Министерство науки и образования РФ

РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ

ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

(РГГМУ)

Факультет ОКЕАНОЛОГИЧЕСКИЙ

Кафедра Морских информационных технологий

К защите допустить

Зав. Кафедрой профессор / П. П. Бескид /

Пояснительная записка

К ДИПЛОМНОЙ РАБОТЕ

на тему: Система охранной сигнализации промышленного объекта на базе разнотипных датчиков.

студент группы ОиБ – 605 /А. Ю. Сидоренко /

Руководитель: / Е. А. Чернецова/

Санкт – Петербург

2011

Содержание

Введение 3

1 Система охранной сигнализации на промышленном объекте: структура и спецификация 8

1.1 Алгоритм работы данной системы охранной сигнализации 12

1.2 Индикатор, мультиплексор, датчики: понятия, виды и характеристики 15

1.3 Микроконтроллер: его виды и характеристики 40

2 Монтаж системы сигнализации на охраняемом объекте для наилучшего функционирования 76

2.1 Планирование разводки кабеля и установка разнотипных датчиков 76

3 Разработка программного обеспечения для микроконтроллера 84

3.1 Выбор инструментальных средств (среда Ассемблер) 84

3.2 Подпрограмма для микроконтроллера 94

4 Результаты функционирование системы охранной сигнализации промышленного объекта 95

4.1 Пакет программного обеспечения для функционирования микроконтроллера 95

4.2 Экономическое обоснование 100

Заключение 103

Список используемых источников 104

Приложение А 105

Приложение Б 119

Введение

В последнее время в нашей стране, да и, пожалуй, во всём мире стала наиболее **актуальна проблема безопасности объекта**, а также проблема информационной безопасности. Системы электронных охранных сигнализаций являются одним из главных препятствий на пути несанкционированного проникновения на объект.

В связи с широким использованием современных электронных компонентов и цифровых методов обработки информации происходит существенная "интеллектуализация" технических средств охраны. Эти средства перестают быть просто вспомогательными и приобретают новые свойства, что оказывает существенное влияние на организацию охраны и её уровень. Современные технические средства охраны могут быть использованы в виде **полностью интегрированной системы** или системы, состоящей из функционально независимых компонентов:

– охранная сигнализация,

– система контроля доступа в помещение с проверкой полномочий,

– система телевизионного наблюдения,

– контрольные и другие устройства.

Тенденции развития электронной техники и электротехники на базе микроминиатюризации требуют применения широкой номенклатуры маломощных и малогабаритных устройств и изделий (преобразователей трансформаторов, усилителей, фильтров, стабилизаторов, выпрямителей и так далее), выполненных на новой конструктивной основе ЭРЭ. Достижения науки и техники на современном этапе развития в области электронной техники позволяют значительно уменьшить массогабаритные характеристики рассматриваемых электронных устройств. В настоящее время конструирование РЭА, РЭУ и ЭРЭ характеризуется резким увеличением применения БИС, что также даёт возможность уменьшить объёмы устройств и одновременно улучшить их **качественные характеристики, показатели надёжности и долговечности**.

Охранная сигнализация используется уже очень давно, и давно перестала быть чем-то экзотическим. Практически каждый второй магазин, офис, склад имеют охранную сигнализацию. Принцип действия охранной сигнализации очень прост. Инсталлятором (монтажной организацией) рассматриваются места возможного проникновения на объект и блокируются охранными датчиками (в этом плане наиболее уязвимыми с точки зрения безопасности являются окна и двери). В помещении охраны устанавливается прибор охранной сигнализации. В случае открытии двери, окна, разбитии стекла, несанкционированном проникновении в офис, срабатывает соответствующий датчик, и сигнал передаётся на прибор охранной сигнализации в помещении охраны. Включается звуковая и световая сигнализация, оповещая охрану о том, что на объект, в таком то месте кто-то проник. Наиболее распространёнными датчиками, используемыми в охранной сигнализации являются инфракрасные датчики движения, акустические датчики разбития стекла, герконы (магнито-контактные).

Приёмно-контрольные приборы, используемые в охранной сигнализации также представлены весьма широкой номенклатурой. От простейших, управляемых нажатием одной, двух кнопок, до компьютерных систем, где оператор может видеть на экране монитора план всего здания.

При выборе конкретных приёмно-контрольных приборов не следует забывать, что охранная сигнализация монтируется не просто для того, чтобы она была, а для облегчения работы охраны. Поэтому монтаж сложных в эксплуатации систем охранной сигнализации не всегда технически оправдан.

Охранные системы большой ёмкости обычно применяются для **обеспечения безопасности крупных предприятий**, банков, гостиниц и характеризуется тем, что обслуживают от нескольких сотен до нескольких тысяч датчиков (извещателей). При этом контролировать нужно не только датчики как средства обнаружения криминальной акции, но различного вида устройства управляющие работой системы. Один из самых главных вопросов. Куда будет приходить сигнал с приёмно-контрольного прибора. В случае наличия на объекте охраны этот вопрос решается просто, конечно в помещении охраны. А если охраны нет? Ведь далеко не каждая фирма способна оплачивать услуги охраны. Здесь существует несколько вариантов: 1) сдача объекта на пульт вневедомственной охраны; 2) сдача объекта на пульт охраны «Гольфстрим охранные системы» по радиоканалу; 3) применение автодозвонщика, который в случае сработки охранной сигнализации по заранее запрограммированным телефонным номерам передаст речевое сообщение о факте проникновения на объект; 4) применение GSM дозвонщика (для нетелефонизированных объектов).

Особенностью организации охраны рассматриваемого нами вида промышленных объектов является то, что непосредственно на территории объекта нет охранников. Задержание злоумышленников осуществляют оперативные группы специального подразделения милиции (вневедомственной охраны), либо какого-либо частного охранного агентства. Оперативные группы действуют  по указаниям оператора центрального пульта охраны. В этом случае функциональные возможности пульта определяют эффективность системы в целом, а сам пульт обычно реализуется на базе ПЭВМ с развитым программным обеспечением.

### *ОО N ОУ N СД N ОО 1 ОУ 1 Р 1 АТС 1 ПЦН ПЦО АТС N Р 1 СД 1* Структурная схема централизованной охраны приведена на рисунке 1.

Рисунок 1 – Структурная схема централизованной охраны

ОО – охраняемый объект.

ПЦО - **пункт централизованной охраны** - это диспетчерский пункт для централизованной охраны ряда рассредоточенных объектов от проникновения и пожара с использованием систем передачи извещений.

ПЦН - **пульт централизованного наблюдения**.

Р – **ретранслятор.**

ОУ - объектовое оконечное устройство - это составная часть системы, устанавливаемая на охраняемом объекте, для приёма извещений от совокупности датчиков (СД), преобразования сигналов и их передачи по каналу связи на ретранслятор ПЦН, а также (при наличии обратного канала) для приёма команд телеуправления от ретранслятора (ПЦН).

Технические средства от комплекса охранной сигнализации до пульта централизованной охраны образуют систему передачи извещений (СПИ), которая предусматривает установку оконечного устройства на объектах, ретрансляторах.

### Классификация систем передачи извещений.

**По информационной ёмкости** (количеству охраняемых объектов) СПИ подразделяют:

– на систему с постоянной информационной ёмкостью,

– с возможностью наращивать.

**По информативности** СПИ подразделяют на системы :

– малой информативной ёмкости - до 2-х видов извещений,

– средней информативной ёмкости - от 3-х до 5 видов извещений,

– большой информативной ёмкости - свыше 5 видов извещений.

**По типу используемых линий (каналов) связи** СПИ подразделяют на системы, использующие:

– телефонные сети,

– специальные линии связи,

– радиоканалы,

– комбинированные линии связи и др.

**По количеству направлений передачи информации** СПИ подразделяют на системы:

– с однонаправленной передачей информации,

– системы с двунаправленной передачей информации (использованием обратного канала).

**По способу отображения поступающей на ПЦН информации** СПИ подразделяют на системы :

– с индивидуальным или групповым отображением информации в виде световых и звуковых сигналов,

– с отображением информации на дисплеях,

– с применением условной обработки и накоплением в банке данных.

С охраняемых объектов могут передаваться следующие виды извещений:

– "проникновение",

– "неисправность",

– "взятие",

– "снятие",

адреса объектов и прочая служебная и диагностическая информация.

1. Система охранной сигнализации на промышленном объекте: структура и спецификация

Все виды охранных сигнализаций можно разделить на стационарные (устанавливаемые в каком-то определенном месте в помещении) и мобильные (которые можно переносить с места на место). Простейший пример стационарного устройства - тревожная кнопка. Вы нажали ее, и охрана в курсе, что на вас совершено нападение. Примером мобильного устройства может служить маленькая сирена, установленная под дверь.

Аналоговые охранные панели. Отличие этого оборудования, прежде всего в том, что оно контролирует положение шлейфа и различает состояние "норма", "тревога", "повреждение шлейфа". Эти особенности делают этот класс охранных панелей более защищенным, устойчивым к интеллектуальному взлому.

Цифровые охранные панели. Они защищают не только от повреждений линий связи, но и от подмены аналогичными устройствами. Опрос каждого шлейфа происходит 180 раз в секунду.

Радио охранные сигнализации - сигнализации, действующие с помощью радиоканала. Основное их преимущество - высокая мобильность, отсутствие строительно-монтажных работ, возможность использования при охране объектов, требующих минимального вмешательства.

Требования к технической укреплённости объекта должны определятся значимостью объекта, видом и концентрацией материальных или иных ценностей на нём, его строительными и архитектурно-планировочными решениями, режимом работы и многими другими факторами, которые необходимо учитывать при проектировании комплексной системы защиты объекта.

Таким образом, техническая укреплённость объекта – это совокупность мероприятий направленных на усиление конструктивных элементов зданий, сооружений, помещений и защищаемых территорий, обеспечивающих необходимое и достаточное противодействие несанкционированному проникновению нарушителя в защищаемую зону, взлому и другим преступным посягательствам.

Данная система охранной сигнализации должна обеспечить охрану по периметру объекта на предмет проникновения вовнутрь, а именно, проникновение через дверь или окно, также сигнализацию на движение, на пролом стен, пола, потолка. Система должна проверять и идентифицировать свое состояние и, в случае срабатывания, передать сигнал на пульт вневедомственной охраны (ВОХР) с указанием кода места нарушения.

Необходимо обеспечить простоту в обращении, надежность, низкую себестоимость.

Описания объекта, на которое возможно установка охранной сигнализации:

Характеристики объекта:

Кол-во помещений……………………..………………..до 3

Площадь одного помещения, м2 .………………..…. до 200

Высота, м2 .…………………………………..……..….….2-8

# Кол-во окон …………………….…………………..……до 6

# Кол-во входных дверей ………………………………....до 2

# Кол-во ворот ………….……………………………….…до 2

Функциональная спецификация:

1. Вход:

– сигналы извещателей движения;

– сигналы извещателей вибрации;

– сигналы оконных извещателей;

– сигналы контактных датчиков (герконов).

1. Выход:

– сигнализация на жидкокристаллический индикатор;

– сигнализация на пульт центрального наблюдения ВОХР.

1. Функции:

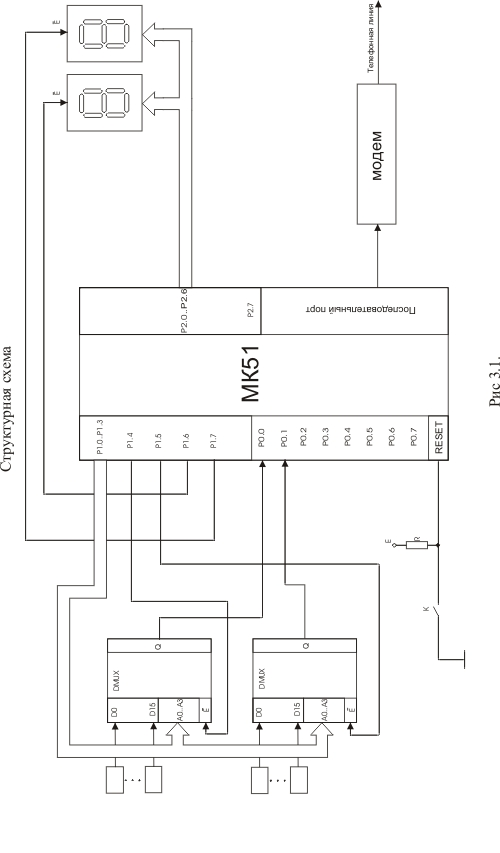
– периодический опрос и анализ герконов, вибрационных извещателей, извещателей движения, оконных извещателей;

– определение места срабатывания датчика;

– если при первом опросе обнаруживается срабатывание какого-либо извещателя или датчика, то передается сигнал на жидкокристаллический индикатор;

– если обнаруживается при втором и последующих опросах срабатывание извещателя или датчика, то передается сигнал на пульт центрального управления ВОХР.

Структурная схема системы охранной сигнализации приведена на рисунке 2.



Д31

Д0

Д16

Д15

MUX

MUX

Рисунок 2 – Структурная схема охранной системы сигнализации

Сокращения, которые использовались в схеме на рисунке 2.

Д0-Д31 –датчики, извещатели.

MUX – мультиплексор

МК51 – микроконтроллер семейства 51.

К - ключ–кнопка включения системы.

Используемые датчики.

Извещатель движения – пассивный, инфракрасный, с релейным выходом (цифровым). Кол-во: 16 шт.

Извещатель вибрации – наружный вибрационный, с релейным выходом (цифровым). Кол-во: 6 шт.

