Программирование микроконтроллеров

План

Введение

Раздел 1. Назначение и область применения, их архитектура

Раздел 2. Среды программирования. Схемы подключения микроконтроллера

Раздел 3. Практическая реализация программы на микроконтроллере

Вывод

Список использованных источников

# Введение

**Актуальность темы.** Микроконтроллеры используются во всех сферах жизнедеятельности человека, устройствах, которые окружают его. Простота подключения и большие функциональные возможности. С помощью программирования микроконтроллера можно решить многие практические задачи аппаратной техники.

**Цель работы**. На основе практического примера показать преимущественные характеристики использования микроконтроллеров, необходимости их внедрения в различные устройства.

Можно считать что микроконтроллер (МК) - это компьютер, разместившийся в одной микросхеме. Отсюда и его основные привлекательные качества: малые габариты; высокие производительность, надежность и способность быть адаптированным для выполнения самых различных задач.

Микроконтроллер помимо центрального процессора (ЦП) содержит память и многочисленные устройства ввода/вывода: аналого-цифровые преобразователи, последовательные и параллельные каналы передачи информации, таймеры реального времени, широтно-импульсные модуляторы (ШИМ), генераторы программируемых импульсов и т.д. Его основное назначение - использование в системах автоматического управления, встроенных в самые различные устройства: кредитные карточки, фотоаппараты, сотовые телефоны, музыкальные центры, телевизоры, видеомагнитофоны и видеокамеры, стиральные машины, микроволновые печи, системы охранной сигнализации, системы зажигания бензиновых двигателей, электроприводы локомотивов, ядерные реакторы и многое, многое другое. Встраиваемые системы управления стали настолько массовым явлением, что фактически сформировалась новая отрасль экономики, получившая название Embedded Systems (встраиваемые системы).

Достаточно широкое распространение имеют МК фирмы ATMEL, которые располагают большими функциональными возможностями.

Применение МК можно разделить на два этапа: первый - программирование, когда пользователь разрабатывает программу и прошивает ее непосредственно в кристалл, и второй - согласование спроектированных исполнительных устройств с запрограммируемым МК. Значительно облегчают отладку программы на первом этапе - симулятор, который наглядно моделирует работу микропроцессора. На втором этапе для отладки используется внутрисхемный эмулятор, который является сложным и дорогим устройством, зачастую недоступным рядовому пользователю.

В тоже время в литературе мало уделено внимания вопросам обучения программированию некоторых недорогих МК, в сочетании с реальными исполнительными устройствами.

Разработка макета программатора отличающегося простотой, наглядностью и низкой себестоимостью, становиться необходимой как для самого программирования кристаллов, так и для наглядного обучения широкого круга пользователей основам программирования МК.

# Раздел 1. Назначение и область применения, их архитектура

Микроконтроллер - компьютер на одной микросхеме. Предназначен для управления различными электронными устройствами и осуществления взаимодействия между ними в соответствии с заложенной в микроконтроллер программой. В отличие от микропроцессоров, используемых в персональных компьютерах, микроконтроллеры содержат встроенные дополнительные устройства. Эти устройства выполняют свои задачи под управлением микропроцессорного ядра микроконтроллера.

К наиболее распространенным встроенным устройствам относятся устройства памяти и порты ввода/вывода (I/O), интерфейсы связи, таймеры, системные часы. Устройства памяти включают оперативную память (RAM), постоянные запоминающие устройства (ROM), перепрограммируемую ROM (EPROM), электрически перепрограммируемую ROM (EEPROM). Таймеры включают и часы реального времени, и таймеры прерываний. Средства I/O включают последовательные порты связи, параллельные порты (I/O линии), аналого-цифровые преобразователи (A/D), цифроаналоговые преобразователи (D/A), драйверы жидкокристаллического дисплея (LCD) или драйверы вакуумного флуоресцентного дисплея (VFD). Встроенные устройства обладают повышенной надежностью, поскольку они не требуют никаких внешних электрических цепей.

В отличие от микроконтроллера контроллером обычно называют плату, построенную на основе микроконтроллера, но достаточно часто при использовании понятия "микроконтроллер" применяют сокращенное название этого устройства, отбрасывая приставку "микро" для простоты. Также при упоминании микроконтроллеров можно встретить слова "чип" или "микрочип", "кристалл" (большинство микроконтроллеров изготавливают на едином кристалле кремния), сокращения МК или от английского microcontroller - MC.

микроконтроллер программа микросхема электронный

Микроконтроллеры можно встретить в огромном количестве современных промышленных и бытовых приборов: станках, автомобилях, телефонах, телевизорах, холодильниках, стиральных машинах. и даже кофеварках. Среди производителей микроконтроллеров можно назвать Intel, Motorola, Hitachi, Microchip, Atmel, Philips, Texas Instruments, Infineon Technologies (бывшая Siemens Semiconductor Group) и многих других. Для производства современных микросхем требуются сверхчистые помещения.

