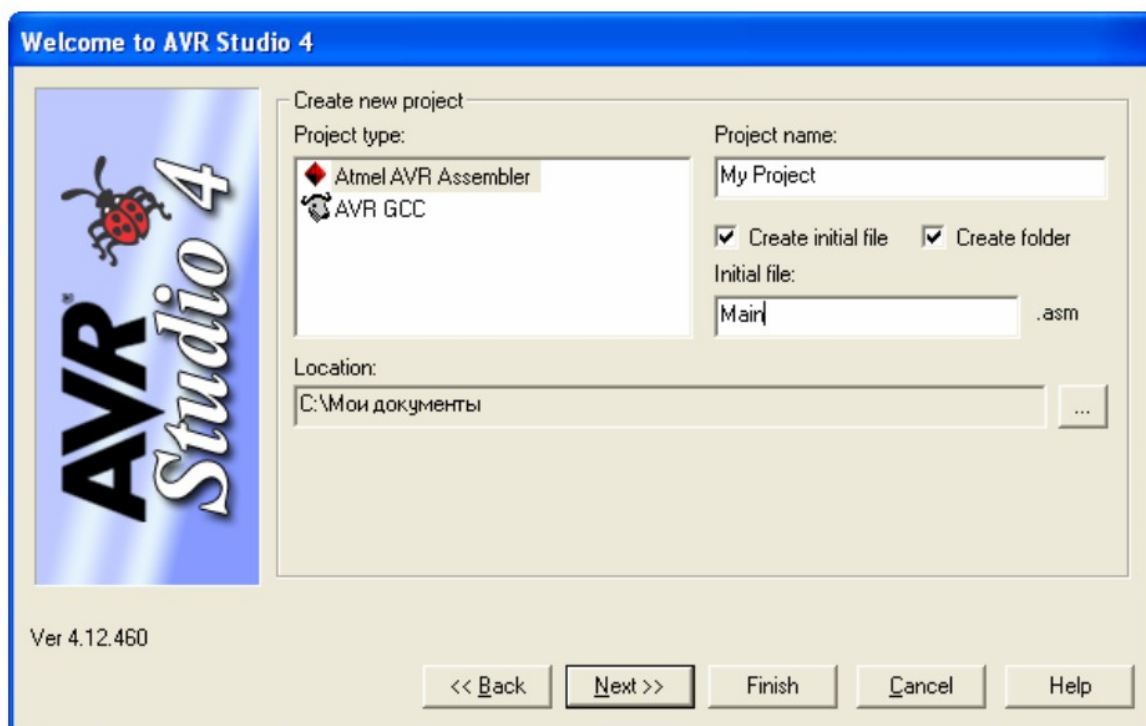
- создание проекта;
- написание программы;
- компиляция;
- симуляция.