Извещатель оконный - охранный поверхностный звуковой, с релейным выходом (цифровым). Кол-во: 4 шт.

Геркон - магнитоуправляемый датчик для наружного монтажа с дискретным выходом. Кол-во: 6 шт.

Используемые мультиплексоры:

Два цифровой мультиплексор на шестнадцать информационных входов.

Используемый микроконтроллер:

Микроконтроллер семейства МК51.

Используемый ЖКИ:

Два семисегментных индикатора.

* 1. Алгоритм работы системы охранной сигнализации промышленного объекта

Постановка на сигнализацию осуществляется человеком, последним покидающим данный объект, путем нажатия кнопки «Запуск» находящемся на пульте управления системы. Сразу после запуска системы идет первый опрос датчиков. Если обнаруживается срабатывание датчика, то опрос прекращается; на ЖКИ выводится код сработавшего датчика. Это необходимо для защиты системы от ложных срабатываний. После чего должны приняться меры к устранению причин срабатывания датчиков. Далее осуществляется повторный пуск системы. Если же после первого опроса датчиков не произошло никакого срабатывания, то делается временная задержка в 1 минуту, для возможности покинуть человеком помещение, и система охранной сигнализации переходит в основной режим работы «охрана».

В режиме «охраны» ведется непрерывный опрос датчиков, и как только обнаруживается срабатывание, то передается сигнал на пульт центрального управления ВОХР с указанием кода сработавшего датчика. После передачи сигнала опрос продолжается, для того чтобы пресечь разные места проникновения.

Снятия объекта с сигнализации производиться путем прохода человека внутрь объекта (при этом происходит срабатывание сигнализации), отключение системы охраны от питания и звонком на пульт ВОХР с сообщением кода объекта и факта его снятия с сигнализации.

Алгоритм работы охранной системы сигнализации (Блок схема) приведен на рисунке 3.



#### Рисунок 3 – Алгоритм работы охранной системы сигнализации

1.2 Индикатор, мультиплексор, датчики: понятия, виды и характеристики

**Мультиплексор.**

Мультиплексор осуществляет подключение одного из входных каналов к выходному под управлением управляющего (адресующего) слова. Разрядности могут быть различными, но серийно выпускаются мультиплексоры двух-, четырех-, восьми- и шестнадцатиразрядные.

Входы мультиплексора делятся на две группы: информационные и адресующие. Адресующий код А задает переключателю определенное положение, соединяя с выходом Q один из информационных входов.

Вход разрешения работы является инверсным. Если на него подать 0, то выход мультиплексора будет равен 0 независимо от информационных и адресных сигналов. На рисунке 4 представлен вид мультиплексора на шестнадцать входов.



MUX

Рисунок 4 – Вид мультиплексора на шестнадцать информационных входов

# Примечание. Возможно наращивание разрядности мультиплексоров с помощью пирамидальной структуры из нескольких мультиплексоров. При этом первый ярус схемы представляет собой столбец, содержащий столько мультиплексоров, сколько необходимо для получения нужного числа информационных входов. Все мультиплексоры столбца адресуются одним и тем же кодом, составленным из соответствующего числа адресного кода (если число информационных входов схемы равно 2n, то общее число адресных разрядов равно n, младшее поле n1 адресного кода используется для адресации мультиплексоров первого яруса). Старшие разряы адресного кода, число которых равно n-n1 , используются во втором ярусе, мультиплексор которого обеспечивает поочередную работу мультиплексоров первого яруса на общий выходной канал.

**Индикаторы. Семисегментный жидкокристаллический индикатор.**

Для общения с операторов системы охранной сигнализации, то есть в нашем случае для индикации кода неисправного датчика используются средства индикации данных. Среди различных средств индикации имеются и сложные устройства, такие как экранные дисплеи, и простые, такие как светодиодные, жидкокристаллические индикаторы и матрицы. Мы используем только простейшие индикаторы – семисегментные жидкокристаллические индикаторы.

Семисегментный индикатор - устройство отображения цифровой информации. Это — наиболее простая реализация индикатора, который может отображать арабские цифры. Для отображения букв используются более сложные многосегментные и матричные индикаторы.

Семисегментный индикатор, как говорит его название, состоит из семи элементов индикации (сегментов), включающихся и выключающихся по отдельности. Включая их в разных комбинациях, из них можно составить упрощённые изображения арабских цифр. Часто семисегментные индикаторы делают в курсивном начертании, что повышает читаемость.

Цифры, 6, 7 и 9 имеют по два разных представления на семисегментном индикаторе.Сегменты обозначаются буквами от A до G; восьмой сегмент — десятичная запятая, предназначенная для отображения дробных чисел.Изредка на семисегментном индикаторе отображают буквы.

Светодиодные индикаторы имеют предельно простую форму, так как в них применяются светодиоды, отлитые в форме сегментов, и чем меньше разных типов светодиодов, тем дешевле устройство. В жидкокристаллических, газорязрядных, вакуумно-люминесцентных (катодно-люминесцентных) и других индикаторах дизайнеры находят место для вариации формы сегментов.

На рисунке 5 приведены различные виды индикаторов.

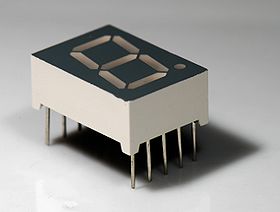
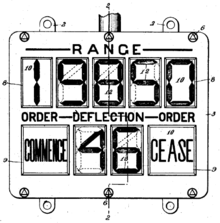


Рисунок 5 – Примеры семисегментных индикаторов

В индикаторе семь сегментов, которые расположены так, что при определенной их комбинации высвечивается та или иная цифра.

Коды сегментов индикатора изображены на рисунке 5.1.

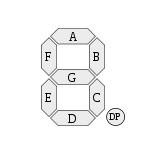


Рисунок 5.1 **­–** Коды сегментов индикатора

**Датчики.**

В стационарных охранных сигнализациях используется множество датчиков — от самых простых (контактных) до сложных, представляющих собой практически самостоятельные интеллектуальные электронные устройства (объемные датчики). Без датчиков не обходится ни одна охранная система.

Рассмотрим назначение и основные особенности датчиков, используемых в стационарных охранных системах, а так же в нашем проекте.

**Извещатели разбития стекла.**

Извещатели разбития стекла предназначены для защиты от проникновения путем разрушения стеклянного полотна окон или витрин. По принципу действия они делятся на три типа: контактные, пьезоэлектрические и акустические.

При применении контактных извещателей на поверхность стекла наносится слой проводящего материала (это может быть, например, фольга) и при его разрушении этот материал разрывает электрическую цепь, вследствие чего и принимается решение о наличии вторжения.

При применении пьезоэлектрических извещателей на поверхность стекла приклеиваются специальные небольшие пьезоэлектрические модули и при его разрушении генерируются акустические колебания, на основе которых и принимается решение о наличии вторжения (принцип их действия аналогичен звукоснимателю для виниловых дисков).

Разумеется, в последнее время извещатели рассмотренных выше двух типов стали применяться весьма редко, поскольку они существенно портят дизайн окна. От этого недостатка свободны только бесконтактные акустические извещатели, которые оснащаются высокочувствительным миниатюрным микрофоном, улавливающим звук, издаваемый при разбитии стекла. Эти извещатели крепятся на стену или потолок около окна. При разбитии стекла возникает два типа звуковых колебаний, причем в строго определенной последовательности. Сначала возникает ударная волна с частотой порядка 100 Гц (эту волну вызывает колебание при ударе всего массива стекла в целом), а потом волна разрушения стекла с частотой порядка 5 Кгц. Извещатель детектирует эти волны, и в случае наличия их правильной последовательности принимается решение о наличии проникновения.

Особенности:

– Акустический извещатель позволяет контролировать несколько окон одним прибором и не требует установки каких-либо элементов на контролируемом стекле.

– Возможность регулировки чувствительности.

– Контроль вскрытия корпуса.

– Режим тестирования

– Устойчивы к акустическим шумам (телефон, транспорт, гроза, град), электростатическим разрядам, помехам по сети питания, воздействию электромагнитных полей.

– Использование микроконтроллеров в извещателях позволяет повысить достоверность обнаружения, реализовать высокий уровень помехозащищенности, расширить сервисные функции, повысить удобство настройки и эксплуатации, повысить надежность.

– Обеспечивает дистанционный контроль охраняемой остекленной конструкции любой конфигурации;

– Совместим с различными видами и размерами стекол;

– Имеет многоуровневую микропроцессорную обработку сигнала, функциональное самотестирование;

– Представляет возможность пользователю производить выбор алгоритма работы извещателя под условия объекта и принятую тактику его охраны;

– Обеспечивает индикацию режимов работы извещателя и шумов внутри помещения с возможностью ее отключения

– Максимальная рабочая дальность действия извещателя - не менее 6 м

– Ток потребления извещателя - не более 22 мА

– Информативность извещателя равна пяти, а именно: - извещение “Норма"; - извещение "Тревога"; - извещение “Вскрытие”; - индикация помехи на первой рабочей частоте; - индикация помехи на второй рабочей частоте.

– Извещение “Норма" формируется извещателем в течение всего времени охраны замкнутыми контактами исполнительного

реле и выключенным состоянием индикатора красного цвета при отсутствии разрушающих воздействий на охраняемое стекло.

– Извещение "Тревога" формируется извещателем разомкнутыми контактами исполнительного реле и включенным состоянием индикатора красного цвета на время не менее 2 с при: - включении извещателя; - обнаружении разрушающих воздействий на охраняемое стекло; - снижении напряжения питания до (8,0+0,8) В.

– Извещение “Вскрытие" формируется извещателем размыканием контактов микровыключателя при вскрытии его корпуса.

– Индикация помехи на первой рабочей частоте осуществляется извещателем включением индикатора желтого цвета.

– Индикация помехи на второй рабочей частоте осуществляется извещателем включением индикатора зеленого цвета.

Примеры акустических извещателей приведены на рисунке 5.



Рисунок 5.2 – Примеры акустических извещателей

**Контактные датчики.**

Контактные извещатели служат для обнаружения несанкционированного открытия различных механических заградительных конструкций: дверей, ворот, окон и т.д. По принципу действия извещатели этого типа делятся на магнитные и механические.

Магнитные извещатели состоят из двух частей: герконового реле (геркона), устанавливаемого на неподвижную часть конструкции, и магнита, устанавливаемого на открывающийся модуль. Когда открывающийся модуль закрыт, магнит находится вблизи геркона и наводит в нем магнитное поле, удерживающее его контакты в замкнутом состоянии. При открытии магнит удаляется от геркона, размыкая его контакты.

Различают три основных вида магнитных датчиков (по принципу монтажа): накладные, врезные и для металлических дверей. Если обычный магнитный извещатель поставить на металлическую дверь без диэлектрической прокладки, со временем за счет намагничивания металла произойдет так называемое "залипание" геркона: его контакты будут замкнуты независимо от наличия вблизи магнита. Именно поэтому для металлических дверей существует специальный тип магнитных извещателей. В том случае, если необходимо контролировать открытие массивных конструкций (ворот, бронированных дверей сейфового типа), применяются механические контактные извещатели, называемые концевыми выключателями. Принцип их действия весьма прост и основан на чисто механическом замыкании/размыкании контактных групп. Контактные извещатели, в случае их срабатывания, обеспечивают самое быстрое обнаружение вторжения и вследствие простоты устройства самую низкую вероятность ложной тревоги. Однако при этом их наиболее просто обойти, поэтому они должны использоваться совместно с извещателями других типов.

Примеры извещателей:

Извещатель охранный магнитоконтактный ИО102-5(образец на рисунке 6) предназначен для скрытого монтажа на немагнитные поверхности.



Рисунок 6 ­– Извещатль ИО102-5

Извещатель охранный магнитоконтактный ИО102-6(пример на рисунке 7) предназначен для скрытого монтажа в сталь.



Рисунок 7 – Извещатель ИО102-6

Извещатель охранный магнитоконтактный MINI-10DM (рисунок 8) предназначен для скрытой установки. Подходит для любых дверей и окон, кроме металлических. Особенно удобен для пластиковых окон.



Рисунок 8 – Извещатель MINI-10DM

**Вибрационные извещатели.**

Вибрационные извещатели служат для защиты от проникновения путем разрушения различных строительных конструкций: бетонных стен и перекрытий, кирпичных стен, деревянных (рамы и двери) и потолочных покрытий, а также сейфов, металлических шкафов и банкоматов. Принцип действия вибрационных датчиков основан на пьезоэлектрическом эффекте, который состоит в изменении электрического сигнала при вибрации пьезоэлемента. Электрический сигнал, пропорциональный уровню вибрации, усиливается и обрабатывается схемой извещателя по специальному алгоритму, чтобы отделить разрушающее воздействие от помехового сигнала. Основными характеристиками таких извещателей является чувствительность к вибрации.

Вибрационные сигнализаторы уровня широко распространены за рубежом и, в меньшей степени, в России и странах СНГ. Чувствительный элемент вибрационного сигнализатора, не будучи погружен в рабочую среду, совершает механические колебания на резонансной частоте возбуждаемые пьезоэлектрическим генератором. Погружение в рабочую среду – жидкость или сыпучий продукт приводит к изменению частоты колебаний, изменению электрических параметров цепи и преобразуется в дискретный выходной сигнал. Пример извещателя на рисунке 9.