Основным классификационным признаком микроконтроллеров является разрядность данных, обрабатываемых арифметико-логическим устройством (АЛУ). По этому признаку они делятся на 4-, 8-, 16-, 32 - и 64-разрядные. Сегодня наибольшая доля мирового рынка микроконтроллеров принадлежит восьмиразрядным устройствам (около 50 % в стоимостном выражении). За ними следуют 16-разрядные и DSP-микроконтроллеры (DSP - Digital Signal Processor - цифровой сигнальный процессор), ориентированные на использование в системах обработки сигналов (каждая из групп занимает примерно по 20 % рынка). Внутри каждой группы микроконтроллеры делятся на CISC - и RISC-устройства. Наиболее многочисленной группой являются CISC-микроконтроллеры, но в последние годы среди новых чипов наметилась явная тенденция роста доли RISC-архитектуры.

Тактовая частота, или, более точно, скорость шины, определяет, сколько вычислений может быть выполнено за единицу времени. В основном производительность микроконтроллера и потребляемая им мощность увеличиваются с повышением тактовой частоты. Производительность микроконтроллера измеряют в MIPS (Million Instruсtions per Second - миллион инструкций в секунду).

Термин контроллер образовался от английского слова to control - управлять. Эти устройства могут основываться на различных принципах работы от механических или оптических устройств до электронных аналоговых или цифровых устройств. Механические устройства управления обладают низкой надежностью и высокой стоимостью по сравнению с электронными блоками управления, поэтому в дальнейшем мы такие устройства рассматривать не будем. Электронные аналоговые устройства требуют постоянной регулировки в процессе эксплуатации, что увеличивает стоимость их эксплуатации. Поэтому такие устройства к настоящему времени почти не используются. Наиболее распространенными на сегодняшний день схемами управления являются схемы, построенные на основе цифровых микросхем.

В зависимости от стоимости и габаритов устройства, которым требуется управлять, определяются и требования к контроллеру. Если объект управления занимает десятки метров по площади, как, например, автоматические телефонные станции, базовые станции сотовых систем связи или радиорелейные линии связи, то в качестве контроллеров можно использовать универсальные компьютеры. Управление при этом можно осуществлять через встроенные порты компьютера (LPT, COM, USB или Ethernet). В такие компьютеры при включении питания заносится управляющая программа, которая и превращает универсальный компьютер в контроллер.

Использование универсального компьютера в качестве контроллера позволяет в кратчайшие сроки производить разработку новых систем связи, легко их модернизировать (путём простой смены программы) а также использовать готовые массовые (а значит дешёвые) блоки.

Если же к контроллеру предъявляются особенные требования, такие, как работа в условиях тряски, расширенном диапазоне температур, воздействия агрессивных сред, то приходится использовать промышленные варианты универсальных компьютеров. Естественно, что эти компьютеры значительно дороже обычных универсальных компьютеров, но всё равно они позволяют экономить время разработки системы, за счёт того, что не нужно вести разработку аппаратуры контроллера.

Контроллеры требуются не только для больших систем, но и для малогабаритных устройств, таких как радиоприёмники, радиостанции, магнитофоны или сотовые аппараты. В таких устройствах к контроллерам предъявляются жёсткие требования по стоимости, габаритам и температурному диапазону работы. Этим требованиям не могут удовлетворить даже промышленные варианты универсального компьютера. Приходится вести разработку контроллеров на основе однокристальных ЭВМ, которые в свою очередь получили название микроконтроллеры. Любые устройства, в том числе и устройства связи, радиоавтоматики или аудиовизуальной аппаратуры требуют присутствия в своем составе устройства управления (контроллера). Контроллеры требуются практически во всех предметах и устройствах, которые окружают нас. Наиболее распространёнными в настоящее время являются микроконтроллеры семейства MCS-51. Это семейство поддерживается рядом фирм - производителей микросхем. Не менее распространёнными в мире являются микроконтроллеры фирмы Motorola. Это такие семейства как HC05, HC07, HC11 и многие другие. Пожалуй, не менее популярными микроконтроллерами являются микроконтроллеры семейства AVR фирмы Atmel. Если представить все типы современных микроконтроллеров (МК), то можно поразиться огромным количеством разнообразных приборов этого класса, доступных потребителю. Однако все эти приоры можно разделить на следующие основные типы: встраиваемые (embedded) 8-разрядные МК; 16 - и 32-разрядные МК; цифровые сигнальные процессоры. Промышленностью выпускаются очень широкая номенклатура встраиваемых МК. В них все необходимые ресурсы (память, устройства ввода-вывода и т.д.) располагаются на одном кристалле с процессорным ядром [3]. Если подать питание и тактовые импульсы на соответствующие входы МК, то можно сказать, что он как бы "оживет" и с ним можно будет работать. Обычно МК содержат значительное число вспомогательных устройств, благодаря чему обеспечивается их включение в реальную систему с использованием минимального количества дополнительных компонентов. В состав этих МК входят:

* Схема начального запуска процессора (Reset);
* Генератор тактовых импульсов;
* Центральный процессор;
* Память программ (E (E) PROM) и программный интерфейс;
* Средства ввода/вывода данных;
* Таймеры, фиксирующие число командных циклов.