Создание проекта

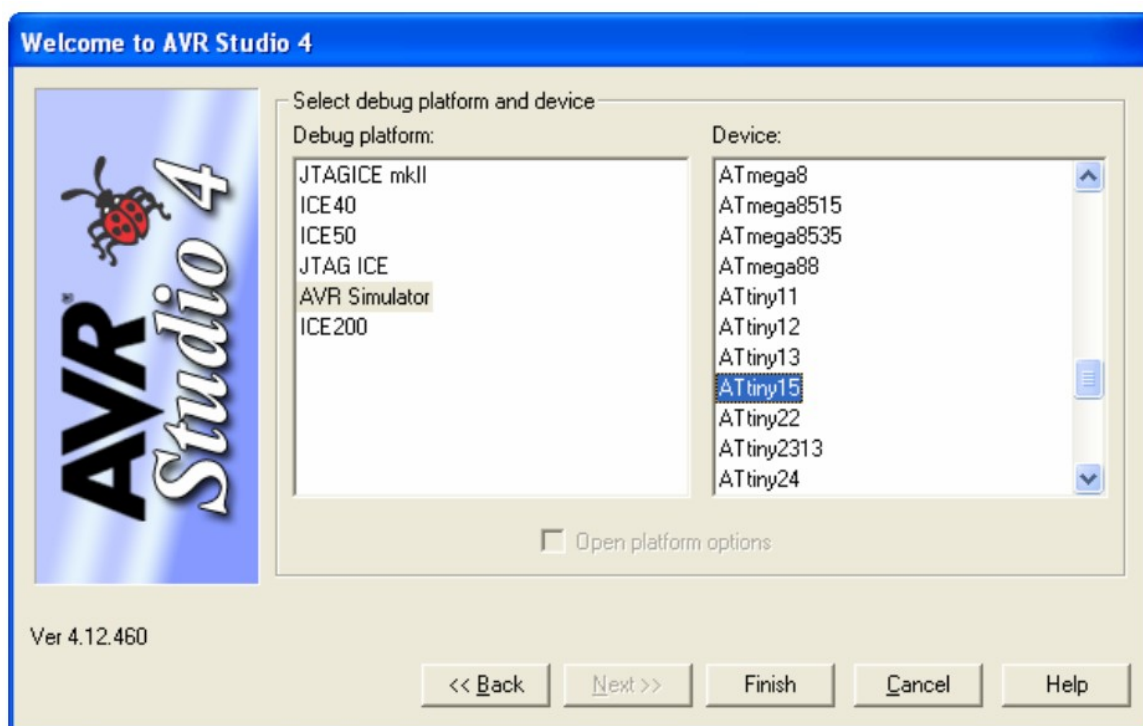
При запуске AVR Studio предлагает либо создать новый проект (New Project) или открыть уже существующий (Open).