Рисунок 9 – Вибрационный извещатель

Вибрационные сигнализаторы мало зависят от физических свойств среды, могут работать в негомогенных, пенящихся, парящих, загазованных средах. Диапазон применимости датчиков по температуре -50°С..+250°С, давлению – до 64 атм., плотность рабочей среды –в пределах 0,5-2,5 г/см³. Датчики обеспечивают точность срабатывания +/-1 мм. Помимо предельных выключателей уровня, характерно применение вибрационных сигнализаторов в качестве датчиков сухого хода в трубопроводах. Вибрационные сигнализаторы выпускаются в широком диапазоне исполнений, в том числе для пищевых производств, взрывоопасных условий, агрессивных сред.

**Применение вибрационного извещателя.**

Основное назначение вибрационных извещателей - это защита различных строительных конструкций, средств инженерной укрепленности и хранилищ материальных ценностей. Они обнаруживают преднамеренное разрушение бетонных стен и перекрытий, стальных дверей, шкафов и сейфов, в том числе - бронированных, засыпных, с железобетонным заполнением внутреннего объема двустенной оболочки. С помощью таких извещателей можно также защитить от вандализма лицевые панели банкоматов, организовать охрану конструкций, выполненных с применением многослойных защитных стекол и стеклоблоков, кирпичных стен и перегородок, стальных решеток, деревянных дверей, оконных рам (при небольшом размере многосекционных остекленных проемов), стен, перекрытий, перегородок, конструкций из фанеры и древесностружечных плит.

**Особенности вибрационного извещателя.**

Принцип действия рассматриваемых извещателей основан на анализе вибрационных сигналов, возникающих в строительных конструкциях при нанесении разрушающих воздействий с целью проникновения в охраняемое помещение. Принцип действия чувствительного элемента требует непосредственного механического контакта с охраняемой поверхностью. Он представляет собой пьезоэлектрический акселерометр, преобразующий механические вибрации в переменный электрический сигнал, амплитуда которого в каждый момент времени пропорциональна величине виброускорения. Переменное напряжение с чувствительного элемента поступает на электронную схему извещателя, которая в соответствии с заложенным в нее алгоритмом производит обработку сигнала в установленном диапазоне (диапазонах) частот, анализирует его параметры на соответствие заданным критериям и формирует тревожное извещение. В многопозиционных извещателях сигнал с чувствительного элемента поступает на предусилитель и фильтр ДСВ, выходной каскад которого пропорционально изменяет ток потребления в линии соединения с БОС и таким образом передает аналоговый сигнал на электронную схему БОС для дальнейшей аналого-цифровой обработки и формирования извещений.

Во всех представленных вибрационных извещателях реализована способность различать виды помеховых воздействий. Таким образом, чтобы не реагировать на помехи случайного характера или естественного происхождения и выдавать извещение о тревоге только при умышленно созданной помехе, направленной на снижение чувствительности извещателя.

**Размещение вибрационного извещателя.**

Вибрационные извещатели устанавливают внутри охраняемого помещения в местах, защищенных от случайных механических повреждений и доступа посторонних лиц (по возможности). При выборе места крепления датчика необходимо ознакомиться со специфическими особенностями охраняемого объекта (формой и размером помещения, расположением дверных и оконных проемов, толщиной и материалом стен, перекрытий и других конструкций, подлежащих защите от попытки пролома, расположением водопроводных труб и элементов системы центрального отопления). Охраняемая зона, создаваемая извещателем на монолитной конструкции, как правило, представляет собой круг, радиус которого определяется как дальность действия датчика.

**Устойчивость к внешним воздействиям.**

Для вибрационных извещателей существуют уже сложившиеся требования по помехозащищенности, эффективность которых подтверждена многолетней практикой. Так, например, извещатели серии "Шорох" устойчивы к воздействию одиночного удара по поверхности охраняемой конструкции в виде затухающего колебания, а также к длительным помеховым воздействиям, таким как шум водопроводных труб, работа лифта, вентиляционных агрегатов, вибрация от автотранспорта, железнодорожных составов, самолетов и т.п. Извещатели серии "Грань" имеют аналогичные параметры по помехозащищенности, приводимые для их ДСВ. Кроме того, в соответствии с требованиями ГОСТ Р извещатели должны быть устойчивы к воздействию электромагнитных полей, электростатических разрядов, помехам по линиям электропитания и связи, пропаданиям сетевого напряжения. Эти виды воздействий имитируют выверенные многолетним опытом условия эксплуатации охранной техники на реальных объектах. Данные параметры извещателей контролируются при сертификации на соответствие требованиям ГОСТ Р.

**Объёмные датчики.**

Анализ номенклатуры датчиков, предлагаемых крупнейшими производителями систем охранной сигнализации, показывает, что в классе датчиков для охраны помещений наиболее популярными являются инфракрасные (ИК) пассивные, комбинированные (в основном ИК + микроволновые). Реже применяются микроволновые, ультразвуковые датчики.

**Активные инфракрасные извещатели.**

ИК-барьеры широко используются для охраны периметра. Другие названия приборов: активный ИК-барьер, охранный линейный извещатель, барьер инфракрасный, ИК-активный лучевой датчик. Принцип действия этой группы датчиков следующий: Передатчик излучает ИК импульсы определенной частоты, приемник же в свою очередь принимает эти импульсы, усиливает их и подает на считыватель, который отслеживает их количество. Ели в зоне действия датчика появляется препятствие, и часть импульсов не доходит до приемника, то выдается сигнал тревоги. Многим из нас еще со школьных уроков физики знакомы приборы, реагирующие на пересечение непрозрачным предметом светового луча. Такие приборы обычно состоят из источника света, часто простой лампочки и фотоприемника с усилителем, нагруженным на исполнительное устройство. Такие приборы широко используются для счета продукции на конвейерах заводов, в системах охранной сигнализации в турникетах метро, в приборах защитного отключения и во многих других местах.

При своей относительной простоте, приборам, построенным на базе лампочки накаливания и фотоприемника с усилителем, свойственен ряд недостатков – они отличаются низкой экономичностью, имеют значительные габариты, требуют дополнительную оптическую систему, плохо работают при наличии внешней засветки, имеют невысокую надежность и высокую вероятность ложных срабатываний. Кроме того, из-за использования диапазона видимого света такие устройства демаскируют себя, что затрудняет их использование в системах охраны.

Для устранения указанных недостатков разработчики профессиональных приборов такого класса в качестве источников используют излучатели, работающие в невидимом инфракрасном диапазоне которые излучают не непрерывный сигнал, а сигнал сложной формы. В приемниках используются специальные оптические фильтры, отсекающие мешающий сигнал видимого излучения, высокочувствительные PIN фотоприемники, усилители с АРУ и сложными системами фильтрации. Все эти меры позволяют создавать весьма надежные и эффективные системы контроля, называемые инфракрасными барьерами. Но из-за сложности и высокой цены они до недавнего времени были доступны только для профессионалов. Обычно барьеры бывают спаренные, то есть в одном корпусе два передатчика, в другом два приемника, расположенных вертикально друг над другом на не большом расстоянии. Это сделано для уменьшения вероятности ложного срабатывания, датчик сработает только тогда, когда по обоим каналам получен тревожный сигнал. Приемник и передатчик монтируют, так что бы они располагались на одной оси, а при юстировке датчика, нужно стараться получить как можно больший сигнал на приемнике, и зафиксировать блоки в этом положении.

На рисунке 10 представлены примеры активных ик – извещателей.



Рисунок 10 - Активные инфракрасные извещатели

Система незаменима для построения периметральной охраны на коттеджных участках, в ландшафтных парках и других зонах, где важное значение придается эстетичному внешнему виду.

Данный тип датчиков не редко используют в роли пожарных извещателей, устанавливая ИК барьеры под потолком. Обычно это большие по площади и высоте помещения, там где не возможно или не целесообразно применение дымовых датчиков. При возгорании в помещении, дым блокирует ИК импульсы, посланные передатчиком, приемник их не получает, в связи, с чем и выдает сигнал тревоги.

В проекте охранной сигнализации используем пассивные ИК-датчики (Рисунок 11). Этот вид датчиков является самым сложным, пишем их более подробно.



Рисунок 11 - Пассивный ИК-датчик

Основу охранных сигнализаций составляют объемные извещатели. Далее рассматриваются принципы действия и особенности применения таких датчиков. Эти датчики предназначены в первую очередь для защиты объема охраняемого помещения.

ИК-пассивные датчики, называемые также оптико-электронными, относятся к классу детекторов движения и реагируют на тепловое излучение движущегося человека. Принцип действия этих датчиков основан на регистрации изменения во времени разницы между интенсивностью ИК излучения от человека и фонового теплового излучения. В настоящее время ИК-пассивные датчики являются самыми популярными, они составляют неотъемлемый элемент охранной системы практически каждого объекта.

Для того чтобы нарушитель был обнаружен ИК-пассивным датчиком, необходимо выполнение следующих условий:

1. нарушитель должен пересечь в поперечном направлении луч зоны чувствительности датчика;
2. движение нарушителя должно происходить в определенном интервале скоростей;
3. чувствительность датчика должна быть достаточной для регистрации разницы температур поверхности тела нарушителя (с учетом влияния его одежды) и фона (стены, пол).

ИК-пассивные датчики состоят из трех основных элементов:

1. оптической системы, формирующей диаграмму направленности датчика и определяющей форму и вид пространственной зоны чувствительности;
2. пироприемника, регистрирующего тепловое излучение человека;
3. блока обработки сигналов пироприемника, выделяющего сигналы, обусловленные движущимся человеком, на фоне помех естественного и искусственного происхождения.

**Оптическая система.** Современные ИК-датчики характеризуются большим разнообразием возможных форм диаграмм направленности. Зона чувствительности ИК-датчиков представляет собой набор лучей различной конфигурации, расходящихся от датчика по радиальным направлениям в одной или нескольких плоскостях. В связи с тем, что в ИК-детекторах используются сдвоенные пироприемники, каждый луч в горизонтальной плоскости расщепляется на два.

Зона чувствительности детектора может иметь вид(схема приведена на рисунке 12):

1. одного или нескольких, сосредоточенных в малом угле, узких лучей;
2. нескольких узких лучей в вертикальной плоскости (лучевой барьер);
3. одного широкого в вертикальной плоскости луча (сплошной занавес) или в виде многовеерного занавеса;
4. нескольких узких лучей в горизонтальной или наклонной плоскости (поверхностная одноярусная зона);
5. нескольких узких лучей в нескольких наклонных плоскостях (объемная многоярусная зона).



Рисунок 12 - Диаграммы зон обнаружения

Существует требование равномерной чувствительности детектора. Ниже приведено описание этого требования. Сигнал на выходе пироприемника при прочих равных условиях тем больше, чем больше степень перекрытия нарушителем зоны чувствительности детектора и чем меньше ширина луча и расстояние до детектора. Для обнаружения нарушителя на большом (10...20 м) расстоянии желательно, чтобы в вертикальной плоскости ширина луча не превышала 5°...10°, в этом случае человек практически полностью перекрывает луч, что обеспечивает максимальную чувствительность. На меньших расстояниях чувствительность детектора в этом луче существенно возрастает, что может привести к ложным срабатываниям, например, от мелких животных. Для уменьшения неравномерной чувствительности используются оптические системы, формирующие несколько наклонных лучей, ИК детектор при этом устанавливается на высоте выше человеческого роста. Общая длина зоны чувствительности тем самым разделяется на несколько зон, причем "ближние" к детектору лучи для снижения чувствительности делаются обычно более широкими. За счет этого обеспечивается практически постоянная чувствительность по расстоянию, что с одной стороны способствует уменьшению ложных срабатываний, а с другой стороны повышает обнаружительную способность за счет устранения мертвых зон вблизи детектора.  
При построении оптических систем ИК-датчиков могут использоваться:

1. линзы Френеля - фасеточные (сегментированные) линзы, представляющие собой пластиковую пластину с отштампованными на ней несколькими призматическими линзами-сегментами;
2. зеркальная оптика - в датчике устанавливается несколько зеркал специальной формы, фокусирующих тепловое излучение на пироприемник;
3. комбинированная оптика, использующая и зеркала, и линзы Френеля.

В большинстве ИК-пассивных датчиков используются линзы Френеля. К достоинствам линз Френеля относятся:

1. простота конструкции детектора на их основе;
2. низкая цена;
3. возможность использования одного датчика в различных приложениях при использовании сменных линз.

Обычно каждый сегмент линзы Френеля формирует свой луч диаграммы направленности. Использование современных технологий изготовления линз позволяет обеспечить практически постоянную чувствительность детектора по всем лучам за счет подбора и оптимизации параметров каждой линзы-сегмента: площади сегмента, угла наклона и расстояния до пироприемника, прозрачности, отражающей способности, степени дефокусировки. В последнее время освоена технология изготовления линз Френеля со сложной точной геометрией, что дает 30% увеличение собираемой энергии по сравнению со стандартными линзами и соответственно увеличение уровня полезного сигнала от человека на больших расстояниях. Материал, из которого изготавливаются современные линзы, обеспечивает защиту пироприемника от белого света. К неудовлетворительной работе ИК-датчика могут привести такие эффекты, как тепловые потоки, являющиеся результатом нагревания электрических компонентов датчика, попадание насекомых на чувствительные пироприемники, возможные переотражения инфракрасного излучения от внутренних частей детектора. Для устранения этих эффектов в ИК-датчиках последнего поколения применяется специальная герметичная камера между линзой и пироприемником (герметичная оптика.