Общая структура МК показана на (Рис.1.1.) Эта структура дает представление о том, как МК связывается с внешним миром. Более сложные встраиваемые МК могут дополнительно реализовывать следующие возможности:

* Встроенный монитор/отладчик программ;
* Внутренние средства программирования памяти программ (ROM);
* Обработка прерываний от различных источников;
* Аналоговый ввод/вывод;
* Последовательный ввод/вывод (синхронный и асинхронный);
* Параллельный ввод/вывод (включая интерфейс с компьютером);
* Подключение внешней памяти (микропроцессорный режим).

Все эти возможности значительно увеличивают гибкость применения МК и делают более простым процесс разработки систем на и основе.



***Рис.1.1*** *Структура микроконтроллера.*

Некоторые МК (особенно 16 - и 32-разрядные) используют только внешнюю память, которая включает в себя как память программ (ROM), так и некоторый объем памяти данных (RAM), требуемый для данного применения. Они применяются в системах, где требуется большой объем памяти и относительное не большое количество устройств (портов) ввода/вывода. Типичным примером применения такого МК с внешней памятью является котроллер жесткого диска (HDD) с буферной кэш-памятью, который обеспечивает промежуточное хранение и распределение больших объемов данных (порядка нескольких мегабайт). Внешняя память дает возможность такому микроконтроллеру работать с более высокой скоростью, чем встраиваемый МК.

Цифровые сигнальные процессоры (DSP) - относительно новая категория процессоров. Назначение DSP состоит в том, чтобы получать текущие данные от аналоговой системы, обрабатывать данные и формировать соответствующий отклик в реальном масштабе времени. Они обычно входят в состав систем, используясь в качестве устройств управления внешним оборудованием, и не предназначены для автономного применения.

### *Система команд.*

В зависимости от числа используемых кодов операций системы команд МК можно разделить на две группы: CISC и RISC. Термин CISC означает сложную систему команд и является аббревиатурой английского определения Complex Instruction Set Computer. Аналогично термин RISC означает сокращенную систему команд и происходит от английского Reduce Instruction Set Computer. Систему команд МК 8051 можно отнести к типу CISC. Однако, не смотря на широкую распространенность этих понятий, необходимо признать, что сами названия не отражают главного различия между системами команд CISC и RISC. Основная идея RISC архитектуры - это тщательный подбор таких комбинаций кодов операций, которые можно было бы выполнить за один такт тактового генератора. Основной выигрыш от такого подхода - резкое упрощение аппаратной реализации ЦП и возможность значительно повысить его производительность.

Очевидно, что в общем случае одной команде CISC соответствует несколько команд RISC. Однако обычно выигрыш от повышения быстродействия в рамках RISC перекрывает потери от менее эффективной системы команд, что приводит к более высокой эффективности RISC систем в целом по сравнению с CISC.

Однако в настоящее время грань между CISC и RISC архитектурой стремительно стирается. Например, МК семейства AVR фирмы Atmel имеют систему команд из 120 инструкций, что соответствует типу CISC. Однако большинство из них выполняется за один такт, что является признаком RISC архитектуры. Сегодня принято считать, что признаком RISC архитектуры является выполнение команд за один такт тактового генератора. Число команд само по себе значения уже не имеет.

### *Типы памяти МК.*

Можно выделить три основных вида памяти, используемой в МК:

а) *память программ*;

б) *память данных*;

в) *регистры МК*.

*Память программ* представляет собой постоянную память, предназначенную для хранения программного кода и констант. Эта память не изменяет содержимого в процессе выполнения программы. *Память данных* предназначена для хранения переменных в ходе выполнения программы*. Регистры МК* - этот вид памяти включает внутренние регистры процессора и регистры, которые служат для управления периферийными устройствами.

Для хранения программ обычно служит один из видов постоянной памяти: ROM (масочные ПЗУ), PROM (однократно программируемые ПЗУ), EPROM (электрически программируемые ПЗУ с ультрафиолетовым стиранием) или EEPROM (ПЗУ с электрической записью и стиранием, к этому виду также относятся современные микросхемы Flash-памяти). Все эти виды памяти являются энергонезависимыми - это означает, что содержимое памяти сохраняется после выключения питания МК.

