

**МИНЕСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ РЕСПУБЛИКИ  
УЗБЕКИСТАН  
НУКУССКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
ИМЕНИ МУХАММАДА АЛЬ-ХОРЕЗМИ**



**Направление:** \_\_\_\_\_

# **Курсовая Работа**

**По предмету** \_\_\_\_\_

**На тему** \_\_\_\_\_

**Выполнил(а):**

**Рузмуродов Х.**

**Принял(а):**

\_\_\_\_\_

**НУКУС 2018**

# **ПРОЕКТИРОВАНИЕ СИСТЕМ ДЛЯ МОНИТОРИНГА ОБЪЕКТОВ**

## **СОДЕРЖАНИЕ**

### **ВВЕДЕНИЕ**

- 1. Анализ основных характеристик микроконтроллера семейства AVR**
- 2. Микроконтроллер Atmega8**
- 3. Методика проектирования в среде CODEVISIONAVR**
- 4. Методика программирования микроконтроллера**
- 5. Заключение**
- 6. Литература**

## **Введение.**

Актуальность работы обусловлена популярностью AVR-микроконтроллеров при проектировании информационно-управляющих средств широкого спектра. Микроконтроллеры Atmel семейства AVR просты в использовании и интеграции, имеют низкую потребляемую мощность и обладают самой эффективной архитектурой для программирования на C или ассемблере .

Цель работы – разработать обобщенную методу проектирования цифровых информационно-управляющих систем на базе МК Atmega8 с использованием интегрированных инструментов среды для программирования codevisionAVR.

В рамках работы проанализированы основные особенности разработки информационно-управляющих систем на базе МК Atmega8 и предложена авторская методика на основе среды codevisionAVR для программирования и настройки МК .

В работе на основе формализованного информационного минимума предложена обобщенная методика проектирования цифровых систем на базе AVR микроконтроллеров, которая обеспечивает минимизацию уверенных манипуляций при разработке ЭС в рамках типовых проектов проектирования ЭС с использованием МК.

Микроконтроллеры получили широкое распространение для управления электронными устройствами и выполнения логических и арифметических операций за счет возможности многократного программирования и универсальности. Они позволяют значительно упростить схемотехническую составляющую разрабатываемого устройства, а также позволяют значительно уменьшить габариты ЭС.

Универсальность микроконтроллера, значительное сокращение габаритов и при этом увеличение производительности, низкая стоимость и энергопотребление сделали микроконтроллер широко используемым при реализации большого спектра устройств различного назначения.

По сравнению с цифровыми сигнальными процессорами МК можно также использовать в области обработки сигналов и выполнении различных арифметических операций, однако, в отличие от сигнального процессора микроконтроллер эффективнее применять в приложениях, где он будет выполнять функции в области управления и ввода-вывода данных. В рамках продвижения решения на базе МК на рынке следует учитывать скорость и масштабы развития технологий в наше время, а следовательно необходимо рационально оценивать актуальность продвигаемого решения и тщательно анализировать возможные варианты исхода того или иного вида продвижений .

Одним из широко распространенных являются микроконтроллеры семейства AVR компании Atmel, предлагающий линейку 8-и, 16-и, 32-х разрядных архитектур микроконтроллеров с различными характеристиками (флеш-память, EEPROM, системы команд, встроенные интерфейсы и т.д.) и назначением (решения задач средней и высокой сложности, задач, требующих большого объема памяти и высокой скорости и т.д.). Еще одним популярным семейством микроконтроллеров является PIC компании Microchip Technology Inc. Производитель также предлагает 8-и, 16-и, 32-х разрядные микроконтроллеры с различными архитектурами и характеристиками. Еще одно популярное семейство это ARM-микроконтроллеры компании ARM Limited. Производитель предлагает различные подсемейства МК с различными возможностями и назначением.

Таким образом, не стоит преуменьшать важность МК, использование которого значительно облегчает решение таких вопросов как управление электронными устройствами, обеспечение минимизации габаритов, увеличение производительности и эффективности.

### **Аннотация**

В данной работе представлена методика программирования AVR-микроконтроллера на примере ATmega8 как этапа проектирования цифровых систем управления на базе AVR-микроконтроллеров. Рассматриваются структура типового AVR-микроконтроллера и функции его основных компонентов, подробно описаны характеристики микроконтроллера ATmega8. В статье приведены этапы программирования ATmega8 посредством использования интегрированных средств среды для программирования CodeVisionAVR и дан последовательный алгоритм. В конце статьи даны рекомендации по тестированию запрограммированного микроконтроллера.

### **Анализ основных характеристик микроконтроллера семейства AVR.**

Типовой микроконтроллер включает в себе функции микропроцессора (это не тоже самое, что микроконтроллер) и периферийных устройств, содержит оперативное запоминающее устройство (ОЗУ) и постоянное запоминающее устройство (ПЗУ). На рисунке 1.1 изображена обобщенная структурная схема типового микроконтроллера .