Создаем новый проект

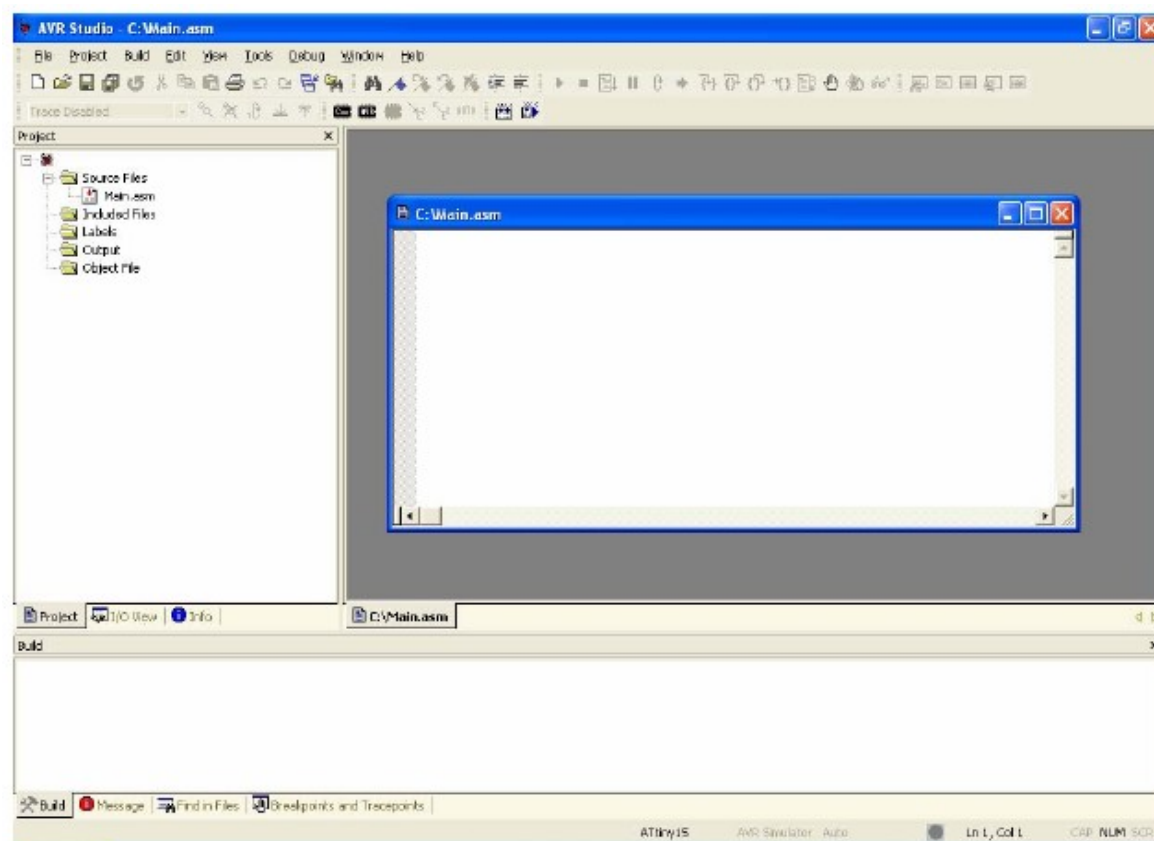


В окне тип проекта (Project Type) выбираем ассемблер (Atmel AVR Assembler), заполняем поля имя проекта (Project Name) и заглавный файл (Initial File). Нажимаем далее (Next)...



В окне платформа отладки (Debug Platform) выбираем симулятор (AVR Simulator), а в окне устройство (Device) – подходящий микроконтроллер (в данном варианте ATmega328P). Нажимаем завершить (Finish) – на этом проект создан, и мы попадаем в главное окно программы.

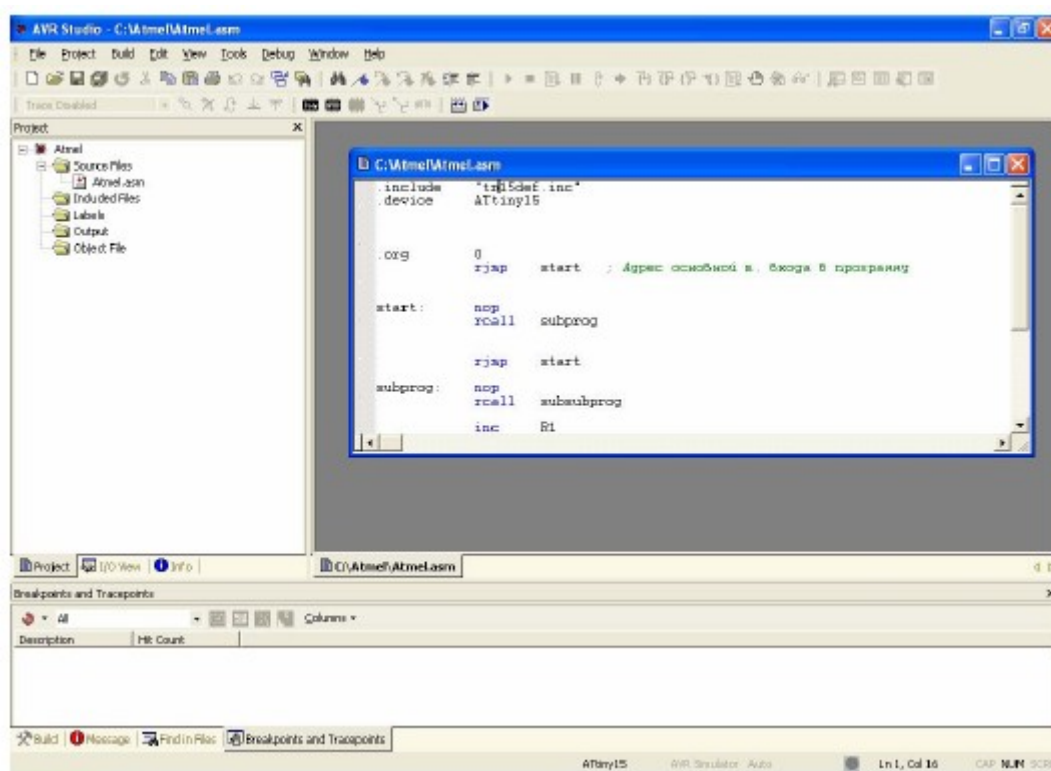
Общий вид окна программы



Окно разделено на 4 части. В верхней части расположены строка меню и «плавающие» панели с кнопками. Чуть ниже слева располагается вкладки Диспетчер проекта (Project), Просмотр ввода/вывода (I/O View), Информация (Info), справа – Текст программы. Снизу расположены следующие вкладки: Сборка программы (Build), Сообщения (Message), Поиск в файлах (Find in Files), Контрольные точки (Breakpoints and Tracepoints).