Зеркальная оптика как единственный элемент оптической системы применяется достаточно редко. Преимуществами зеркальной оптики являются возможность более точной фокусировки и, как следствие, увеличение чувствительности, что позволяет обнаруживать нарушителя на больших расстояниях. Использование нескольких зеркал специальной формы, в том числе многосегментных, позволяет обеспечить практически постоянную чувствительность по расстоянию, причем эта чувствительность на дальних расстояниях приблизительно на 60% выше, чем для простых линз Френеля. С помощью зеркальной оптики проще обеспечивается защита ближней зоны, расположенной непосредственно под местом установки датчика (так называемая антисаботажная зона). По аналогии со сменными линзами Френеля, ИК-датчики с зеркальной оптикой комплектуются сменными отстегивающимися зеркальными масками, применение которых позволяет выбирать требуемую форму зоны чувствительности и дает возможность адаптировать датчик к различным конфигурациям защищаемого помещения.

В современных высококачественных ИК-детекторах используется комбинация линз Френеля и зеркальной оптики. При этом линзы Френеля используются для формирования зоны чувствительности на средних расстояниях, а зеркальная оптика - для формирования антисаботажной зоны под датчиком и для обеспечения очень большого расстояния обнаружения.

Пироприемник.

Оптическая система фокусирует ИК излучение на пироприемнике, в качестве которого в ИК-датчиках используется сверхчувствительный полупроводниковый пироэлектрический преобразователь, способный зарегистрировать разницу в несколько десятых градуса между температурой тела человека и фона. Изменение температуры преобразуется в электрический сигнал, который после соответствующей обработки вызывает сигнал тревоги. В ИК-датчиках обычно используются сдвоенные (дифференциальные) пироэлементы. Это связано с тем, что одиночный пироэлемент одинаковым образом реагирует на любое изменение температуры независимо от того, чем оно вызвано - человеческим телом или, например, обогревом помещения, что приводит к повышению частоты ложных срабатываний. В дифференциальной схеме производится вычитание сигнала одного пироэлемента из другого, что позволяет существенно подавить помехи, связанные с изменением температуры фона, а также заметно снизить влияние световых и электромагнитных помех. Сигнал от движущегося человека возникает на выходе сдвоенного пироэлемента только при пересечении человеком луча зоны чувствительности и представляет собой почти симметричный двухполярный сигнал, близкий по форме к периоду синусоиды. Сам луч для сдвоенного пироэлемента по этой причине расщепляется в горизонтальной плоскости на два. В последних моделях ИК-датчиков с целью дополнительного снижения частоты ложных срабатываний используются счетверенные пироэлементы - это два сдвоенных пироприемника, расположенные в одном датчике (обычно размещаются один над другим). Радиусы наблюдения этих пироприемников делаются различными, и поэтому локальный тепловой источник ложных срабатываний не будет наблюдаться в обоих пироприемниках одновременно. При этом геометрия размещения пироприемников и схема их включения выбирается таким образом, чтобы сигналы от человека были противоположной полярности, а электромагнитные помехи вызывали сигналы в двух каналах одинаковой полярности, что приводит к подавлению и этого типа помех. Для счетверенных пироэлементов каждый луч расщепляется на, в связи с чем максимальное расстояние обнаружения при использовании одинаковой оптики уменьшается приблизительно вдвое, так как для надежного обнаружения человек должен своим ростом перекрывать оба луча от двух пироприемников. Повысить расстояние обнаружения для счетверенных пироэлементов позволяет использование прецизионной оптики, формирующей более узкий луч. Другой путь, позволяющий в некоторой степени исправить это положение - применение пироэлементов со сложной переплетенной геометрией.

Блок обработки сигналов.

Блок обработки сигналов пироприемника должен обеспечивать надежное распознавание полезного сигнала от движущегося человека на фоне помех. Для ИК-датчиков основными видами и источниками помех, могущими вызвать ложное срабатывание, являются:

1. источники тепла;
2. конвенционное движение воздуха;
3. солнечная радиация и искусственные источники света;
4. электромагнитные и радиопомехи (транспорт с электродвигателями, электросварка, линии электропередачи, мощные радиопередатчики, электростатические разряды);
5. сотрясения и вибрации;
6. термическое напряжение линз;
7. насекомые и мелкие животные.

Выделение блоком обработки полезного сигнала на фоне помех основано на анализе параметров сигнала на выходе пироприемника. Такими параметрами являются величина сигнала, его форма и длительность. Сигнал от человека, пересекающего луч зоны чувствительности ИК-датчика, представляет собой почти симметричный двухполярный сигнал, длительность которого зависит от скорости перемещения нарушителя, расстояния до датчика, ширины луча, и может составлять приблизительно 0,02...10 с при регистрируемом диапазоне скоростей перемещения 0,1...7 м/с. Помеховые сигналы в большинстве своем являются несимметричными или имеющими отличную от полезных сигналов длительность. Основным параметром, анализируемым всеми датчиками, является величина сигнала. В простейших датчиках этот регистрируемый параметр является единственным, и его анализ производится путем сравнения сигнала с некоторым порогом, который определяет чувствительность датчика и влияет на частоту ложных тревог. С целью повышения устойчивости к ложным тревогам в простых датчиках используется метод счета импульсов, когда подсчитывается, сколько раз сигнал превысил порог (то есть, по сути, сколько раз нарушитель пересек луч или сколько лучей он пересек). При этом тревога выдается не при первом превышении порога, а только если в течение определенного времени количество превышений становится больше заданной величины (обычно 2...4). Недостатком метода счета импульсов является ухудшение чувствительности, особенное заметное для датчиков с зоной чувствительности типа одиночного занавеса и ей подобной, когда нарушитель может пересечь только один луч. С другой стороны, при счете импульсов возможны ложные срабатывания от повторяющихся помех (например, электромагнитных или вибраций).  
В более сложных датчиках блок обработки анализирует двухполярность и симметрию формы сигналов с выхода дифференциального пироприемника. Суть обработки состоит в сравнении сигнала с двумя порогами (положительным и отрицательным) и, в ряде случаев, сравнении величины и длительности сигналов разной полярности. Возможна также комбинация этого метода с раздельным подсчетом превышений положительного и отрицательного порогов. Анализ длительности сигналов может проводиться как прямым методом измерения времени, в течение которого сигнал превышает некоторый порог, так и в частотной области путем фильтрации сигнала с выхода пироприемника, в том числе с использованием "плавающего" порога, зависящего от диапазона частотного анализа.

Еще одним видом обработки, предназначенным для улучшения характеристик ИК-датчиков, является автоматическая термокомпенсация. В диапазоне температур окружающей среды 25°С...35°С чувствительность пироприемника снижается за счет уменьшения теплового контраста между телом человека и фоном, при дальнейшем повышении температуры чувствительность снова повышается, но "с противоположным знаком". В так называемых "обычных" схемах термокомпенсации осуществляется измерение температуры, и при ее повышении производится автоматическое увеличение усиления. При "настоящей" или "двухсторонней" компенсации учитывается повышение теплового контраста для температур выше 25°С...35°С. Использование автоматической термокомпенсации обеспечивает почти постоянную чувствительность ИК-датчика в широком диапазоне температур.

Перечисленные виды обработки могут проводиться аналоговыми, цифровыми или комбинированными средствами. В современных ИК-датчиках все шире начинают использоваться методы цифровой обработки с использованием специализированных микроконтроллеров с АЦП и сигнальных процессоров, что позволяет проводить детальную обработку тонкой структуры сигнала для лучшего выделения его на фоне помех. В последнее время появились сообщения о разработке полностью цифровых ИК-датчиков, вообще не использующих аналоговых элементов.

Как известно, вследствие случайного характера полезных и помеховых сигналов наилучшими являются алгоритмы обработки, основанные на теории статистических решений.

Косвенными признаками хорошего современного датчика могут быть наличие АЦП, микропроцессора и, что стали в последнее время сообщать производители, объема используемой программы обработки, который имеет величину несколько тысяч байт.

Другие элементы защиты ИК-датчиков.

В ИК-датчиках, предназначенных для профессионального использования, применяются так называемые схемы антимаскинга. Суть проблемы состоит в том, что обычные ИК-датчик могут быть выведены нарушителем из строя путем предварительного (когда система не поставлена на охрану) заклеивания или закрашивания входного окна датчика. Для борьбы с этим способом обхода ИК-датчиков и используются схемы антимаскинга. Метод основывается на использовании специального канала ИК-излучения, срабатывающего при появлении маски или отражающей преграды на небольшом расстоянии от датчика (от 3 до 30 см). Схема антимаскинга работает непрерывно, пока система снята с охраны. Когда факт маскирования обнаруживается специальным детектором, сигнал об этом подается с датчика на контрольную панель, которая, однако, не выдает сигнала тревоги до тех пор, пока не придет время постановки системы на охрану. Именно в этот момент оператору и будет выдана информация о маскировании. Причем, если это маскирование было случайным (крупное насекомое, появление крупного объекта на некоторое время вблизи датчика и т.п.) и к моменту постановки на сигнализацию самоустранилось, сигнал тревоги не выдается.

Для устранения срабатываний ИК-датчика от мелких животных используются либо специальные линзы с зоной нечувствительности от уровня пола до высоты порядка 1 м, либо специальные методы обработки сигналов. Следует учитывать, что специальная обработка сигналов позволяет игнорировать животных только в том случае, если их общий вес не превышает 7...15 кг, и они могут приблизиться к датчику не ближе 2 м.

Установка и использование ИК-датчиков.

При выборе типов и количества датчиков для обеспечения охраны конкретного объекта следует учитывать возможные пути и способы проникновения нарушителя, требуемый уровень надежности обнаружения; расходы на приобретение, монтаж и эксплуатацию датчиков; особенности объекта; тактико-технические характеристики датчиков. Особенностью ИК-пассивных датчиков является их универсальность - с их использованием возможно блокирование от подхода и проникновения самых разнообразных помещений, конструкций и предметов: окон, витрин, прилавков, дверей, стен, перекрытий, перегородок, сейфов и отдельных предметов, коридоров, объемов помещений. При этом в ряде случаев не потребуется большого количества датчиков для защиты каждой конструкции - может оказаться достаточным применения одного или нескольких датчиков с нужной конфигурацией зоны чувствительности. Остановимся на рассмотрении некоторых особенностей применения ИК-датчиков.

Общий принцип использования ИК-датчиков - лучи зоны чувствительности должны быть перпендикулярны предполагаемому направлению движения нарушителя. Место установки датчика следует выбирать так, чтобы минимизировать мертвые зоны, вызванные наличием в охраняемом помещении крупных предметов, перекрывающих лучи (например, мебель, комнатные растения). Если в помещении двери открываются внутрь, следует учитывать возможность маскировки нарушителя открытыми дверьми. При невозможности устранить мертвые зоны следует использовать несколько датчиков. При блокировке отдельных предметов датчик или датчики нужно устанавливать так, чтобы лучи зоны чувствительности блокировали все возможные подходы к защищаемым предметам.

Должен соблюдаться задаваемый в документации диапазон допустимых высот подвески (минимальная и максимальная высоты). В особенности это относится к диаграммам направленности с наклонными лучами: если высота подвески будет превышать максимально допустимую, то это приведет к уменьшению сигнала из дальней зоны и увеличению мертвой зоны перед датчиком, если же высота подвески будет меньше минимально допустимой, то это приведет к уменьшению дальности обнаружения с одновременным уменьшением мертвой зоны под датчиком.  
К ложным срабатываниям ИК-датчиков могут привести помехи теплового, светового, электромагнитного, вибрационного характера. Несмотря на то, что современные ИК-датчики имеют высокую степень защиты от указанных воздействий, все же целесообразно придерживаться следующих рекомендаций:

1. для защиты от потоков воздуха и пыли не рекомендуется размещать датчик в непосредственной близости от источников воздушных потоков (вентиляция, открытое окно);
2. следует избегать прямого попадания на датчик солнечных лучей и яркого света; при выборе места установки должна учитывается возможность засветки в течение непродолжительного времени рано утром или на закате, когда солнце низко над горизонтом, или засветки фарами проезжающего снаружи транспорта;
3. на время постановки на охрану целесообразно отключать возможные источники мощных электромагнитных помех, в частности источники света не на основе ламп накаливания: люминесцентные, неоновые, ртутные, натриевые лампы;
4. для снижения влияния вибраций целесообразно устанавливать датчик на капитальных или несущих конструкциях;
5. не рекомендуется направлять датчик на источники тепла (радиатор, печь) и колеблющиеся предметы (растения, шторы), в сторону нахождения домашних животных.
   1. Микроконтроллер: его виды и характеристики

**Выбор микроконтроллера.**

В настоящее время среди всех 8-разрядных микроконтроллеров - семейство MCS-51 является несомненным чемпионом по количеству разновидностей и количеству компаний, выпускающих его модификации. Оно получило свое название от первого представителя этого семейства - микроконтроллера 8051, выпущенного в 1980 году на базе технологии HMOS. Удачный набор периферийных устройств, возможность гибкого выбора внешней или внутренней программной памяти и приемлемая цена обеспечили этому микроконтроллеру успех на рынке.

Важную роль в достижении такой высокой популярности семейства 8051 сыграла открытая политика фирмы Intel, родоначальницы архитектуры, направленная на широкое распространение лицензий на ядро 8051 среди большого количества ведущих полупроводниковых компаний мира.

В результате на сегодняшний день существует более 200 модификаций микроконтроллеров семейства 8051, выпускаемых почти 20-ю компаниями. Эти модификации включают в себя кристаллы с широчайшим спектром периферии: от простых 20-выводных устройств с одним таймером и 1К программной памяти до сложнейших 100-выводных кристаллов с 10-разрядными АЦП, массивами таймеров-счетчиков, аппаратными 16-разрядными умножителями и 64К программной памяти на кристалле. Каждый год появляются все новые варианты представителей этого семейства. Основными направлениями развития являются: увеличение быстродействия (повышение тактовой частоты и переработка архитектуры), снижение напряжения питания и потребления, увеличение объема ОЗУ и FLASH памяти на кристалле с возможностью внутрисхемного программирования, введение в состав периферии микроконтроллера сложных устройств типа системы управления приводами, CAN и USB интерфейсов и т.п.