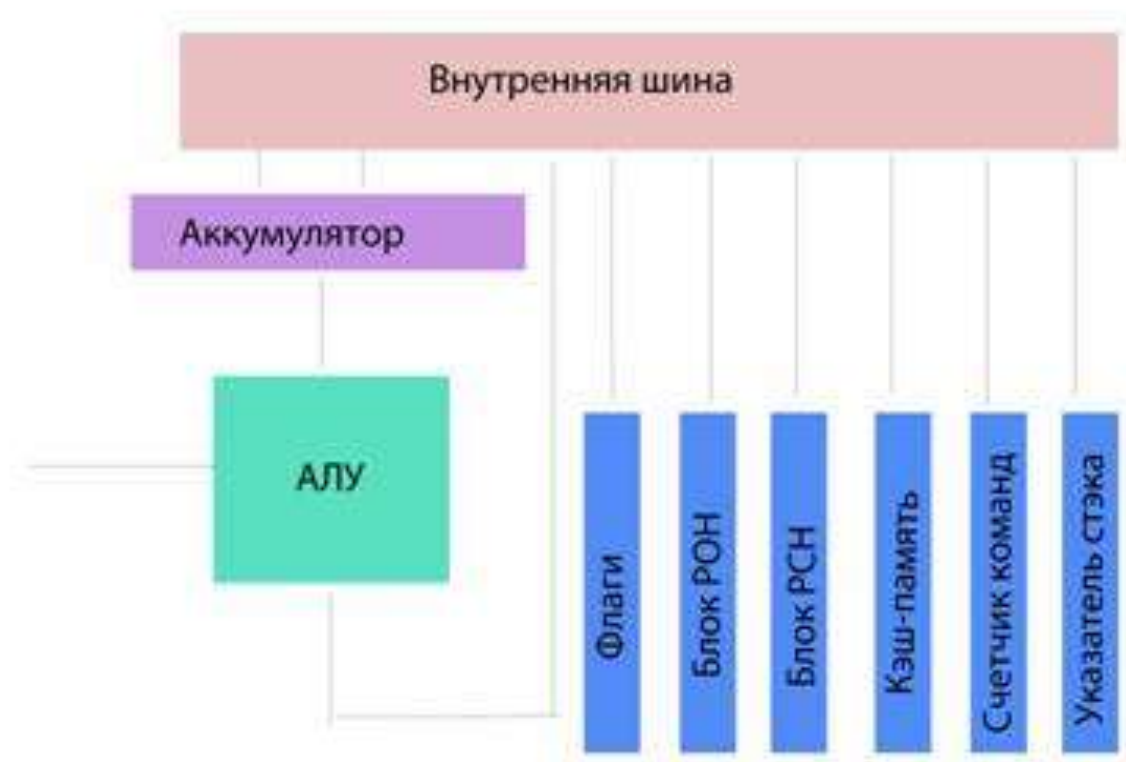


Рисунок 1.1 – Типовая обобщенная структурная схема МК

Можем выделить основные функциональные части:

- внутренняя шина соединяет все функциональные блоки между собой ;
- арифметическо – логическое устройство (АЛУ) – блок МК, выполняющий арифметические и логические операции;
- аккумулятор – это регистр МК, предназначенный для временного хранения операнда или промежуточного результата арифметических или логических операций, производимых АЛУ;
- регистры общего назначения (РОН) – регистры, используемые в арифметических операциях;
- регистры специального назначения (РСН) – регистры, в которых содержится служебная информация, определяющая настройки МК;

- счетчик команд – регистр, содержащий адрес текущей выполняющейся команды
- стэк- структура данных, организованная по принципу LIFO( last in first out)
- кэш память – это высокоскоростная память доступа, используемая МК для временного хранения информации, которая вероятнее всего будет использована;
- регистр флагов – содержит флаги МК, отражающие его состояние;

Микроконтроллер Atmega8 это самый популярный микроконтроллер семейства AVR производителя Atmel, который максимально прост в использовании и реализации конкретных решений в сфере электроники даже на уровне радиолюбителей.

AVR микроконтроллеры имеют гарвардскую архитектуру, которую можно определить как архитектуру, основными отличительными особенностями которой являются:

-хранилище инструкций и хранилище данных представляют собой разные физические устройства;

– канал инструкций и канал данных так же физически разделены.

Схема МК с гарвардской архитектурой приведена на рисунке 1.2.



Рисунок 1.2 – Гарвардская архитектура МК

Таким образом микроконтроллер уже не представляет собой черный ящик, а является совокупностью функциональных блоков, имеющих определенное назначение.

Следующим шагом будет обзор характеристик конкретного микроконтроллера ATmega8, основных принципов и подходов при программировании данного МК и формирование алгоритма методики его программирования.

### **Микроконтроллер Atmega8.**

Микроконтроллер Atmega8 как уже было сказано один из самых простых и в тоже время надежных и эффективных МК, которые можно использовать при проектировании ЭС. Реализации микросхем может быть в нескольких



видах корпуса PDIP, TQFP и MLF, выбор конкретной реализации зависит от конструктивных особенностей разрабатываемого устройства.

Рассмотрим микроконтроллер в корпусе PDIP приведенный на рисунке 1.3.