Написание текста программы

В окне Текст программы пользователь создает программу. Для первого знакомства можно взять программу из Приложения.



При написании программы инструкции выделяются синим цветом, комментарии – зеленым, остальное – черным. При написании ПО следует не забывать периодически её сохранять.

Компиляция

Компиляция – процесс перевода текста программы, написанной на языке программирования, в исполняемый модуль, содержащий машинные команды конкретного процессора. Применительно к микроконтроллерам процесс компиляции называется ассемблированием.

Ассемблирование — трансляция с языка ассемблера в команды машинного языка.



Данные кнопки на верхней панели запускают процесс ассемблирования. Кнопка слева ассемблирует проект, справа – ассемблирует и запускает на выполнение. Если при написании текста программы были допущены синтаксические ошибки, компиляция прерывается и во вкладке **Сборка (Build)** выдается сообщения о допущенных ошибках.

```
Build
AVRASM: AVR macro assembler 2.1.2 (build 99 Nov  4 2005 09:35:05)
Copyright (C) 1995-2005 ATMEL Corporation

C:\Atmel\Atmel.asn(1): Including file 'C:\Program Files\Atmel\AVR Tools\AvrAssembler2\appnotes\tn15def.inc'
• C:\Atmel\Atmel.asn(7): error: rjp: Unknown instruction or macro
• C:\Atmel\Atmel.asn(7): error: start: Unknown instruction or macro

Assembly failed, 2 errors, 0 warnings

Build Message Find in Files Breakpoints and Tracepoints
```

После удачного ассемблирования можно переходить к симуляции.

Симуляция

Симуляция – моделирование процесса выполнения программы микроконтроллером на персональном компьютере. Иначе говоря – режим отладки (Debugging) с помощью моделирования. Отладка – этап компьютерного решения задачи, при котором происходит устранение явных ошибок в программе. Часто производится с использованием специальных программных средств — отладчиков. Для управления режимом отладки предназначены следующие кнопки.

-  - Запустить отладку (симуляцию).
-  - Остановить отладку.
-  - Запустить программу на выполнение.
-  - Пауза в выполнении программы.
-  - Показать выполняемую инструкцию.
-  - Перезапустить программу.
-  - Шаг вперед с заходом в подпрограммы.
-  - Шаг вперед без захода в подпрограммы.
-  - Перейти к последней инструкции программы (подпрограммы).
-  - Выполнить программу до места указанного курсором.
-  - Автоматическое пошаговое выполнение программы.
-  - Установить/снять контрольную точку.
-  - Удалить все контрольные точки.

Инструкция которая будет выполняться следующей обозначена желтой стрелочкой.

Контрольная точка – инструкция в программе, дойдя до которой выполнение программы приостановится. Установленная контрольная точка отмечена красным кружком.

```

subsubprog:  nop
             rcall    sublast
             inc      R2
             ret

```

В пакете AVR Studio предусмотрено множество средств облегчающих программирование. А именно...



- **Смотровое окно** – предназначено для просмотра и редактирования значений всех определенных символов.

Watch				
Name	Value	Type	Location	
R1	0 ''	Register	R1	
R2	0 ''	Register	R2	
R13	0 ''	Register	R13	
PortB	0 ''	I/O Register	0x0018 [I0]	
tmp	0 ''	Register	R14	
⏮ ⏪ ⏩ ⏭ Watch 1 Watch 2 Watch 3 Watch 4				



- **Окно регистров** – предназначено для просмотра и редактирования значений регистров.