Все микроконтроллеры из семейства MCS-51 имеют общую систему команд. Наличие дополнительного оборудования влияет только на количество регистров специального назначения.

Основными производителями клонов 51-го семейства в мире являются фирмы Philips, Siemens, Intel, Atmel, Dallas, Temic, Oki, AMD, MHS, Gold Star, Winbond, Silicon Systems и ряд других.

В рамках СССР производство микроконтроллера 8051 осуществлялось в Киеве, Воронеже (1816ВЕ31/51, 1830ВЕ31/51), Минске (1834ВЕ31) [9].

Для данной задачи – разработки системы охранной сигнализации - микроконтроллер этого семейства является оптимальным, так как сочетает в себе большие возможности управления, необходимые для решения поставленной нами задачи. А так же к его качествам можно отнести высокую надежность, что является обязательным условием охранных систем, и дешевизну.

Приведем описание микроконтроллера семейства 8051.(Рисунок 13)

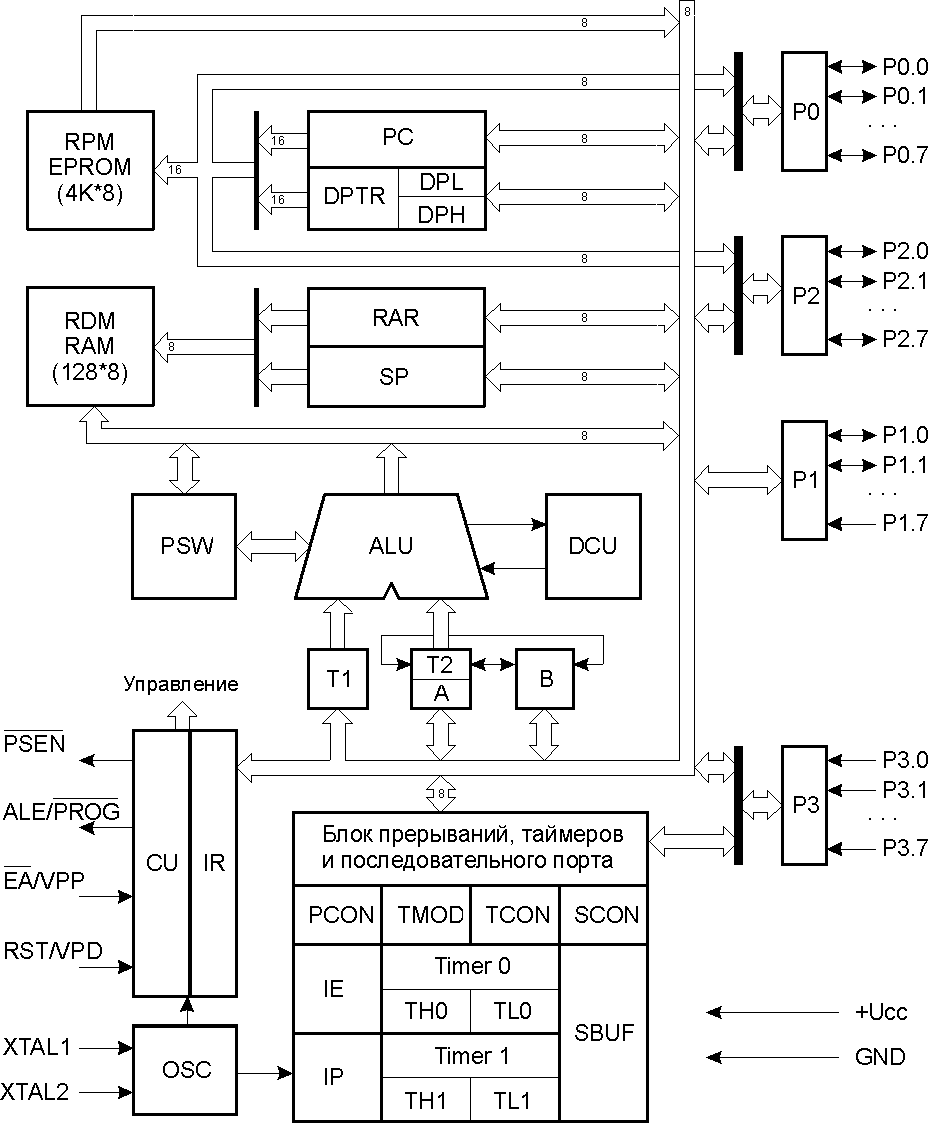


Рисунок 13 - Структурная схема микроконтроллера КМ1816ВЕ51

#### Общие характеристики.

#### Микроконтроллер семейства 8051 имеют следующие аппаратные особенности:

– внутреннее ОЗУ объемом 128 байт;

– четыре двунаправленных побитно настраиваемых восьмиразрядных порта ввода-вывода;

– два 16-разрядных таймера-счетчика;

– встроенный тактовый генератор;

– адресация 64 КБайт памяти программ и 64 Кбайт памяти данных;

– две линии запросов на прерывание от внешних устройств;

– интерфейс для последовательного обмена информацией с другими микроконтроллерами или персональными компьютерами.

Микроконтроллер выполнен на основе высокоуровневой n-МОП технологии. Через четыре программируемых параллельных порта ввода/вывода и один последовательный порт микроконтроллер взаимодействует с внешними устройствами. Основу структурной схемы (рис. 1) образует внутренняя двунаправленная8-битная шина, которая связывает между собой основные узлы и устройства микроконтроллера: резидентную память программ (RPM), резидентную память данных (RDM), арифметико-логическое устройство (ALU), блок регистров специальных функций, устройство управления (CU) и порты ввода/вывода (P0-P3).

### 

### Арифметико-логическое устройство.

8-битное арифметико-логическое устройство (ALU) может выполнять арифметические операции сложения, вычитания, умножения и деления; логические операции И, ИЛИ, исключающее ИЛИ, а также операции циклического сдвига, сброса, инвертирования и т.п. К входам подключены программно-недоступные регистры T1 и T2, предназначенные для временного хранения операндов, схема десятичной коррекции (DCU) и схема формирования признаков результата операции (PSW).

Простейшая операция сложения используется в ALU для инкрементирования содержимого регистров, продвижения регистра-указателя данных (RAR) и автоматического вычисления следующего адреса резидентной памяти программ. Простейшая операция вычитания используется в ALU для декрементирования регистров и сравнения переменных.

Простейшие операции автоматически образуют “тандемы” для выполнения таких операций, как, например, инкрементирование 16-битных регистровых пар. В ALU реализуется механизм каскадного выполнения простейших операций для реализации сложных команд. Так, например, при выполнении одной из команд условной передачи управления по результату сравнения в ALU трижды инкрементируется счётчик команд (PC), дважды производится чтение из RDM, выполняется арифметическое сравнение двух переменных, формируется 16-битный адрес перехода и принимается решение о том, делать или не делать переход по программе. Все перечисленные операции выполняются всего лишь за 2 мкс.

Важной особенностью ALU является его способность оперировать не только байтами, но и битами. Отдельные программно-доступные биты могут быть установлены, сброшены, инвертированы, переданы, проверены и использованы в логических операциях. Эта способность достаточно важна, поскольку для управления объектами часто применяются алгоритмы, содержащие операции над входными и выходными булевыми переменными, реализация которых средствами обычных микропроцессоров сопряжена с определенными трудностями.

Таким образом, ALU может оперировать четырьмя типами информационных объектов: булевыми (1 бит), цифровыми (4 бита), байтными (8 бит) и адресными (16 бит). В ALU выполняется 51 различная операция пересылки или преобразования этих данных. Так как используется 11 режимов адресации (7 для данных и 4 для адресов), то путем комбинирования операции и режима адресации базовое число команд 111 расширяется до 255 из 256 возможных при однобайтном коде операции.

На рисунке 14 приведены назначение выводов микроконтроллера семейства 8051.



### Рисунок 14 – Назначение выводов 8051

Обозначения на этом рисунке:

– Vss - потенциал общего провода ("земли");

– Vcc - основное напряжение литания +5 В;

– X1,X2 - выводы для подключения кварцевого резонатора;

– RESET - вход общего сброса микроконтроллера;

– PSEN - разрешение внешней памяти программ; выдается только при обращении к внешнему ПЗУ;

– ALE - строб адреса внешней памяти;

– ЕА - отключение внутренней программной память; уровень 0 на этом входе заставляет микроконтроллер выполнять программу только внешнее ПЗУ; игнорируя внутреннее(если последнее имеется);

– P1 - восьми битный квази двунаправленный порт ввода/вывода: каждый разряд порта может быть запрограммирован как на ввод, так и на вывод информации, независимо от состояния других разрядов;

– P2 - восьми битный квази двунаправленный порт, аналогичный Р1; кроме того, выводы этого порта используются для выдачи адресной информации при обращении к внешней памяти программ или данных (если используется 16-битовая адресация последней). Выводы порта используются при программировании 8751 для ввода в микроконтроллер старших разрядов адреса:

– РЗ - восьми битный квази двунаправленный порт, аналогичный. Р1; кроме того, выводы этого порта могут выполнять ряд альтернативных функций, которые используются при работе таймеров, порта последовательного ввода-вывода, контроллера прерываний, и внешней памяти программ и данных;

– P0 - восьми битный двунаправленный порт ввода-вывода информации: при работе с внешними ОЗУ и ПЗУ по линиям порта в режиме временного мультиплексирования выдается адрес внешней памяти, после чего осуществляется передача или прием данных.

# **Организация ОЗУ, ПЗУ и регистров микроконтроллера 8051.**

### Память программ (ПЗУ).

Как и у большинства микроконтроллеров, у микроконтроллеров семейства 8051, память программ и память данных являются самостоятельными и независимыми друг от друга устройствами, адресуемыми различными командами и управляющими сигналами.

Объем встроенной памяти программ, расположенной на кристалле микроконтроллера 8051, равен 4 Кбайт. При обращении к внешней памяти программ все микроконтроллеры семейства 8051 всегда используют 16-разрадный адрес, что обеспечивает им доступ к 64 Кбайт ПЗУ. Микроконтроллер обращается к программной памяти при чтении кода операции и операндов (используя счетчик команд PC), а также при выполнении команд переноса байта из памяти программ в аккумулятор. При выполнении команд переноса данных адресация ячейки памяти программ, из которой будут прочитаны данные, может осуществляться с использованием как счетчика PC, так и специального двухбайтового регистра-указателя данных DPTR.

### Память данных (ОЗУ).

Объем расположенной на кристалле памяти данных—128 байт. Объем внешней памяти данных может достигать 64 Кбайт. Первые 32 байта организованы в четыре банка регистров общего назначения, обозначаемых соответственно банк 0 — банк 3. Каждый из них состоит из восьми регистров R0 — R7. В любой момент программе доступен только один банк регистров, номер которого содержится в третьем и четвертом битах слова состояния программы PSW (см. ниже).

Оставшееся адресное пространство может конфигурироваться разработчиком по своему усмотрению: в нем располагаются стек, системные и пользовательские области данных. Обращение к ячейкам памяти данных возможно двумя способами. Первый способ — прямая адресация ячейки памяти. В этом случае адрес ячейки является операндом соответствующей команды. Второй способ — косвенная адресация с помощью регистров R0 или R1: перед выполнением соответствующей команды в один из них должен быть занесен адрес ячейки, к которой необходимо обратиться.

Для обращения к внешней памяти данных используется только косвенная адресация с помощью регистров R0 и R1 или с помощью 16-разрядного регистра-указателя DPTR. Он относится к группе регистров специальных функций, и с его помощью можно адресовать все 64 Кбайта внешней памяти.

Часть памяти данных представляет собой так называемую битовую область, в ней имеется возможность при помощи специальных битовых команд адресовываться к каждому разряду ячеек памяти. Адрес прямо адресуемых битов может быть записан либо в виде (Адрес Байта ).(Разряд), например выражение 21.3 означает третий разряд ячейки памяти с адресом 21H, либо в виде абсолютного битового адреса. Соответствие этих двух способов адресации можно определить по таблице.

Таблица 1 – Адреса битовых областей памяти микроконтроллера 8051

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адрес  байта | Адреса битов по разрядам | | | | | | | |
| Adr | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 2FH | 7F | 7E | 7D | 7C | 7B | 7A | 79 | 78 |
| 2EH | 77 | 76 | 75 | 74 | 73 | 72 | 71 | 70 |
| 2DH | 6F | 6E | 6D | 6C | 6B | 6A | 69 | 68 |
| 2CH | 67 | 66 | 65 | 64 | 63 | 62 | 61 | 60 |
| 2BH | 5F | 5E | 5D | 5C | 5B | 5A | 59 | 58 |
| 2AH | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 |
| 29H | 4F | 4E | 4D | 4C | 4B | 4A | 49 | 48 |
| 28H | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 |
| 27H | 3F | 3E | 3D | 3C | 3B | 3A | 39 | 38 |
| 26H | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 |
| 25H | 2F | 2E | 2D | 2C | 2B | 2A | 29 | 28 |
| 24H | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| 23H | 1F | 1E | 1D | 1C | 1B | 1A | 19 | 18 |
| 22H | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
| 21H | 0F | 0E | 0D | 0C | 0B | 0A | 09 | 08 |
| 20H | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |

Примечание*.* Адрес прямо адресуемых битов может быть записан либо в виде выражения (Адрес Байта ).(Разряд), например выражение 21.3 означает адрес третьего разряда ячейки памяти с адресом 21H, либо в виде абсолютного битового адреса, который для данного бита равен (см. таблицу) 0B.