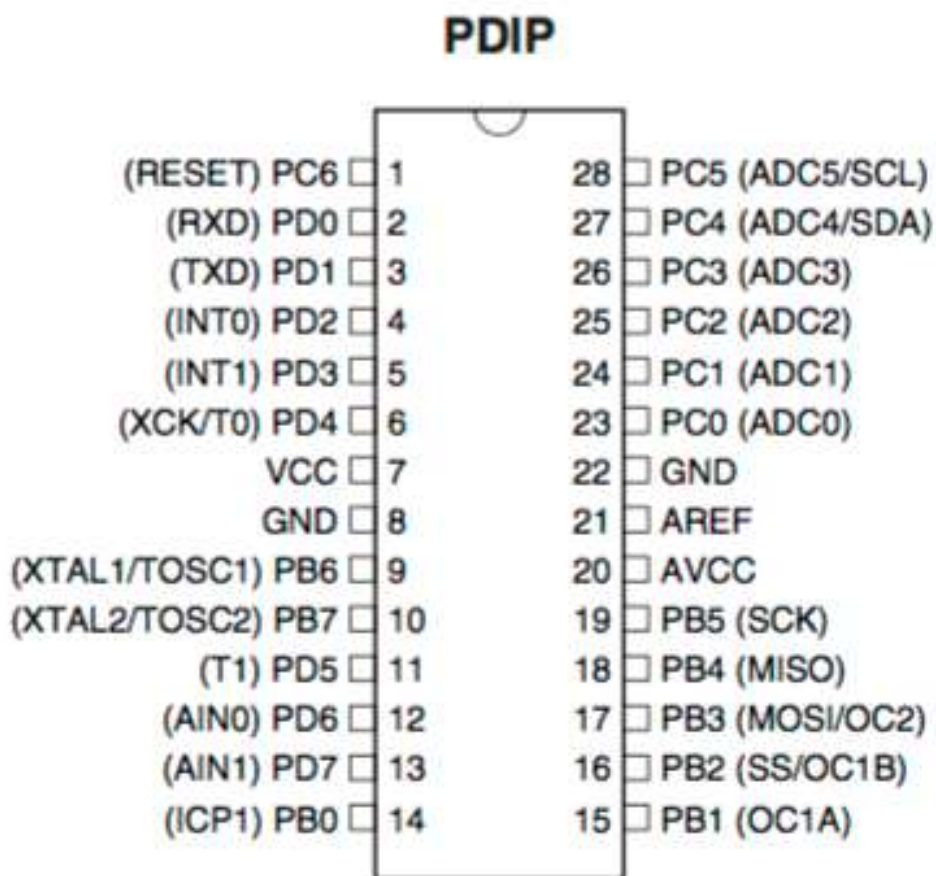


Рисунок 1.3 – Микроконтроллер Atmega8 в корпусе PDIP

Характеристики микроконтроллера приведены в сопроводительной документации к МК, о них не стоит забывать и учитывать при разработке устройства.

Для того чтобы осилить программирование микроконтроллера необходимо прежде всего знать о распиновке микросхемы и назначению той или иной ножки МК. Подробное описание каждого пина можно узнать из Datasheet к данному микроконтроллеру, стоит заметить, что вообще всю важную

информацию о том или ином электронном компоненте можно получить непосредственно в datasheet на него.

Стоит обратить внимание на структуру пинов на рисунке 1.3 группы PC(0...6) PB(0...6) PD(0...7) это пины конкретных портов ввода/вывода C, B и D соответственно, эти порты предназначены для подключения микроконтроллера в основную схему, например, если необходимо подключить ЖК экран, то он будет подключен с помощью какого либо из этих трех портов. Следует отметить назначение линии – AREF – напряжение на этом выводе поступает на АЦП и конвертируется, после этого определяется напряжение, которое позволяет АЦП генерировать максимальное цифровое значение, т.е 1023. Остальные значения напряжения, поступающие в АЦП, конвертируются пропорционально. При настройке по умолчанию напряжения 5 В, напряжение 2.5 В в АЦП будет конвертироваться в 512.

- AVCC – пин для подключения питания для АЦП, он должен быть внешне подключен к VCC, даже если АЦП не используется. Если АЦП используется, он должен быть подключен к VCC через низкочастотный фильтр.

Основные пины описаны, однако на рисунке 1.3 возле некоторых пинов можно увидеть помимо обычного обозначения обозначение в скобках, например возле PC6 есть значение в скобках RESET, данная надпись означает, что в зависимости от настройки МК, его ножки могут работать в разном режиме, например в данном случае при соответствующей настройке МК, при подаче низкого уровня сигнала на данный пин инициируется сброс микроконтроллера и перезапуск. Все остальные нюансы дополнительных режимов работы можно узнать из datasheet atmega8.

Настройка выводов микроконтроллера осуществляется через fuse-биты или конфигурационные биты. Каждый конфигурационный бит можно установить в 0 – это будет означать что он запрограммирован или в 1 – не запрограммирован. По умолчанию все конфигурационные биты микроконтроллера установлены в 1, т.е. не запрограммированы. Настройка данных битов устанавливает конкретные правила для микроконтроллера и определяет его режим работы как в случае описанном выше для РС6, либо он работает как пин in/out, либо является пином сброса МК. Правильно настроить fuse-биты можно с помощью специальной программной среды для прошивки AVR-микроконтроллера CodeVisionAVR, работа с которой будет рассмотрена в следующей главе. Подробное назначение конфигурационных битов представлено в .

Все разнообразие комбинаций установки конфигурационных битов и возможные функции описаны в datasheet к МК и используются для реализации определенных задач и целей. Как можно заметить из описания назначения битов эти настройки крайне важны и требуют осознанные манипуляции ними, если целью является разработка работоспособного устройства с МК, который работает корректно.

Таким образом после выбора микроконтроллера для конкретного решения необходимо внимательно изучить:

- характеристики МК, которые должны полностью удовлетворять требованиям к разрабатываемому устройству;
- распиновку и назначение пинов, для дальнейшей разработки печатной платы и правильного соединения с компонентами схемы и корректного использования каждого пина;

- схему, разрабатываемого устройства, для решения есть ли необходимость в использовании fuse-битов, если необходимость есть, разобраться какие комбинации fuse-битов нужно установить;

Из описания конфигурационных битов вполне ясно, что далеко не все из них нужно использовать и все зависит исключительно от состава разрабатываемого устройства.