Register	
R00= 0x00	R01= 0x00
R02= 0x00	R03= 0x00
R04= 0x00	R05= 0x00
R06= 0x00	R07= 0x00
R08= 0x00	R09= 0x00
R10= 0x00	R11= 0x00
R12= 0x00	R13= 0x00
R14= 0x00	R15= 0x00
R16= 0x00	R17= 0x00
R18= 0x00	R19= 0x00
R20= 0x00	R21= 0x00
R22= 0x00	R23= 0x00
R24= 0x00	R25= 0x00
R26= 0x00	R27= 0x00
R28= 0x00	R29= 0x00
R30= 0x00	R31= 0x00



- **Окно памяти** – показывает распределение и содержание памяти в микроконтроллере (памяти программ, регистров общего пользования, регистров ввода/вывода, EEPROM и др.).

Memory				
Register	8/16	abc.	Address: 0x0	
EEPROM	00 00 00			
I/O	00 00 00			
Program	00 00 00			
Register	00 00 00			
000012	00 00 00	00 00 00		
000018	00 00 00	00 00 00		
00001E	00 00		..	



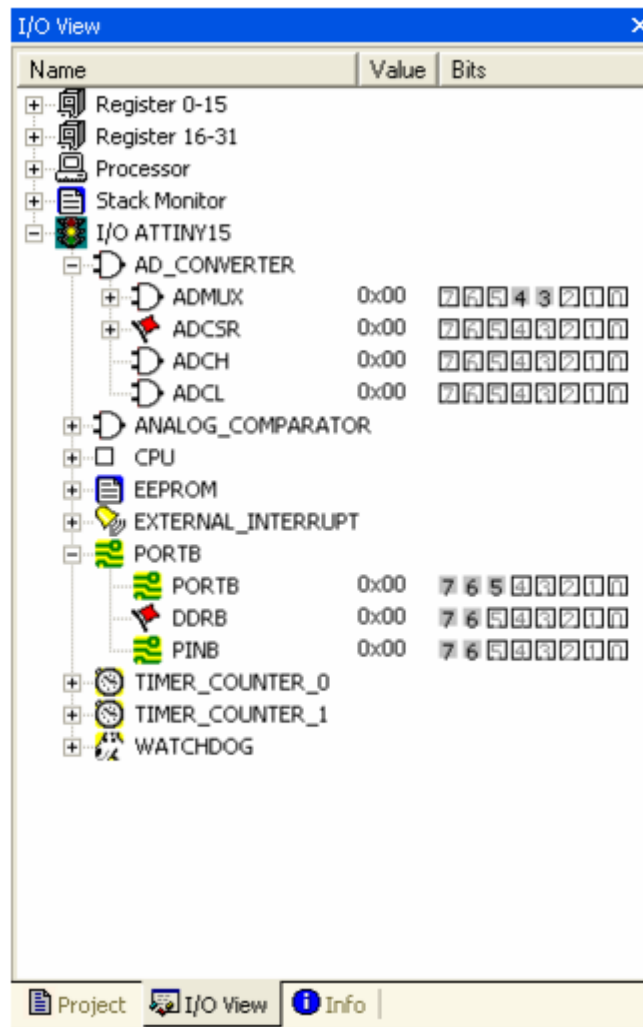
- Окно

дизассемблирования – показывает как программа транслировалась в машинные инструкции и по каким адресам они расположены.