### Таблица 2 – Карта адресуемых битов в блоке регистров специальных функций.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адрес  байта | Адреса битов по разрядам | | | | | | | | Имя  регистра |
| **Adr** | **D7** | **D6** | **D5** | **D4** | **D3** | **D2** | **D1** | **D0** | **Name** |
| F0H | F7 | F6 | F5 | F4 | F3 | F2 | F1 | F0 | B |
| ... | . . . | | | | | | | | ... |
| E0H | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 | ACC |
| ... | . . . | | | | | | | | ... |
| D0H | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | PSW |
| ... | . . . | | | | | | | | ... |
| B8H | - | - | - | BC | BB | BA | B9 | B8 | IP |
| ... | . . . | | | | | | | | ... |
| B0 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | P3 |
| ... | . . . | | | | | | | | ... |
| A8H | AF | - | - | AC | AB | AA | A9 | A8 | IE |
| ... | . . . | | | | | | | | ... |
| A0H | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | P2 |
| ... | . . . | | | | | | | | ... |
| 98H | 9F | 9E | 9D | 9C | 9B | 9A | 99 | 98 | SCON |
| ... | . . . | | | | | | | | ... |
| 90H | 97 | 96 | 95 | 94 | 93 | 92 | 91 | 90 | P1 |
| ... | . . . | | | | | | | | ... |
| 88H | 8F | 8E | 8D | 8C | 8B | 8A | 89 | 88 | TCON |
| ... | . . . | | | | | | | | ... |
| 80H | 87 | 86 | 85 | 84 | 83 | 82 | 81 | 80 | P0 |

Примечание*.* Адрес прямо адресуемых битов может быть записан либо в виде выражения (Название Регистра). (Разряд), например выражение SCON.3 означает адрес третьего разряда регистра SCON, либо в виде абсолютного битового адреса, который для данного бита равен (см. таблицу) 9B. Кроме того, некоторые биты управляющих регистров имеют собственные названия, так например данный бит имеет название TB8

### Регистры специальных функций.

К адресному пространству памяти данных примыкает адресное пространство регистров специальных функций SFR (Special Function Register).

Таблица 3 – Размещение регистров специальных функций в пространстве SFR

|  |  |  |
| --- | --- | --- |
| Адрес | Символ | Наименование |
| 0E0H | \*АСС | Аккумулятор (Accumulator) |
| 0F0H | \*В | Регистр расширитель аккумулятора (Multiplication Register) |
| 0D0H | \*PSW | Слово состояния программы (Program Status Word) |
| 080Н | \*P0 | Порт 0 (SFR P0) |
| 090H | \*Р1 | Порт 1 (SFR P1) |
| 0A0H | \*P2 | Порт 2 (SFR P2) |
| 0B0H | \*P3 | Порт 3 (SFR РЗ) |
| 081H | SP | Регистр указатель стека (Stack Pointer) |
| 083H | DPH | Старший байт регистра указателя данных DPTR (Data Pointer High) |
| 082H | DPL | Младший байт регистра указателя данных DPTR (Data Pointer Low) |
| 08CH | TH0 | Старший байт таймера 0 () |
| 08AH | TL0 | Младший байт таймера 0 () |
| 08DH | TH1 | Старший байт таймера 1 () |
| 08BH | TL1 | Младший байт таймера 1 () |
| 089H | TMOD | Регистр режимов таймеров счетчиков (Timer/Counter Mode Control Register) |
| 088H | \*TCON | Регистр управления статуса таймеров (Timer/Counter Control Register) |
| 0B8H | \*IP | Регистр приоритетов (Interrupt Priority Control Register) |
| 0A8H | \*IE | Регистр маски прерывания (Interrupt Enable Register) |
| 087H | PCON | Регистр управления мощностью (Power Control Register) |
| 098H | \*SCON | Регистр управления приемопередатчиком (Serial Port Control Register) |
| 099H | SBUF | Буфер приемопередатчика (Serial Data Buffer) |

Примечание. Регистры, символ которых отмечен знаком (\*), допускают адресацию своих отдельных бит при использовании команд из группы команд операций над битами.

Адреса, по которым расположены эти регистры, приведены в таблице.

Отметим, что регистры занимают только часть 128-байтового адресного пространства. Те ячейки памяти с адресами 80H-0FFH, которые не заняты регистрами, физически отсутствуют, на кристаллах микроконтроллеров семейства 8051 при обращении к ним можно прочитать лишь код команды возврата.

Регистры специальных функций управляют работой блоков, входящих в микроконтроллер.

– Регистры-защелки SFR параллельных портов P0...P3 - служат для ввода-вывода информации.

– Две регистровые пары с именами TH0, TL0 и TH1, TL1 представляют собой регистры, двух программно-управляемых 16-битных таймеров-счетчиков.

– Режимы таймеров-счетчиков задаются с использованием регистра TMOD, а управление ими осуществляется с помощью регистра TCON.

– Для управления режимами энергопотребления микро-ЭВМ используется регистр PCON.

– Регистры IP и IE управляют работой системы прерываний микро-ЭВМ,

– Регистры SBUF и SCON — работой приемопередатчика последовательного порта.

– Регистр-указатель стека SP в микро-ЭВМ рассматриваемого семейства — восьми битный. Он может адресовать любую область внутренней памяти данных. В отличие от микропроцессора КР580ВМ80, у микро-ЭВМ семейства 8051 стек «растет вверх», т.е. перед выполнением команды PUSH или CALL содержимое SP инкрементируется, после чего производится запись информации в стек. Соответственно при извлечении информации из стека регистр SP декрементируется после извлечения информации. В процессе инициализации микро-ЭВМ после сигнала сброса или при включении питающего напряжения в SP заносится код 07Н. Это означает, что первый элемент стека будет располагаться в ячейке памяти с адресом 08Н.

– Регистр-указатель данных DPTR чаще всего используют для фиксации 16-битного адреса в операциях обращения к внешней памяти программ и данных. С точки зрения программиста он может выступать как в виде одного 16-битного регистра, так и в виде двух независимых регистров DPL и DPH

– Аккумулятор (АСС) является источником операнда и местом фиксации результата при выполнении арифметических, логических операций и ряда операций передачи данных. Кроме того, только с использованием аккумулятора могут быть выполнены операции сдвигов, проверка на нуль, формирование флага паритета и т.п. В распоряжении пользователя имеются 8 регистров общего назначения R0–R7 одного из четырёх возможных банков. При выполнении многих команд в АЛУ формируется ряд признаков операции (флагов), которые фиксируются в регистре PSW.

– Регистр В используется как источник и как приемник при операциях умножения и деления, обращение к нему, как к регистру SFR, производится аналогично аккумулятору.

– При выполнении ряда команд в арифметико-логическом устройстве (АЛУ) формируются признаки операций — флаги, которые фиксируются в регистре PSW.

### Таблица 4 – Перечень флагов, их символические имена и условия формирования

|  |  |  |
| --- | --- | --- |
| Символ | Позиция | Имя и назначение |
| P | PSW.0 | Флаг приоритета. Устанавливается и сбрасывается аппаратурно в каждом цикле команды и фиксирует нечетное/четное число единичных бит в аккумуляторе |
| - | PSW.1 | Не используется |
| OV | PSW.2 | Флаг переполнения. Устанавливается и сбрасывается аппаратурно при выполнении арифметических операций |
| RS0-RS1 | PSW.3 - PSW.4 | Биты выбора используемого банка регистров. Могут быть изменены программным путем  RS0  RS1  Банк  Границы адресов ОЗУ  0  0  0  00H - 07H  1  0  1  08H - 0FH  0  1  2  10H - 17H  1  1  3  18H - 1FH |
| F0 | PSW.5 | Флаг пользователя. Может быть установлен, сброшен или проверен программой пользователя |
| АС | PSW.6 | Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратными средствами при выполнении команд сложения и вычитания и сигнализирует о переносе или заеме в бите 3 аккумулятора |
| C | PSW.7 | Флаг переноса. Устанавливается и сбрасывается как аппаратурно, так и программным путем |

Наиболее “активным” флагом PSW является флаг переноса, который принимает участие и модифицируется в процессе выполнения множества операций, включая сложение, вычитание и сдвиги. Кроме того, флаг переноса (C) выполняет функции “булева аккумулятора” в командах, манипулирующих с битами. Флаг переполнения (OV) фиксирует арифметическое переполнение при операциях над целыми числами со знаком и делает возможным использование арифметики в дополнительных кодах. ALU не управляет флагами селекции банка регистров (RS0, RS1), их значение полностью определяется прикладной программой и используется для выбора одного из четырёх регистровых банков.

В микропроцессорах, архитектура которых опирается на аккумулятор, большинство команд работают с ним, используя неявную адресацию. В Intel 8051 дело обстоит иначе. Хотя процессор имеет в своей основе аккумулятор, он может выполнять множество команд и без его участия. Например, данные могут быть переданы из любой ячейки RDM в любой регистр, любой регистр может быть загружен непосредственным операндом и т.д. Многие логические операции могут быть выполнены без участия аккумулятора. Кроме того, переменные могут быть инкрементированы, декрементированы и проверены без использования аккумулятора. Флаги и управляющие биты могут быть проверены и изменены аналогично.

### Устройство управления и синхронизации.

Кварцевый резонатор, подключаемый к внешним выводам микроконтроллера, управляет работой внутреннего генератора, который в свою очередь формирует сигналы синхронизации. Устройство управления (CU) на основе сигналов синхронизации формирует машинный цикл фиксированной длительности, равной 12 периодам резонатора. Большинство команд микроконтроллера выполняется за один машинный цикл. Некоторые команды, оперирующие с 2-байтными словами или связанные с обращением к внешней памяти, выполняются за два машинных цикла. Только команды деления и умножения требуют четырех машинных циклов. На основе этих особенностей работы устройства управления производится расчёт времени исполнения прикладных программ.

На схеме микроконтроллера к устройству управления примыкает регистр команд (IR). В его функцию входит хранение кода выполняемой команды.

Входные и выходные сигналы устройства управления и синхронизации:

– PSEN - разрешение программной памяти,

– ALE - выходной сигнал разрешения фиксации адреса,

– PROG - сигнал программирования,

– EA - блокировка работы с внутренней памятью,

– VPP - напряжение программирования,

– RST - сигнал общего сброса,

– VPD - вывод резервного питания памяти от внешнего источника,

– XTAL - входы подключения кварцевого резонатора.

## Организация портов ввода вывода микроконтроллера 8051.

### Общие сведения.

– Количество портов - 4. Название - P0...P3, они адресуются как регистры специальных функций.

– Разрядность - 8 с возможностью побитной адресации разрядов.

– Направление обмена информацией через порты - все порты двунаправленные, причем имеется возможность в каждом порту часть разрядов использовать для ввода данных, а часть для вывода.

### Альтернативные функции.

Из-за ограниченного количества выводов корпуса ИМС микроконтроллера, большинство выводов используется для выполнения двух функций - в качестве линий портов и для альтернативных функций

– Порты P0 и Р2 используются при обращении к внешней памяти. При этом на выходах P0 младший байт адреса внешней памяти мультиплексируется с вводимым/выводимым байтом. Выходы Р2 содержат старший байт адреса внешней памяти, если адрес 16-разрядный. При использовании восьмиразрядного адреса портом Р2 можно пользоваться для ввода-вывода информации обычным образом. При обращении к внешней памяти в P0 автоматически заносятся 1 во все биты. Информация в Р2 при этом остается неизменной.

– Порт P3 помимо обычного ввода и вывода информации используется для формирования и приема специальных управляющих и информационных сигналов. Разряды порта (все или частично) при этом могут выполнять следующие альтернативные функции:

Альтернативные функции могут быть активированы только в том случае, если в соответствующие биты порта P3 предварительно занесены 1. Неиспользуемые альтернативным образом разряды могут работать как обычно.

## Таймеры / счетчики микроконтроллеров семейства 8051.

В базовых моделях семейства имеются два программируемых 16-битных таймера/счетчика (T/C0 и T/C1), которые могут быть использованы как в качестве таймеров, так и в качестве счетчиков внешних событий. В первом случае содержимое соответствующего таймера/счетчика (далее для краткости Т/С) инкрементируется в каждом машинном цикле, т.е. через каждые 12 периодов колебаний кварцевого резонатора, во втором оно инкрементируется под воздействием перехода из 1 в 0 внешнего входного сигнала, подаваемого на соответствующий (T0,T1) вывод микро-ЭВМ 8051. Так как на распознавание периода требуются два машинных цикла, максимальная частота подсчета входных сигналов равна 1/24 частоты резонатора. На длительность периода входных сигналов ограничений сверху нет. Для гарантированного прочтения входной сигнал должен удерживать значение 1, как минимум, в течение одного машинного цикла микро-ЭВМ.

Для управления режимами работы Т/С и для организации их взаимодействия с системой прерываний используются два регистра специальных функций (TMOD и TCON), описание которых приведено в таблице 5.