### **Методика проектирования в среде CODEVISIONAVR.**

Среда CODEVISIONAVR представляет собой интегрированную среду разработки ПО для AVR микроконтроллеров .

CodeVisionAVR позволяет осуществлять разработку программного кода и осуществлять его компиляцию. Редактор позволяет осуществлять работу сразу с несколькими проектами одновременно и настроить автосохранение. Есть возможность генерирования емкого и результативного автокода для всего семейства микроконтроллеров AVR. Кроме компилирования CVAVR снабжен модулем прошивки, позволяющим записать уже скомпилированную программу на микроконтроллер с помощью программатора. Программа поддерживает все официальные популярные программаторы, которые можно выбрать в процессе настройки программатора( подробное описание ниже).

Отметим основные модули, входящие в состав CodeVisionAVR:

- трансляторы программ, составленных на Си-подобных языках или ассемблере, в машинный код для AVR;
- элементы для инициализации периферийных устройств;
- модуль для работы с платой отладки STK-500;
- компоненты взаимодействия с внешними программаторами;

- редактор первичного кода;
- терминальный модуль.

После создания готовой работоспособной программы в CVAVR, ее можно сохранить с расширением HEX, ROM или BIN, файлы с такими расширениями позволяют осуществлять прямую прошивку с использованием другого программатора. Кроме этого, данные файлы могут быть протестированы в таких программных средах как Proteus, где можно создать модель разрабатываемого устройства, загрузить HEX-файл и посмотреть корректно ли работает модель в анимации, данный способ уменьшает количество бесполезных прошивок микроконтроллера, количество которых ограничено(с количеством можно ознакомиться в datasheet) . Также, программа может быть передана в формате COFF (файл отладчика) или OBJ. Программная среда постоянно развивается и с каждым обновлением расширяется число библиотек и продукции, поддерживаемой средой, например таких известных производителей как Philips, National Semiconductor, Maxim-Dallas Semiconductor и других.

Стоит заметить, что компилятор CodeVisionAVR отличается от AVR-GCC, используемый в среде разработки WinAVR, синтаксисом, количеством библиотек и поддерживаемых продуктов и быстродействием выходного кода. Специфика разработки, командное меню и программные возможности среды CVAVR похожа на программу PonyProg.

Программа CVAVR позволяет осуществлять прошивку не только самой программы в flash-память МК, но и настраивать конфигурационные биты (см гл 1.2) и записывать EEPROM микроконтроллера, но не стоит забывать, что все описанные процедуры самостоятельные, требуют ручной настройки и осознанного подхода. Разработчик рекомендует следующий алгоритм прошивки:

- запись пользовательского кода программы;
- запись EEPROM;
- настройка конфигурационных битов (fuse-битов);

При запуске программы необходимо сконфигурировать CVAVR, для этого необходимо выбрать микроконтроллер, который в дальнейшем будем прошивать. В рамках данной статьи это Atmega8. Для этого необходимо выбрать CodeWizardAVR, после чего откроется окно как показано на рисунке