----- Atmel asm -----				
7:		rjmp start ; Адрес основной м. Входа в программу		
+00000000:	C000	RJMP PC+0x0001	Relative jump	
@00000001:	start			
10:	start:	nop	No operation	
+00000001:	0000	NOP	No operation	
11:		rcall subprog		
+00000002:	D001	RCALL PC+0x0002	Relative call subroutine	
14:		rjmp start		
+00000003:	CFD	RJMP PC-0x0002	Relative jump	
@00000004:	subprog			
16:	subprog:	nop	No operation	
+00000004:	0000	NOP	No operation	
17:		rcall subsubprog		
+00000005:	D002	RCALL PC+0x0003	Relative call subroutine	
19:		inc R1	Increment	
+00000006:	9413	INC R1	Increment	
20:		ret	Subroutine return	
+00000007:	9508	RET	Subroutine return	
@00000008:	subsubprog			
22:	subsubprog:	nop	No operation	
+00000008:	0000	NOP	No operation	
23:		rcall sublast		
+00000009:	D002	RCALL PC+0x0003	Relative call subroutine	
25:		inc R2	Increment	
+0000000A:	9423	INC R2	Increment	
26:		ret	Subroutine return	
+0000000B:	9508	RET	Subroutine return	
@0000000C:	sublast			
28:	sublast:	nop	No operation	
+0000000C:	0000	NOP	No operation	
31:		ret	Subroutine return	
+0000000D:	9508	RET	Subroutine return	
----- No Source -----				
+0000000E:	FFFF	???	Data or unknown opcode	
+0000000F:	FFFF	???	Data or unknown opcode	

Дизассемблирование - перевод машинных кодов какой-либо программы в ее представление на языке ассемблера.

Информация о регистрах ввода/вывода, процессоре и регистрах общего пользования расположена и распределена по группам в вкладке **Просмотр ввода/вывода (I/O View)**.



После того как проект прошел отладку данную программу можно загружать (зашивать) в микроконтроллер, т. е. записывать во внутреннюю память программ микроконтроллера.

Контрольные вопросы

1. Что такое AVR Studio.
2. Какая стандартная последовательность действий при разработке программы для микроконтроллера.
3. Что такое ассемблирование.
4. Чем ассемблирование отличается от компиляции.
5. Что такое симуляция.
6. Зачем производят отладку программы.
7. Что такое контрольная точка.
8. Что такое дизассемблирование.
9. Зачем предназначена вкладка Просмотр ввода/вывода.

Литература (файлы)

1. Евстифеев__AVR.pdf
2. Atmega 48A-48P...datasheet.pdf
3. Commands_AVR.pdf
4. m328P def.inc

Приложение 1

Текст программы

```
; PROJECT: CPU

;Тестовая программа для изучения основных команд центрального процессора
;с помощью ИСР AVR Studio 4

; AUTOR: студент группы ЭЛ-05м-24 Ф.И.О

.include "m328Pdef.inc" ; подключаемые файлы

; Обозначения констант

.equ   PORT_SET = 0x04
.equ   PORT_OUT= 0x05

; Обозначения переменных

.def   temp = R16
.def   DATA1 = R17
.def   DATA2 = R18

;***

; Вектора функций

.cseg           ;директива программного сегмента
.org   0        ;установка программного счётчика в 0
rjmp setup      ;переход на функцию (начальный адрес) инициализации системы
; ***

; Функции

setup:          ;инициализация (начальная настройка) системы
ldi   temp, 0b11111111
out   PORT_SET, temp

; Основная программа (главная функция)

main:
ldi DATA1, 0xB8
ldi DATA2, 57
sbc DATA1, DATA2
out PORT_OUT, DATA1
rjmp main
```

Приложение 2

Титульный лист отчёта по практической работе

ФГБОУ ВО

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ "МЭИ"
ИНСТИТУТ ЭЛЕКТРОТЕХНИКИ**



КАФЕДРА ЭКАО И ЭТ

Направление подготовки: 13.04.02 Электроэнергетика и электротехника

Профиль подготовки: Теория движения электроподвижного состава и проблема оптимизации тягового оборудования и устройств электроснабжения транспортных систем

Квалификация (степень) выпускника: магистр

Форма обучения: очная

ОТЧЁТ

по практической работе № 1

по дисциплине ИЭТ; Б1.Б.3 "Информационные и компьютерные технологии в электротехнике"

Тема: Изучение работы центрального процессора микроконтроллера

Студент _____

Научный руководитель _____

Фамилия, и., о. группа подпись
доцент_к.т.н._Комаров_В.Г.
должность, звание, фамилия, и., о.

Допущен	Принято	Оценка

Москва

2026 г.