Многократно программируемые ПЗУ - EPROM и EEPROM (Electrically Erasable Programmable Memory) подразделяются на ПЗУ со стиранием ультрафиолетовым (УФ) облучением (выпускаются в корпусах с окном), и МК с электрически перепрограммируемой памятью, соответственно.

В настоящее время протоколы программирования современной EEPROM памяти позволяют выполнять программирование МК непосредственно в составе системы, где он работает. Такой способ программирования получил название - ISP (In System Programming). И теперь можно периодически обновлять программное обеспечение МК без удаления из платы. Это дает огромный выигрыш на начальных этапах разработки систем на базе МК или в процессе их изучения, когда масса времени уходит на многократный поиск причин неработоспособности системы и выполнение последующих циклов стирания-программирования памяти программ.

Функционально Flash-память мало отличается от EEPROM. Основное различие состоит в способности стирания записанной информации. В памяти EEPROM стирание производится отдельно для каждой ячейки, а во Flash-памяти стирание осуществляется целыми блоками.

*ОЗУ* (RAM) - оперативное запоминающее устройство, используется для хранения данных. Эту память называют еще памятью данных. Число циклов чтения и записи в ОЗУ неограниченно, но при отключение питания вся информация теряется.

# Раздел 2. Среды программирования. Схемы подключения микроконтроллера

Программная среда "**AVR Studio"** - это мощный современный программный продукт, позволяющий производить все этапы разработки программ для любых микроконтроллеров серии AVR. Пакет включает в себя специализированный текстовый редактор для написания программ, мощный программный отладчик.

Кроме того, "AVR Studio" позволяет управлять целым рядом подключаемых к компьютеру внешних устройств, позволяющих выполнять аппаратную отладку, а также программирование ("прошивку") микросхем AVR.

Программная среда "AVR Studio" работает не просто с программами, а с проектами. Для каждого проекта должен быть отведен свой отдельный каталог на жестком диске. В AVR Studio одновременно может быть загружен только один проект.

При загрузке нового проекта предыдущий проект автоматически выгружается. Проект содержит всю информацию о разрабатываемой программе и применяемом микроконтроллере. Он состоит из целого набора файлов.

Главный из них - файл проекта. Он имеет расширение aps. Файл проекта содержит сведения о типе процессора, частоте тактового генератора и т.д. Он также содержит описание всех остальных файлов, входящих в проект. Все эти сведения используются при отладке и трансляции программы.

Кроме файла aps, проект должен содержать хотя бы один файл с текстом программы. Такой файл имеет расширение asm. Недостаточно просто поместить файл asm в директорию проекта. Его нужно еще включить в проект. Проект может содержать несколько файлов asm. При этом один из них является главным. Остальные могут вызываться из главного при помощи оператора. include. На этом заканчивается список файлов проекта, которые создаются при участии программиста.

**Algorithm Builder** предназначен для производства полного цикла разработки начиная от ввода алгоритма, включая процесс отладки и заканчивая программированием кристалла.

Algorithm Builder довольно нетрадиционная программа в плане языка программирования; ассемблер, завернутый в красивую обертку визуального программирования. Algorithm Builder - визуальный ассемблер или построитель ассемблера с помощью которого на выходе можно получить максимально эффективный код.

Работа с переменными и константами организована гениально просто. Инициализация производиться в отдельном окне в виде таблицы - освобождая алгоритм от лишних записей. В буквальном смысле слова все разложено по полочкам.

Algorithm Builder имеет удобный настройщик периферии (таймеры, UART, ADC, SPI и т.д.) позволяющий, не читая даташитов, просто выбрать необходимые параметры работы устройства в окне настройки. В этом же окне Builder честно покажет набор инструкций, обеспечивающих эти параметры.

Переходы осуществляются в программе очень наглядно - вектором. Если требуется перейти по условию в какую либо точку программы - нужно просто провести вектор в эту точку. Это освобождает программу от бесчисленных имен меток, которые в классическом ассемблере являются неизбежным балластом. Переходы по именованным меткам так же возможны.

Внутрисхемное программирование кристалла. При использовании внутрисхемного программатора микроконтроллер подключается к COM порту компьютера через несложный адаптер (три диода и несколько резисторов). Также есть вариант USB подключения. Программатор ведет подсчет числа перепрограммирований кристалла, сохраняя счетчик непосредственно в кристалле. Процесс программирования кристалла очень прост - в два "хода".

Мониторная отладка на кристалле. Algorithm Builder обеспечивает мониторную отладку на кристалле (On Chip debug) которая позволяет наблюдать содержимое реального кристалла в заданной точке останова. При этом для связи микроконтроллера с компьютером используется только один вывод, причем по выбору пользователя. Мониторная отладка может быть применена к любому типу кристалла, имеющего SRAM. Это софтверный вариант debugWIRE.