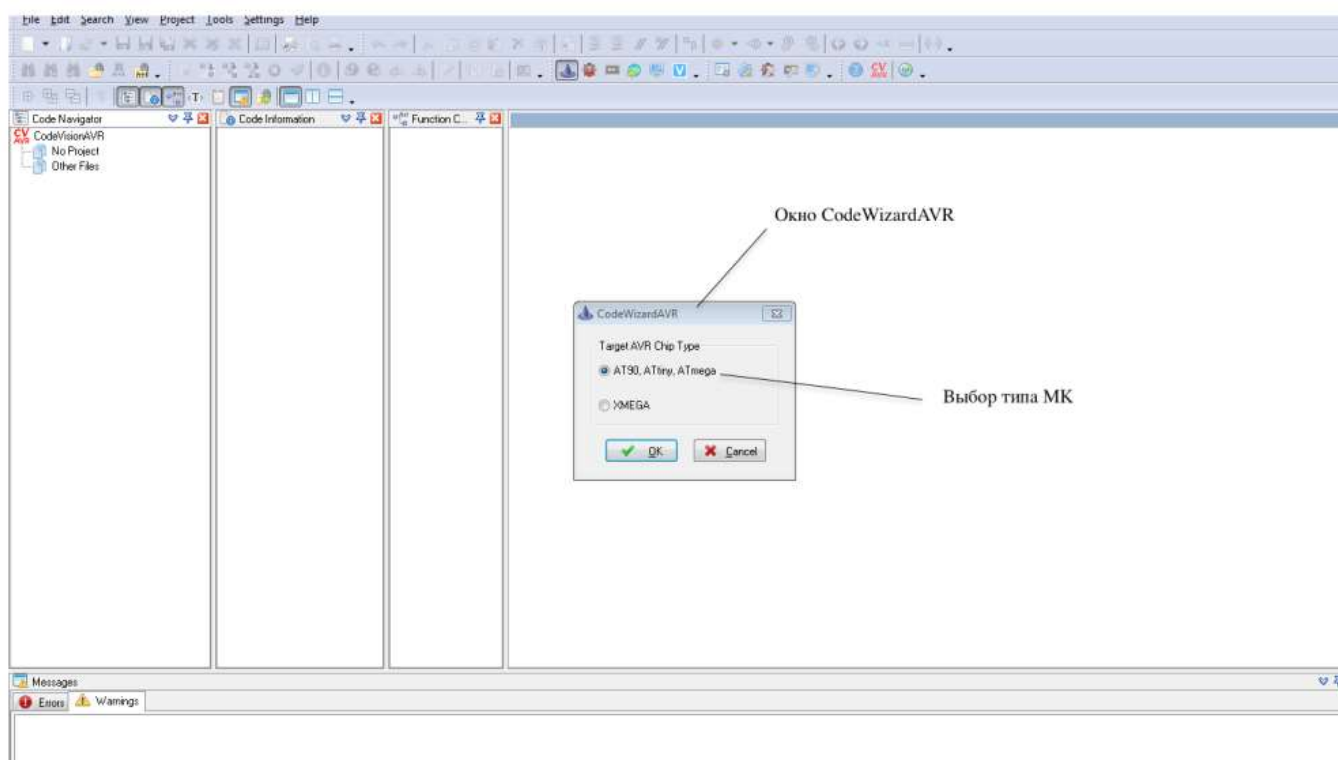


Рис. 2.1 – Окно CodeWizardAVR

В открывшемся окне необходимо выбрать пункт AT90, Attiny, Atmega, после чего откроется окно с первичными настройками МК (рисунок 2.2) В выпадающем окне chip необходимо выбрать свой МК, в нашем случае это Atmega8.

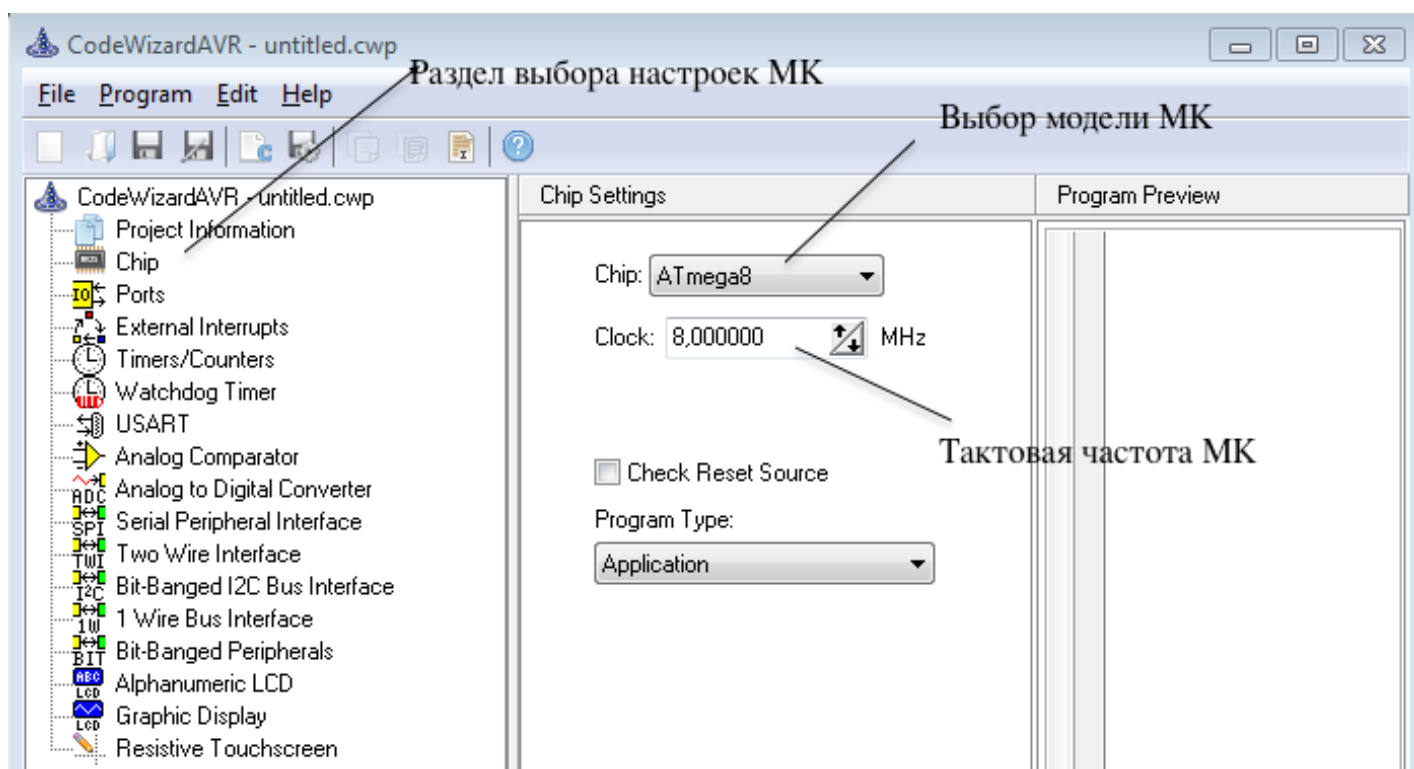


Рис. 2.2 – Окно выбора микроконтроллера в CVAVR

После выбора микроконтроллера в поле ниже отобразится частота внутреннего RC-генератора в данном микроконтроллере, в нашем случае это 8МГц. Здесь стоит пояснить о том, что это за частота и какие возможны изменения и настройки в связи с этим при нестандартных решениях при проектировании ЭС.