Таблица 5 – Регистр режима работы таймера/счетчика TMOD, TCON

|  |  |  |
| --- | --- | --- |
| Символ | Позиция | Имя и назначение |
| GATE | TMOD.7 для Т/С1 и TMOD.3 для Т/СО | Управление блокировкой. Если бит установлен, то таймер/счетчик "x" разрешен до тех пор, пока на входе "lNTx" высокий уровень и бит управления "TRx" установлен. Если бит сброшен, то Т/С разрешается, как только бит управления "TRx" устанавливается |
| С/Т | TMOD.6 для T/C1 и TMOD.2 для Т/СО | Бит выбора режима таймера или счетчика событий. Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации. Если; установлен, то работает счетчик от внешних сигналов на входе "Tx" |
| M1 | TMOD.5 для T/C1 и TMOD.1 для Т/СО | Режим работы  M1  M0   0  0  Таймер ВЕ48. "TLx" работает как 5-битный предделитель  0  1  16 битный таймер/счетчик. "THx" и "TLx" включен последовательно  1  0  8-битный авто перезагружаемый таймер/счетчик. "THx" хранит значение, которое должно быть перезагружено в "TLx" каждый раз по переполнению  1  1  Таймер/счетчик 1 останавливается. Таймер/счетчик 0: TLO работает как 8-битный таймер/счетчик, и его режим определяется управляющими битами таймера 0. TH0 работает только как 8 битный таймер, и его режим определяется управляющими битами таймера 1 |
| M0 | TMOD.4 для Т/С1 и TMOD.0 для Т/СО |
| TF1 | TCON.7 | Флаг переполнения таймера 1. Устанавливается аппаратно при переполнении таймера/счетчика. Сбрасывается при обслуживании прерывания аппаратно |
| TR1 | TCON.6 | Бит управления таймера 1. Устанавливается, / сбрасывается программой для пуска/останова |
| TF0 | TCON.5 | Флаг переполнения таймера 0. Устанавливается аппаратурно. Сбрасывается при обслуживании прерывания |
| TR0 | TCON.4 | Бит управления таймера 0. Устанавливается / сбрасывается программой для пуска/останова таймера/счетчика |
| IE1 | TCON.3 | Флаг фронта прерывания 1. Устанавливается аппаратно, когда детектируется срез внешнего сигнала INT1. Сбрасывается при обслуживании прерывания |
| IT1 | TCON.2 | Бит управления типом прерывания 1. Устанавливается / сбрасывается программно для спецификации запроса INT1 (срез/низкий уровень) |
| IE0 | TCON.1 | Флаг фронта прерывания 0. Устанавливается по срезу сигнала INT0. Сбрасывается при обслуживании прерывания |
| IT1 | TCON .0 | Бит управления типом прерывания 0. Устанавливается / сбрасывается программно для спецификации запроса INT0 (срез/низкий уровень) |

### 

### Режимы работы таймеров-счетчиков.

Как следует из описания управляющих бит TMOD, для обоих Т/С режимы работы 0, 1 и 2 одинаковы. Режимы 3 для Т/СО и Т/С1 различны. Рассмотрим кратко работу Т/С в каждом из режимов.

– Режим 0 .Перевод любого Т/С в режим 0 делает его похожим на таймер КМ1816ВЕ48 (восьми битный счетчик), к входу которого подключен пяти-битный предделитель частоты на 32. Работу Т/С в режиме 0 на примере T/C1 иллюстрирует рис а. В этом режиме таймерный регистр имеет разрядность 13 бит. При переходе из состояния "все единицы" в состояние "все нули" устанавливается флаг прерывания от таймера TF 1. Входной синхросигнал таймера 1 разрешен (поступает на вход Т/С1), когда управляющий бит TR1 установ­лен в 1 либо управляющий бит GATE (блокировка) равен 0, либо на внешний вывод запроса прерывания INT1 поступает уровень 1. Отметим попутно, что установка бита GATE в 1 позволяет использовать таймер для измерения длительности импульсного сигнала подаваемого на вход запроса прерывания.

– Режим 1. Работа любого Т/С в этом режиме такая же, как и в режиме 0, за исключением того, что таймерный регистр имеет разрядность 16 бит.

– Режим 2. В этом режиме работа организована таким образом, что переполнение (переход из состояния "все единицы" в состояние, "все нули") восьмибитного счетчика TL1 приводит не только к установке флага TF1 (см. рис. б), но и автоматически перезагружает в TL1 содержимое старшего байта (TH 1) таймерного регистра, которое предварительно было задано программным путем. Перегрузка оставляет содержимое TH1 неизменным. В режиме 2 Т/С0 и Т/С1 также работают совершенно одинаково.

– Режим 3. В режиме 3 Т/С0 и Т/С1 работают по-разному. Т/С1 сохраняет неизменным свое текущее содержимое. Иными словами, эффект такое же как и при сбросе управляющего бита TR1 в 0. Работу Т/С0 иллюстрирует рис. в. В режиме 3 TL0 и TH0 функционируют как два независимых восьмибитных счетчика. Работу TL0 определяют управляющие биты Т/С0 (С/Т, GATE TR0), входной сигнал INT0 и флаг переполнения TF0. Работу TH0, который может выполнять только функции таймера (подсчёт машинных циклов микро-ЭВМ), определяет управляющий бит TR1. При этом TH0 использует флаг переполнения TF1. Режим 3 используется в тех случаях, когда требуется наличие дополнительного восьми битного таймера или счетчика событий. Можно считать, что в этом режиме микро-ЭВМ 8051 имеет в своем составе три таймера/счетчика. В случае же, если Т/С0 используется в режиме 3, Т/С1 может быть или выключен, или переведен в режим 0, 1 или 2, или может быть использован последовательным портом в качестве генератора частоты передачи.

В модернизированных моделях микроконтроллеров семейства MCS-51 может иметься третий таймер счетчик T/C2 и (или) блок программных счетчиков PCA, которые тоже могут быть использованы для отсчета временных интервалов.

## Последовательный порт микроконтроллера 8051.

Через универсальный асинхронный приемопередатчик UART (Universal Asynchronous Receiver-Transmitter) осуществляются прием и передача информации, представленной последовательным кодом (младшими битами вперед), в полном дуплексном режиме обмена. В состав приемопередатчика, называемого часто последовательным портом входят принимающий и передающий сдвигающие регистры, а также специальный буферный регистр (SBUF) приемопередатчика.

Кроме того, работой последовательного порта управляют два служебных регистра:

– Регистр управления/статуса приемопередатчика SCON

– Бит SMOD регистра управления мощностью PCON

Запись байта в буфер приводит к автоматической переписи байта в сдвигающий регистр передатчика и инициирует начало передачи байта. Наличие буферного регистра приемника позволяет совмещать операцию чтения ранее принятого байта с приемом очередного. Но если к моменту окончания приема байта предыдущий не был считан из SBUF, то он будет потерян.

Последовательный порт 8051 может работать в четырех различных режимах.

– Режим 0. Информация и передается, и принимается через вывод входа приемника (RXi TXi). Принимаются или передается 8 бит данных. Через вывод выхода передатчика (TXD; выдаются импульсы сдвига, которые сопровождают каждый бит. Частота передачи бита информации равна 1/12 частоты кварцевого резонатора

– Режим 1. В этом режиме передаются через вывод TXD или принимаются через RXD 10 бит информации: старт-бит (0), 8 бит данных и стоп-бит (1) при приеме информации в бит RB8 регистра управления/статуса приемопередатчика SCON заносятся стоп-бит Скорость приема/передачи — величина переменная и задается таймером.

– Режим 2. В этом режиме через вывод TXD передаются или через RXD принимаются 11 бит информации: старт-бит, 8 бит данных, программируемый девятый бит и стоп-бит. При передаче девятый бит данных может принимать значение 0 или 1 или, например, для повышения достоверности передачи путем контроля по четности в него может быть помещено значение признака паритета из слова состояния программы (PSW.0). При приеме девятый бит данных помещается в бит RB8 SCON, а стоп-бит, в отличие от режима 1, теряется. Частота приема/передачи выбирается программой и может быть равна либо 1/32, либо 1/64 частоты резонатора в зависимости от управляющего бита SMOD.

– Режим 3. совпадает с режимом 2 во всех деталях, за исключением частоты приема/передачи, которая является величиной переменной и задается таймером.

Во всех случаях передача инициализируется инструкцией, в которой данные перемещаются в SBUF. Прием инициализируется при обнаружении перепада из 1 в 0 на входе приемника. При этом в режиме 0 этот переход должен сопровождаться выполнением условий R1 = 0 и REN= 1 (см. табл. 8), а для остальных режимов - REN = 1.

## Регистр управления/статуса приемопередатчика SCON.

Управление режимом работы приемопередатчика осуществляется через специальный регистр с символическим именем SCON. Этот регистр содержит не только управляющие биты, определяющие режим работы последовательного порта, но и девятый бит принимаемых или передаваемых данных (RB8 и ТВ8) и биты прерывания приемопередатчика (R1 и Т1).

Прикладная программа путем загрузки в старшие биты регистра SCON двухбитного кода определяет режим работы приемопередатчика. Во всех четырех режимах работы передача инициализируется любой командой, в которой буферный регистр SBUF указан как получатель байта. Как уже отмечалось, прием в режиме 0 осуществляется при условии, что R1 = 0 и REN = 1, в остальных режимах - при условии, что REN = 1.

В бите ТВ8 программно устанавливается значение девятого бита данных, который будет передан 8 режиме 2 или 3. В бите RB8 в этих режимах фиксируется девятый принимаемый бит данных. В режиме 1 в бит RB8 заносится стоп-бит. В режиме 0 бит RB8 не используется.

Флаг прерывания передатчика ТI устанавливается аппаратно в конце периода передачи стоп-бита во всех режимах. Соответствующая подпрограмма обслуживания прерывания должна сбрасывать бит TL.

Флаг прерывания приемника RI устанавливается аппаратно в конце периода приема восьмого бита данных в режиме 0 и в середине периода приема стоп-бита в режимах 1, 2 и 3. Подпрограмма обслуживания прерывания должна сбрасывать бит RI.

Таблица 6 – Функциональное назначение бит регистра управления/статуса приемопередатчика SCON

|  |  |  |
| --- | --- | --- |
| Символ | Позиция | Имя и назначение |
| SM0 | SCON.7 | Биты управления режимом работы приемопередатчика. Устанавливаются/сбрасываются программно см. примечание 1  SM0  SM1  Режим работы приемопередатчика  0  0  Сдвигающий регистр расширения ввода/вывода  0  1  8 битовый приемопередатчик, изменяемая скорость передачи  1  0  9 битовый приемопередатчик. Фиксированная скорость передачи  1  1  9 битовый приемопередатчик, изменяемая скорость передачи |
| SM1 | SCON.6 |
| SM2 | SCON.5 | Бит управления режимом приемопередатчика. Устанавливается программно для запрета приема сообщения, в котором девятый бит имеет значение 0 |
| REN | SCON.4 | Бит разрешения приема. Устанавливается/сбрасывается программно для разрешения/запрета приема последовательных данных |
| TB8 | SCON. 3 | Передача бита 8. Устанавливается/сбрасывается программно для задания девятого передаваемого бита в режиме 9-битового передатчика |
| RB8 | SCON.2 | Прием бита 8. Устанавливается/сбрасывается аппаратно для фиксации девятого принимаемого бита в режиме 9-битового приемника |
| TI | SCON. 1 | Флаг прерывания передатчика. Устанавливается аппаратно при окончании передачи байта. Сбрасывается программно после обслуживания прерывания |
| RI | SCON.0 | Флаг прерывания приемника. Устанавливается аппаратно при приеме байта. Сбрасывается программно после обслуживания прерывания |

### 

### Скорость приема/передачи информации через последовательный порт.

Скорость приема/передачи, т.е. частота работы приемопередатчика в различных режимах, определяется различными способами.

В режиме 0 частота передачи зависит только от резонансной частоты кварцевого резонатора f РЕЗ:

f=fРЕЗ/12.

За машинный цикл последовательный порт передает один бит информации. В режимах 1, 2 и 3 скорость приема/передачи зависит от значения управляющего бита SMOD в регистре специальных функций PCON.

Таблица 7 – Регистр управления мощностью PCON

|  |  |  |
| --- | --- | --- |
| Символ | Позиция | Наименование и функция |
| SMOD | PCON.7 | Удвоенная скорость передачи. Если бит установлен в 1, то скорость передачи вдвое больше, чем при SMOD = 0. По сбросу SMOD = 0. |
|  | PCON.6 | Не используется |
|  | PCON.5 | Не используется |
|  | PCON.4 | Hе используется |
| GF1 GF0 | PCON.3 PCON.2 | Флаги, специфицируемые пользователем (флаги общего назначения) |
| PD | PCON.1 | Бит пониженной мощности. При установке бита в 1 микро-ЭВМ переходит в режим пониженной потребляемой мощности |
| IDL | PCON.0 | Бит холостого хода. Если бит установлен в 1, то микро-ЭВМ переходит в режим холостого хода |

Примечание. При одновременной записи 1 в PD и IDL бит PD имеет преимущество. Сброс содержимого PCON выполняется путем загрузки в него кода 0XXX0000.

В режиме 2 частота передачи определяется выражением

f =2SMODfРЕЗ/64.

Иными словами, при SMOD = 0 частота передачи равна 1/64 частоты fРЕЗ, а пои SMOD = 1 — 1/32 частоты fРЕЗ.

В режимах 1 и 3 в формировании частоты передачи, кроме управляющего бита SMOD, принимает участие таймер 1. При этом частота передачи f зависит от частоты переполнения fOVLT и определяется следующим образом:

f=2SMODfOVTL1/32

Прерывание от таймера 1 в этом случае должно быть заблокировано. Сам же таймер может работать как в режиме таймера, так и в режиме счетчика. Номер режима (0, 1, 2) роли не играет. Наиболее типично использование его в режиме таймера с автоперезагрузкой (старшая тетрада TMOD = 0010В). При этом частота передачи определяется выражением:

f = 2SMODfРЕЗ/(32х12х(256 —ТН1)).