Для того, чтобы написанная программа превратилась в результирующий код и заработала в конкретном микропроцессорном устройстве, ее нужно оттранслировать и "зашить" в программную память микроконтроллера.

При написании программ обычно нельзя обойтись без процедуры отладки. Отладка выполняется на компьютере при помощи специальной инструментальной программы - отладчика. Он позволяет пошагово выполнять отлаживаемую программу, а также выполняет ее поэтапно с использованием, так называемых точек останова.

В процессе выполнения программы под управлением отладчика можно на экране компьютера:

1. видеть содержимое любого регистра микроконтроллера;
2. видеть содержимое ОЗУ и EEPROM;
3. наблюдать за последовательностью выполнения команд, контролируя правильность отработки условных и безусловных переходов;
4. наблюдать за работой таймеров, отработкой прерываний.

В процессе отладки также можно наблюдать логические уровни на любом внешнем выходе микроконтроллера. А также имитировать изменение сигналов на любом входе. Процесс отладки позволяет убедиться в том, что разрабатываемая программа работает именно так, как нужно.

Существует три основных вида отладчиков:

* программные;
* аппаратные;
* комбинированные программно-аппаратные.

*Программный отладчик*

Программный отладчик - это компьютерная программа, которая имитирует работу процессора на экране компьютера. Она не требует наличие реальной микросхемы или дополнительных внешних устройств и позволяет отладить программу чисто виртуально.

Однако программный отладчик позволяет проверить только логику работы программы. При помощи такого отладчика невозможно проверить работу схемы в режиме реального времени или работу всего микропроцессорного устройства в комплексе, т. е невозможно гарантировать правильную работу и всех подключенных к микроконтроллеру дополнительных микросхем и элементов.

*Аппаратный отладчик*

Основа аппаратного отладчика - специальная плата, подключаемая к компьютеру, работающая под его управлением и имитирующая работу реальной микросхемы микроконтроллера. Плата имеет выводы, соответствующие выводам реальной микросхемы, на которых в процессе отладки появляются реальные сигналы.

При помощи этих выводов отладочная плата может быть включена в реальную схему. Возникающие в процессе отладки электрические сигналы можно наблюдать при помощи осциллографа. Можно нажимать реальные кнопки и наблюдать работу светодиодов и других индикаторов.

Здесь как и в предыдущем случае, можем видеть всю информации об отлаживаемой программе: наблюдать содержимое регистров, ОЗУ, портов ввода-вывода; контролировать ход выполнения программы.

В аппаратном отладчике так же, как и в программном, можно выполнять программу в пошаговом режиме и применять точки останова. Недостатком аппаратного отладчика является его высокая стоимость.

*Имитаторы*

Имитаторы - программы, которые позволяют на экране компьютера "собрать" любую электронную схему, включающую в себя самые разные электронные компоненты:

a) транзисторы;

b) резисторы;

c) конденсаторы;

d) операционные усилители;

e) логические и цифровые микросхемы, в том числе и микроконтроллеры.

Такие программы обычно содержат обширные базы электронных компонентов и конструктор электронных схем. Собрав схему, можно виртуально записать в память микроконтроллера вашу программу, а затем "запустить" всю схему в работу.

Для контроля результатов работы схемы имитатор имеет виртуальные вольтметры, амперметры и осциллографы, которые можно "подключать" к любой точке схемы, "измерять" различные напряжения, а также "снимать" временные диаграммы.

Такие программы в настоящее время получают все большее распространение. Они позволяют разработать любую схему с микроконтроллером или без него, без использования паяльника и реальных деталей. На экране компьютера можно полностью отладить свою схему и лишь потом браться за паяльник.

Недостатком данного отладчика является то, что он требует значительных вычислительных ресурсов. Особенно в том случае, когда отлаживается схема, включающая как микроконтроллер, так и некоторую аналоговую часть. Кроме того, имитатор не всегда верно имитирует работу некоторых устройств. Однако подобные программы имеют очень большие перспективы.