Микроконтроллер AVR работает на определенной тактовой частоте, которая характеризует скорость его работы. Этот параметр крайне важен для некоторых приложений( например приложений реального времени), скорость работы которых имеет прямую зависимость с работоспособностью и корректностью выполнения основной задачи. Поэтому тактовая частота без сомнения важна и необходимо ее знать или уметь определять. Одной инструкции на языке низкого уровня ассемблере соответствует длительность одного такта( стоит заметить, что существуют

инструкции, выполняемые за большее количество тактов), следовательно чем больше тактовая частота, тем быстрее реагирует МК и тем быстрее выполняет программный код. Также микроконтроллер содержит множество интерфейсов, таких как SPI, АЦП, UART, I2C, работа данных интерфейсов зависит от величины тактовой частоты.

Для приложений, где нет необходимости в точной генерации интервалов времени, или когда быстроедействие микроконтроллера не является самоцелью, можно использовать встроенный в микроконтроллер RC-генератор. Генератор имеет тактовую частоту равную 8 МГц, однако по умолчанию установлен предделитель на 8, поэтому реальная тактовая частота по умолчанию будет составлять 1 МГц. Стоит отметить, что предделитель может быть сконфигурирован не только на значение 8, но и на другие коэффициенты деления или же вовсе отключен, поэтому частоту внутреннего генератора можно настроить на диапазон частот от 1МГц до 8МГц. Наличие внутреннего генератора заметно облегчает и удешевляет устройство, вследствие исключения необходимости в внешнем кварцевом резонаторе. Однако необходимо учитывать недостаток RC-генератора, который заключается в его неточности, по сравнению с внешним кварцевым резонатором. Данный недостаток заметно ограничивает диапазон применения внутреннего генератора.

Тип подключаемого генератора и его тактовая частота определяется конфигурационными битами микроконтроллера AVR. По умолчанию микроконтроллер AVR сконфигурирован на взаимодействие с внутренним генератором и, как уже было сказано ранее, настроен на тактовую частоту, равную 1МГц.

После настройки раздела chip можно перейти к разделу ports (рисунок 2.3).

В данном разделе можно сконфигурировать порты микроконтроллера на



вход или выход, установив, соответственно, 1(out) – работы на выход, 0(in) – режим работы на вход.

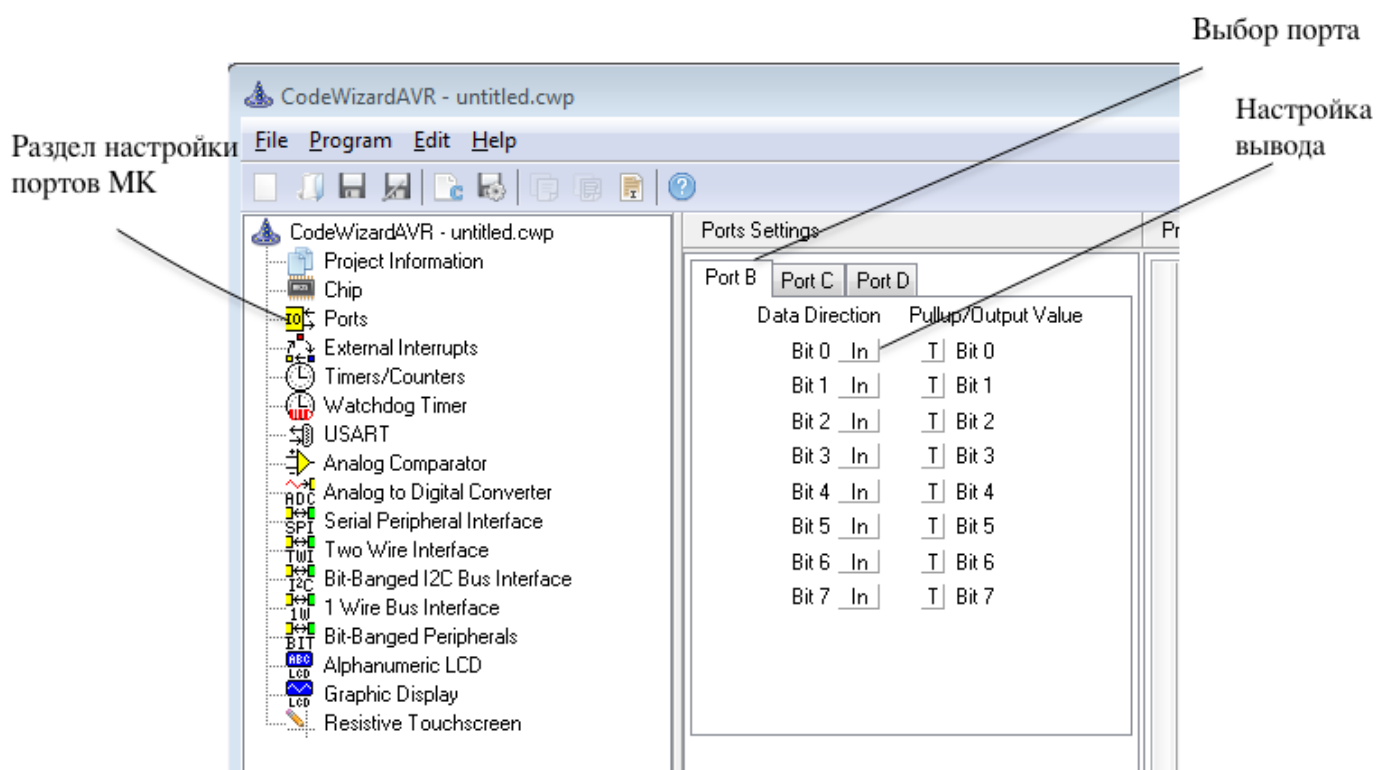


Рис. 2.3 – Окно настройки портов CVAVR

Здесь необходимо рассмотреть подробнее, мы не устанавливаем 0 и 1 в регистре конкретного порта, мы устанавливаем эти значения в специально существующем для этого регистре направления данных (DDR), который есть у каждого порта. Для того, чтобы сделать это не через CVAVR достаточно написать в коде программы запись, например: `DDRB=0x02`. Данная запись означает двоичное число 0000010 (разрядность определяется из количества пинов порта, в нашем случае 8), таким образом получаем, что 2 пин порта установится на выход, а все остальные пины будут настроены на выход.

Далее идут вкладки настроек различных видов соединений, таких как USART, 1 Wire, Two Wire, подключить и настроить таймеры, подключить АЦП, LCD экран и прочие настройки.

Как только предварительная настройка микроконтроллера завершена, необходимо нажать Program/Generate, Save and Exit для сохранения файлов проекта.

Далее откроется проект, файлом, в котором будет формироваться сама программа будет файл с расширением .c. Этот файл автоматически создается и после настройки микроконтроллера в нем отражены в виде кода все установки конфигурации портов, все подключаемые периферийные устройства, однако в закомментированном виде.

### **Методика программирования микроконтроллера.**

Как только программа написана и откомпилирована (компиляция кода должна быть успешна для дальнейшей прошивки на микроконтроллер) ее можно загрузить в микроконтроллер.

Как уже до этого было сказано для программирования МК необходим посредник между компьютером и микроконтроллером, которым является программатор и отладочная плата, в которую вставляется МК и подключается к ноутбуку. Программаторы бывают различные по принципу действия и типу подключения, выбираются в соответствии с характеристиками и возможностями. В общем случае методика проектирования базируется на конкретной методологии проектирования исходя из критериев качество-сроки-сопровождение .

Для начала приведен краткий алгоритм действий при прошивке МК в программной среде CVAVR:

- убедиться, что компиляция программы была успешно и все ошибки исправлены;
- подключить программатор к компьютеру и убедиться, что он активен (индикация на программаторе);
- выбрать в CVAVR свой программатор;
- настроить прошивку;
- прошить микроконтроллер;
- протестировать корректность прошивки.

Теперь рассмотрим подробнее предлагаемый алгоритм. Итак для выбора программатора необходимо выбрать Settings/Programmer и в открывшемся окне выбираем наш программатор (рисунок 2.4) и нажимаем ok.

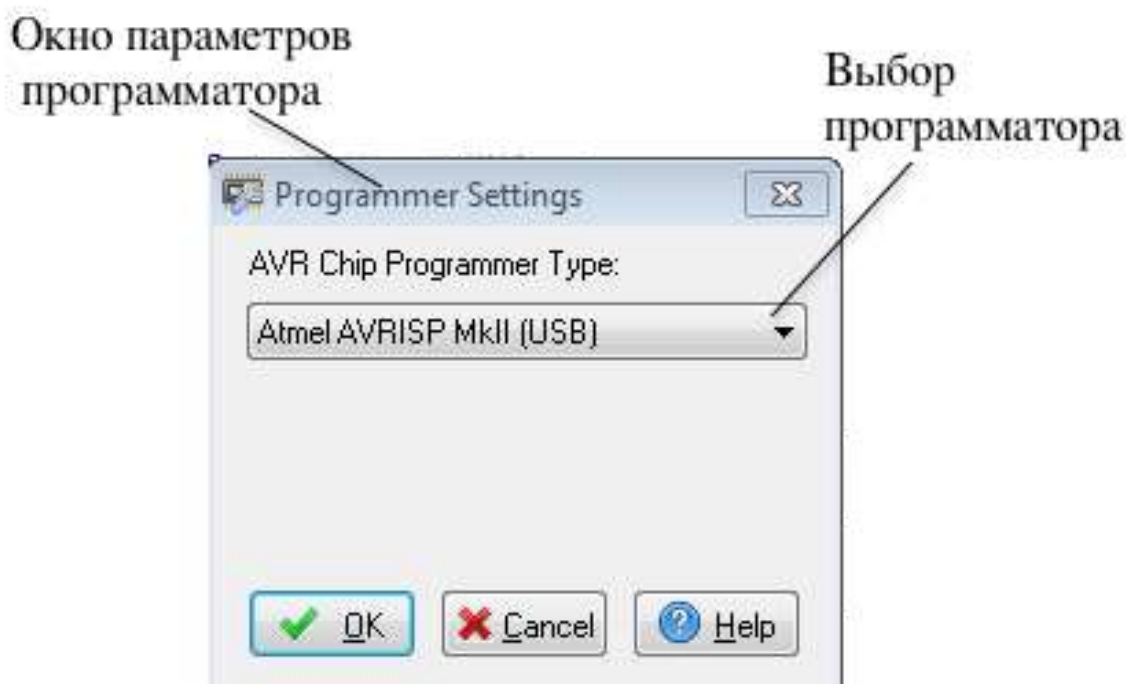


Рис. 2.4 – Выбор типа программатора CVAVR

Далее заходим в меню прошивки Tools/Chip Programmer и для проверки информацией о цифровой подписи необходимо выбрать read/chip signature. В случае, если все верно подключено, то появиться окно как на рисунке 2.5