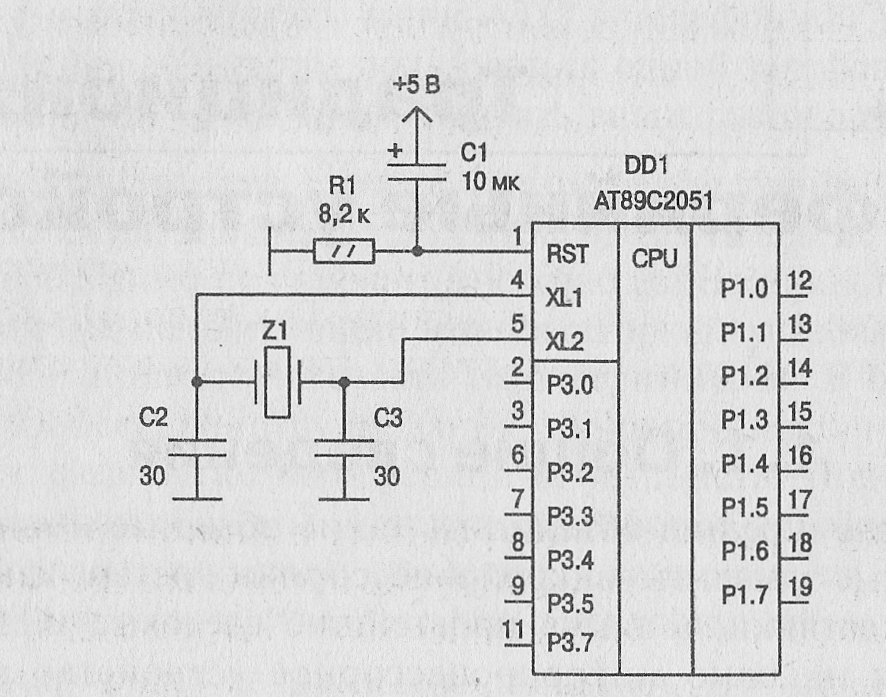
**WinAVR**

WinAVR представляет собой набор инструментальных средств для работы с микроконтроллерами семейства AVR фирмы ATMEL. В него вошли следующие компоненты:

1. компилятор языка C avr-gcc,
2. библиотека компилятора avr-libs,
3. ассемблер avr-as,
4. интерфейс программатора avrdude,
5. интерфейс JTAG ICE avarice,
6. Debuger avr-gdb,
7. редактор programmers notepad.

Весь этот набор собран в один инсталляционный пакет и предназначен для установки на платформу Windows.

Главным преимуществом моего выбора именно этой микросхемы является ее широкая доступность и не высокая цена.

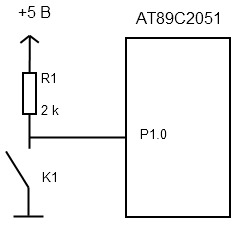


***Рис.2.1*** *Типовая схема включения микроконтроллера АТ89С2051*

Элементы R1, С1 составляют цепь начального сброса микроконтроллера. Она служит для перевода в исходное состояние всех внутренних систем микроконтроллера сразу после включения питания. Кварцевый резонатор Z1 определяет частоту встроенного тактового генератора микроконтроллера. Этот генератор предназначен для синхронизации всех внутренних процессов микроконтроллера. Микросхема АТ89С2051 допускает выбирать частоту кварцевого резонатора до 24 МГц. Нижний предел частоты не ограничивается. Конденсаторы С2 и СЗ - это согласующие элементы для кварца. Микроконтроллер АТ89С2051 допускает применение в качестве времязадающей цепи резонансного контура, и даже подключение внешнего тактового генератора. Оставшиеся выводы микроконтроллера представляют собой два порта ввода/вывода, которые обозначены Р1 и РЗ. Именно к этим двум портам и подключаются периферийные устройства.

Практически ни одно микропроцессорное устройство не обходится без кнопок и простейших датчиков на основе обычных контактов. При помощи этого вида периферийных элементов в микропроцессорное устройство поступает различная информация, которая используется для изменения алгоритма работы программы.

Примером может служить датчик поворота (Рис.2.2) - механические контакты, связанные с поворачиваемым устройством.



***Рис.2.2*** *Простая схема подключения датчика на основе геркона*

На вход микроконтроллера через резистор R1 подается напряжение от источника питания +5 В. Микросхема воспринимает это напряжение как сигнал логической единицы. При срабатывании датчика контакты замыкаются и соединяют вывод микроконтроллера с общим проводом. В результате напряжение на входе Р1.0 падает до нуля. В следствии микросхема воспринимает входной уровень сигнала как логический ноль. Резистор R1 при этом служит токоограничивающим элементом, предотвращая короткое замыкание между шиной питания и общим проводом.

Рассмотрим программу для обслуживания вышеупомянутого датчика (*Листинг 2.3.),* которая, постоянно опрашивает датчик и в зависимости от состояния запускает одну из двух специальных процедур.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | ; Программа обработки сигнала с датчика  m1: mov p1.0,#1 ; Записываем 1 в соответствующий разряд порта  mov c,p1.0 ; Читаем состояние датчика в битовый аккумулятор  jc m2 ; Если контакты датчика разомкнуты, перейти к m2  call proc1 ; Вызов процедуры обработки нажатия контакта  jmp m1 ; Возврат к началу (следующий цикл считывания)  m2: call proc1 ; Вызов процедуры обработки размыкания контакта  jmp m1 ; Возврат к началу (следующий цикл считывания) |

***Листинг 2.3.***

Здесь явно видно, что программа записывает в линию Р1.0 сигнал логической единицы (строка 1). Это необходимо для того, чтобы данная линия могла работать на ввод информации.