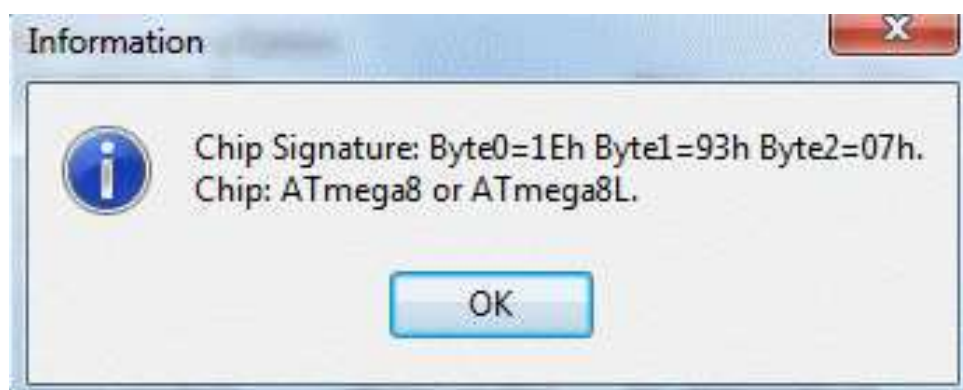


Рис. 2.5 – Информация о цифровой подписи МК

В случае, если программа уже была записана на ваш микроконтроллер, то стереть ее можно командой Program/Erase chip.

В правой части окна расположен список конфигурационных битов, настройка этих битов позволяет сконфигурировать микроконтроллер например на работу от внешнего кварцевого генератора с определенной частотой и прочие варианты, необходимость в которых может возникнуть.

Если необходимости в установке конфигурационных битов нет или отсутствует точное понимание как ими пользоваться необходимо снять метку с поля Program fuse bits и прошить только flash память МК, выбрав program/flash. После чего программа будет загружена в микроконтроллер.

Безусловно в данной статье приводится описание далеко не всех возможностей CVAVR, как например есть возможность программирования EEPROM памяти, актуальной для хранения пароля доступа в замке или других данных, которые необходимы в рамках конкретного ЭС.

После прошивки микроконтроллера, для того чтобы убедиться, что программа работает корректно можно собрать схему разрабатываемого устройства на макетной плате и подключить в схему прошитый микроконтроллер. Для того, чтобы увеличить вероятность успеха стоит перед прошивкой микроконтроллера смоделировать схему в программной среде Proteus, если программа работает корректно в модели, то скорее всего она заработает и на макетной плате в реальности.

Данная методика актуальна при работе цифровых управляющих систем, она может быть полезна при организации и проведении занятий для студентов технических специальностей, при выполнении курсовых и домашних работ, а также радиолюбителей.

## **Заключение.**

В данной работе был приведен обзор информации о конструкции структуры микроконтроллера, определены его главные функциональные составляющие и их назначение. Были рассмотрены основные характеристики микроконтроллера Atmega8, их назначение и возможности их настройки с помощью конфигурационных битов.

Вторая часть работы представляет собой методику программирования и способы настройки микроконтроллера на примере Atmega8, а также даны рекомендации по тестированию и моделированию разрабатываемого устройства.

В работе позволяет ознакомиться с интегрированной программной средой CodeVisionAVR для программирования управляющей программы микроконтроллера для разрабатываемого ЭС. Рассмотрены основные принципы и подходы к программированию в среде, возможности среды и составляющие модули программы.

## Литература

1. Datasheet ATmega8/ATmega8L[electron source]/ Atmel Corporation / – electron text data – USA, San Jose, 2006 – access mode:  
[http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8\\_1\\_datasheet.pdf](http://www.atmel.com/images/atmel-2486-8-bit-avr-microcontroller-atmega8_1_datasheet.pdf) , free – language – Eng.
2. В. Н. Баранов Применение микроконтроллеров AVR: схемы, алгоритмы, программы, 2-е издание. – М.: Издательство Додэка XXI. 2006.
3. Мысловский Э., Власов А., Акристиный М. КРАТКИЙ ОБЗОР ПОПУЛЯРНЫХ СЕМЕЙСТВ СОВРЕМЕННЫХ МИКРОКОНТРОЛЛЕРОВ// Электронные компоненты. 2002. № 5. С. 47-50.
4. Мысловский Э.В., Власов А.И., Кузнецов А.С. ЦИФРОВЫЕ СИГНАЛЬНЫЕ ПРОЦЕССОРЫ С ФИКСИРОВАННОЙ ТОЧКОЙ СЕМЕЙСТВА ADSP21XX: Учебное пособие – Москва. Изд-во МГТУ им.Н.Э. Баумана. 2003. Том 2. 64 с.
5. Мысловский Э.В., Власов А.И., Меньшов К.А. ЦИФРОВЫЕ СИГНАЛЬНЫЕ ПРОЦЕССОРЫ С ПЛАВАЮЩЕЙ ТОЧКОЙ СЕМЕЙСТВА ADSP2106X: Учебное пособие – Москва. Изд-во МГТУ им. Н. Э. Баумана. 2003. Часть 3. 64 с.
6. Тимошкин А.Г., Власов А.И. О СТРАТЕГИИ И ТАКТИКЕ МАРКЕТИНГОВОЙ ПОЛИТИКИ МНОГОПРОФИЛЬНОЙ КОМПЬЮТЕРНОЙ ФИРМЫ // Приборы и системы. Управление, контроль, диагностика. 1996. № 9. С. 59-61.