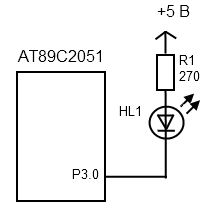
Следующая команда считывает бит информации, поступающей от датчика, и помещает ее в регистр признака переноса (строка 2). В микропроцессорной технике принято ячейку признака переноса обозначать как **CY**. Ячейка **CY** используется как аккумулятор для битовых операций. Если в момент считывания сигнала контакты датчика были разомкнуты, то в ячейке **CY** окажется логическая единица. Если контакты замкнуты, то там будет логический ноль.

Оператор условного перехода **jc** осуществляет оценку содержимого **CY** (строка 3). Если в **CY** логический ноль, то управление передается на метку **m2,** и выполняется команда **call ргос2** (строка 6). В противном случае передача управления не происходит и выполняется команда **call proc1** (строка 4). Оператор **call** - это вызов подпрограммы. Поэтому, в зависимости от состояния датчика вызывается одна из двух подпрограмм: **proc1** или **ргос2.**

# Раздел 3. Практическая реализация программы на микроконтроллере

**Задача.** Построить схему на микроконтроллере, в моем случае это микроконтроллер АТ89С2051 фирмы Atmel,который в зависимости от положения 4 переключателей зажигал бы 6 светодиодов (лампочек).

**Решение:** для реализации построил простейшую схему светодиодного индикатора (*Рис.3.1.),* которое используется не только в моем примере, а и в любом устройстве в качестве светового индикатора.



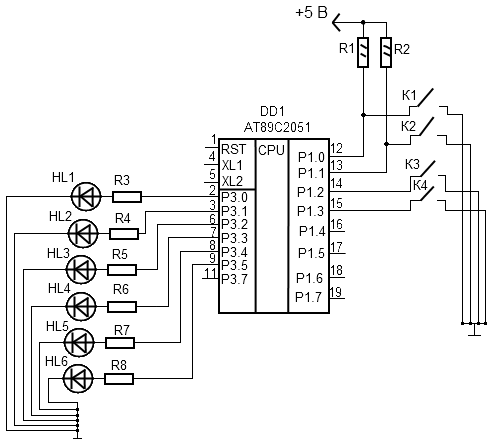
***Рис.3.1*** *Простейшая схема светодиодного индикатора*

Использовав схемы вышеупомянутых схем (*Рис.2.1.,2.2.,3.1.),* построил схему согласно поставленной задачи. Код программы написан на языке ассемблера для микроконтроллера АТ89С2051 (Cross-Assembler 8051, Version 1.2h).

Как видно из схемы порт Р1 работает как порт ввода. Шесть младших P3 работают на вывод. Возможен и обратный вариант (Р1 на вывод, Р3 на ввод).

Я же выбрал тот вариант, который наиболее рационален с точки зрения удобства составления программы.

Рассмотрим принцип работы построенной схемы.



***Рис.4.5*** *Схема светодиодной индикации с переключателями*

В исходном состоянии на выводы Р3.0…Р3.5 подается сигнал логической единицы. На все выводы порта Р1 (Р1.0…Р1.3) также поданы единицы. Но во втором случае единицы поданы для того, чтобы обеспечить возможность работы линий в режиме ввода. Контроллер периодически опрашивает состояние клавиш путем изменения сигнала на выходах Р3.0…Р3.5 и считывания сигнала из порта Р1. В случае обнаружения замыкания контакта одной из клавиш, программа выполняет закрепленные за этой клавишей действия.

Каким же образом осуществляется опрос клавиш? Процедура опроса клавиш поочередно переводит одну из линий Р3.0…Р3.5 в нулевое состояние. Сначала в нулевое состояние переводится линия Р3.5 Сразу после этого контроллер производит чтение числа из порта Р1. Если ни одна из клавиш не нажата, то все разряды считанного числа будут равны единице (считанное число будет равно 0FFH). Если хотя бы одна из клавиш К1…К4 окажется нажатой, то число, прочитанное из порта Р1, будет отличаться от значения 0FFH. Предположим, что нажали клавишу К1. Тогда сигнал логического нуля с выхода Р3.0 поступит на вход Р1.0 и младший разряд считанного числа окажется равным нулю. В этом случае процессор из порта прочитает 0FЕH. Нажатие любых других клавиш К2. К4 приведет к обнулению других разрядов считываемого числа. В результате для разных комбинаций получим разные коды.

4 выключателя 6 светодиодов

K1 K2 K3 K4 HL1 HL2 HL3 HL4 HL5 HL6

0 0 0 0 0 0 0 0 0 0

1 0 0 0 1 0 0 0 0 0

0 1 0 0 0 1 0 0 0 0

0 0 1 0 0 0 1 0 0 0

0 0 0 1 0 0 0 1 0 0

1 1 1 1 1 1 1 1 1 1

1 0 0 1 1 0 0 0 1 1

1 0 1 0 1 0 1 0 1 1

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | $mod 2051  ;----------------------Определение констант  bank0 EQU 000000000B ; Коды банков памяти  bank1 EQU 00001000B  bank2 EQU 00010000B  bank3 EQU 00011000B  ;-----------------Резервирование ячеек памяти  DSEG  ORG 20H ;Начинаем резервирование с адреса 20Н  p3buf: DS 1 ;Буфер порта р3  ;-----------Начало программного кода  CSEG  ORG 00H ;Начинаем программный код с адреса 00Н  ;----------------------------------------------------------------  ; В этом месте вы должны поместить основной текст вашей программы  ;-----------------------------------------------------------------  ;###########################################  ;## Подпрограмма опроса клавиш  ;###########################################  klav: push psw ;Сохранение регистра флагов в стеке  mov psw,#bank3 ;Переключение на Банк 3 регистров Р0Н  mov p1,#0FFH ;Перевод Р1.0-Р1.3 в единичное состояние  call clrU ;Перевод Р3.0-Р3.5 в единичное состояние  kl1: call setU ;Вывод кода в порт Р3  mov a,p1 ;Считывание состояния клавиш  cjne a,#0FFH,kl3 ;Если клавиша нажата, переходим к kl3  ;----------------------Окончание процедуры опроса клавиатуры  klfin call clrU ;Переход в исходное состояние порта  mov a,r0 ;Запись в аккумулятор кода клавиши  pop psw ;Восстановление регистра флагов  ret ;Выход из подпрограммы  ;--------------------Вывод столбцов  setU mov a,p3buf ;Считывание содержимого буфера порта Р3  orl a,#0FH ;Перевод разрядов Р3.0-Р3.5 в единицу  anl a,r2 ;Вывод Р3.0-Р3.5 кода сканирования  mov p3buf,a ;Запись результата назад в буфер  mov p3,p3buf ;Вывод содержимого буфера в порт Р3  ret  ;-----------------------------------------------------------------  ;Сюда вы можете поместить другие подпрограммы  ;-----------------------------------------------------------------  end |

# Вывод

Рассмотренная выше практическая задача дает четкое представление о значимости представленного устройства. Трудно представить сферу современной деятельности человека, где не использовались бы данные устройства. Телефоны, телевизоры, жидкокристаллические мониторы, кондиционеры, холодильники, новогодние гирлянды, компьютеры и многое другое не могут работать без микроконтроллеров. Микроконтроллеры намного лучше своих предшественников: ламп и полупроводников. Они намного меньших размеров и обладают большей производительностью.

В процессе выполнения курсовой работы были рассмотрены практические примеры подключения микроконтроллеров и их программирование.

Применение микроконтроллеров в технике очень актуально. Так как они существенно ускоряют работу поставленной им задачи. Отсюда и важность их изучения и применения в устройствах.

# Список использованных источников

1. Белов А.Б. Конструирование устройств на микроконтроллерах / Наука и Техника, 2005. - 255 с.
2. Предко М. Руководство по микроконтроллерам. Том 1. / Пер. с англ. под ред.И. И. Шагурина и С.Б. Лужанского - М.: Постмаркет, 2001. - 416 с.
3. Предко М. Руководство по микроконтроллерам. Том 2. / Пер. с англ. под ред.И. И. Шагурина и С.Б. Лужанского - М.: Постмаркет, 2001. - 488 с.
4. Вуд А. Микропроцессоры в вопросах и ответах. / Пер. с англ. под ред. Д.А. Поспелова. - М.: Энергоатомиздат. 1985. - 184 с.
5. Уильямс Г.Б. Отладка микропроцессорных систем: / Пер. с. англ. - М.: Энергоатомиздат, 1988. - 253с.
6. Угрюмов Е.П. Цифровая схемотехника. - Спб.: БВХ - Санкт-Петербург, 2000. - 528 с.
7. Алексенко А.Г., Шагурин И.И. Микросхемотехника. - М.: Радио и связь, 1990. - 496 с.
8. Бродин Б.В., Шагурин И.И. Микроконтроллеры: Справочник. - М.: ЭКОМ, 1999. - 395 с.
9. Программируемые логические ИМС на КМОП-структурах и их применение. / П.П. Мальцев, Н.И. Гарбузов, А.П. Шарапов, А.А. Кнышев. - М.: Энергоатомиздат, 1998. - 158 с.
10. Соловьев В.В., Васильев А.Г. Программируемые логические интегральные схемы и их применение. - Мн.: Беларуская наука, 1998. - 270 с.
11. Лаптев В. Цифровой измеритель температуры на базе AVR микроконтроллера и RC-цепочки. - Электронные компоненты, 2001. №2, с.46 - 49.