Отметим, что скорости приема и передачи могут различаться.

Предельно низких частот приемопередачи можно достичь при использовании таймера в режиме 1 (16-битный таймер) и разрешении прерываний от таймера (старший полубайт TMOD = 0001В). Перезагрузка 16-битного таймера должна осуществляться программным путем.

## Система прерываний микроконтроллера 8051.

Внешние прерывания INT 0 и INT 1 могут быть вызваны либо уровнем, либо переходом сигнала из 1 в 0 на входах 8051 в зависимости от значений управляющих бит IT0 и IT1 в регистре TCON. От внешних прерываний устанавливаются флаги IE0 и IE1 в регистре TCON, которые инициируют вызов соответствующей программы обслуживания прерывания. Сброс этих флагов выполняется аппаратно только в том случае, если прерывание было вызвано по переходу (срезу) сигнала. Если же прерывание вызвано уровнем входного сигнала, то сбросом флага I должна управлять соответствующая подпрограмма обслуживания прерывания путем воздействия на источник прерывания с целью снятия им запроса.

Флаги запросов прерывания от таймеров TF0 и TF1 сбрасываются автоматически при передаче управления подпрограмме обслуживания. Флаги запросов прерывания RI и TI устанавливаются блоком управления приемопередатчика аппаратно, но сбрасываться должны программным путем.

Прерывания могут быть вызваны или отменены программой, так как все названные флаги программно доступны и могут быть установлены/ сброшены программой с тем же результатом, как если бы они были установлены/сброшены аппаратными средствами.

В блоке регистров специальных функций есть два регистра, предназначенных для управления режимом прерываний IE и уровнями приоритета IP. Возможность программной установки/сброса любого управляющего бита в этих двух регистрах делает систему прерываний 8051 исключительно гибкой.

В более сложных модификациях микроконтроллеров семейства MCS-51 количество периферийных устройств увеличено, что приводит к необходимости использовать один вектор прерывания для нескольких устройств (разделение подпрограмм обслуживания прерываний в этом случае необходимо реализовать программно), либо добавить еще два регистра - режима (маски) (таблица 8) и приоритета прерываний (таблица 9).

Таблица 8 – Регистр масок прерывания (IE)

|  |  |  |
| --- | --- | --- |
| Символ | Позиция | Имя и назначение |
| EA | IE.7 | Снятие блокировки прерывания. Сбрасывается, программно для запрета всех прерываний независимо от состояний IE.4 - IE.0 |
|  | IE.6 | Не используется |
|  | lE.5 | Не используется |
| ES | IE.4 | Бит разрешения прерывания, от приемопередатчика Установка/сброс программой для разрешения/запрета прерываний от флагов TI или RI . |
| ET1 | IE.3 | Бит разрешения прерывания от таймера. Установка/сброс программой для разрешения/запрета прерываний от таймера 1 |
| EX1 | IE.2 | Бит разрешения внешнего прерывания 1. Установка/сброс программой для разрешения/запрета прерывания 1 |
| ET0 | IE.1 | Бит разрешения прерывания от таймера 0. Установка/сброс программой для разрешения/запрета прерываний от таймера 0 . |
| EX0 | IE.0 | Бит разрешения внешнего прерывания 0. Установка/сброс программой для разрешения/запрета прерывания 0 |

### 

### Таблица 10 – Регистр приоритетов прерываний (IP)

|  |  |  |
| --- | --- | --- |
| Символ | Позиция | Имя и назначение |
| - | IP.7 - IP.5 | Не используется |
| PS | IP.4 | Бит приоритета приемопередатчика. Установка/сброс программой для присваивания прерыванию от приемопередатчика высшего/низшего приоритета |
| РТ1 | IP.3 | Бит приоритета таймера 1. Установка/сброс программой для присваивания прерыванию от таймера 1 высшего/низшего приоритета |
| РХ1 | IP.2 | Бит приоритета внешнего прерывания 1. Установка/сброс программой для присваивания высшего/низшего приоритета внешнему прерыванию INT1 |
| РТ0 | IP.1 | Бит приоритета таймера 0. Установка/сброс программой для присваивания прерыванию от таймера 0 высшего/низшего приоритета |
| РХ0 | IP.0 | Бит приоритета внешнего прерывания 0. Установка/сброс программой для присваивания высшего/низшего приоритета внешнему прерыванию INT0 |

### 

### Выполнение подпрограммы прерывания.

Система прерываний формирует аппаратный вызов (LCALL) соответствующей подпрограммы обслуживания, если она не заблокирована одним из следующих условий:

– в данный момент обслуживается запрос прерывания равного или высокого уровня приоритета;

– текущий машинный цикл — не последний в цикле выполняемой команды;

– выполняется команда RETI или любая команда, связанная с обращением к регистрам IЕ или IP

Отметим, что если флаг прерывания был установлен, но по одному из указанных выше условий не получил обслуживания и к моменту окончания блокировки уже сброшен, то запрос прерывания теряется и нигде не запоминается.

По аппаратно сформированному коду LCALL система прерывания помещает в стек только содержимое счетчика команд (PC) и загружает в него адрес вектора соответствующей подпрограммы обслуживания. По адресу вектора должна быть расположена команда безусловной передачи управления (JMP) к начальному адресу подпрограммы обслуживания прерывания. В случае необходимости она должна начинаться командами записи в стек (PUSH) слова состояния программы (PSW), аккумулятора, расширителя, указателя данных и т.д. и должна заканчиваться командами восстановления из стека (POP). Подпрограммы обслуживания прерывания должны завершаться командой RETI, по которой в счетчик команд перезагружается из стека сохраненный адрес возврата в основную программу. Команда RET также возвращает управление прерванной основной программе, но при этом не снимут блокировку прерываний, что приводит к необходимости иметь программный механизм анализа окончания процедуры обслуживания данного прерывания.

## 

## Работа с внешней памятью микроконтроллера 8051.

Обращения к внешней памяти подразделяются на обращения к внешней памяти программ и обращения к внешней памяти данных. В первом случае для формирования сигнала, активирующего ПЗУ с программой, используется сигнал PSEN, во втором — сигналы RD и WR, активизирующие ОЗУ с данными.

Если используется 16-битовый адрес, старшие восемь бит выводятся через порт Р2 , где они сохраняются в течение всего цикла обращения к внешней памяти. Отметим, что выходные каскады порта Р2 имеют внутреннюю нагрузку, несколько отличающуюся от Р1 и РЗ , благодаря чему в SFR Р2 при выводе адресной информации вовсе не обязательно защелкивать все единицы. Добавим также, что при выводе адресной информации информация из SFR Р2, хотя и не присутствует на выводах микроЭВМ, но и не теряется, восстанавливаясь на них после окончания обращений к внешней памяти (если в процессе этих обращений SFR Р2 не был модифицирован).

Если при обращении к внешней памяти данных используется восьми битный адрес, то на выводах порта остается та же информация, которая там была до начала обращения к внешней памяти. Это позволяет организовать постраничную адресацию внешней памяти данных.

Как уже отмечалось, на выводах порта P0 младший байт адреса мультиплексируется с данными. Сигналы адреса/ данных задействуют оба полевых транзистора выходного каскада порта P0. Таким образом, в этом случае выводы P0 уже не являются выводами с открытым стоком и не требуют внешних нагрузочных элементов.

Сигнал ALE используется для фиксации младшего байта адреса во внешнем регистре-защелке. Адресная информация достоверна в момент окончания сигнала ALE.

Выводимый в цикле записи байт заносится в P0 непосредственно перед активацией сигнала WR и остается неизменным до окончания этого сигнала. В цикле чтения данные на выводах P0 для достоверного считывания должны быть установившимися к моменту окончания сигнала RD.

Во время обращения к внешней памяти CPU записывает 0FFH в SFR P0, уничтожая, таким образом, хранимую там информацию. Таким образом, использовать для записи порт P0 при работе с внешней памятью надо с известной долей осторожности.

Обращение к внешней памяти программ возможно в двух случаях:

– когда сигнал ЕА активен, т.е. имеет нулевой уровень,

– когда программный счетчик РС содержит число больше 0FFH.

Следовательно, при использовании микро-ЭВМ, не имеющей встроенного ПЗУ или не использующей его, на входе ЕА должен присутствовать сигнал с нулевым уровнем.

Когда CPU работает с внешней памятью программ, все линии порта Р2 используются для вывода старшего байта адреса и не могут быть использованы для обычного ввода\вывода информации. При этом, как отмечалось выше, в SFR Р2 может быть занесена любая информация — адресная информация, выводимая через Р2, не зависит от состояния его SFR.

## Режимы микроконтроллера 8051 с пониженным энергопотреблением.

Во многих вариантах применения микро-ЭВМ энергопотребление является одним из основных параметров. В этих случаях целесообразно использовать КМОП версии микро-ЭВМ. В них предусмотрены дополнительные возможности снижения энергопотребления, отсутствующие в стандартных n-МОП изделиях. Ранее выпускались и n-МОП версии микро-ЭВМ, имевшие режимы работы с пониженным энергопотреблением. В настоящее время их выпуск почти повсеместно прекращен. Однако время от времени они все же будут попадать в руки радиолюбителей, поэтому трудно гарантировать наличие или отсутствие этих режимов в тех или иных конкретных n-МОП изделиях.

МКОП микро-ЭВМ имеют два режима с пониженным потреблением тока: режим холостого хода (XX) и режим выключенного напряжения питания (ВНП). Последний иногда называют режимом пониженного энергопотребления. По мнению авторов настоящей статьи, это вряд ли корректно, так как режим XX также характеризуется пониженным энергопотреблением в сравнении с обычным режимом работы микро-ЭВМ. дословно второй режим, называемый в англоязычной литературе режимом "Power Down Mode", можно перевести так, как указано выше. Этот перевод корректнее отражает суть режима - в нем допустимо вообще отключить от микро-ЭВМ питающее напряжение, подаваемое на вход Vcc (вывод 40 микро-ЭВМ). для сохранения содержимого внутреннее ОЗУ в этом случае нужно запитать от резервного источника. Резервное питающее напряжение подают через вход RST (вывод 9).

В режиме XX (IDL = 1) генератор микро-ЭВМ работает, тактовые сигналы поступают на систему прерываний, последовательный порт и таймеры/счетчики. Все регистры сохраняют свое содержимое, на выводах всех портов удерживаются логические состояния, которые были на них в момент перехода в режим XX. Однако сигнал генератора синхронизирующий CPU, отключается. В режиме ВНП (PD = 1) генератор останавливается. Прекращается тактирование не только CPU, но и последовательного порта, таймеров/ счетчиков, системы прерываний. Как и в режиме XX, состояние регистров, резидентного ОЗУ и выводов портов остается неизменным.

Режимы XX и ВНП активизируются при установке соответствующих битов в специальном функциональном регистре-регистре управления мощностью PCON. Адрес этого регистра 87Н, его описание приведено в табл.

Отметим следующие особенности этого регистра. Если одновременно установлены в 1 биты IDL и PD, то последний имеет преимущество - микро-ЭВМ переходит в режим ВНП. Содержимое регистра PCON после сброса - 0ХХХ0000.

В n-МОП версиях микроЭВМ регистр PCON содержит, как правило, только SMOD. Остальные четыре бита присутствуют только в КМОП устройствах. Пользовательские программы никогда не должны заносить 1 в незанятые биты (PCON4 - PCON6), так как они могут использоваться в следующих модификациях микроЭВМ. Последнее распространяется на все регистры и на все адреса в области регистров специальных функций, которые не заняты регистрами. Разработчики предполагают использовать их в новых изделиях. Однако для достижения совместимости уже разработанного программного обеспечения с новыми изделиями дополнительные возможности последних будут включаться установкой в 1 битов в соответствующих регистрах. Поэтому пользовательская программа, устанавливающая в 1 неиспользуемые биты, будет нормально работать на микро-ЭВМ, имеющихся в наличии сегодня, но вовсе не обязательно будет работать на новых микро-ЭВМ. несмотря на их полную программную совместимость с семейством 8051.

### 

### Режим ХХ.

В этот режим микро-ЭВМ переводится любой командой, устанавливающей в 1 бит PCON.0. Она оказывается последней в цепочке выполняемых команд: в режиме XX выполнение программы приостанавливается ,так как на CPU перестает поступать сигнал тактового генератора. Однако содержимое внутреннего ОЗУ и регистров специальных функций остается неизменным, выводы портов удерживают значения, которые были на них до перехода в режим XX, на таймеры/счетчики, приемопередатчик и на систему прерываний продолжают поступать тактовые сигналы. На выводах ALE и PSEN устанавливаются сигналы единичного уровня.

Состояние выводов портов зависит от типа ОЗУ, с которым микро-ЭВМ обменивалась информацией перед тем как перейти в режим XX. При работе с внутренним ОЗУ на выводах портов присутствуют данные из соответствующих SFR (естественно, если порт в режиме вывода информации). При работе с внешним ОЗУ выводы порта 0 переходят в высокоимпедансное состояние, а на выводах порта 2. сохраняется адресная информация. На выводах портов 1 и 3 присутствуют данные из SFR портов.

Вывести микро-ЭВМ из режима XX можно двумя способами. Так, вызов любого из прерываний приведет к аппаратному стиранию бита PCON.0, прекращающему XX. Прерывание будет обслужено, и очередной после RETI выполняемой командой будет та, которая следует за командой, приведшей к переходу микро-ЭВМ в режим XX. флаги GF0 и GF1 могут использоваться для индикации того, произошло ли прерывание во время нормальной работы или во время XX. Например, команда, запускающая этот режим, может также устанавливать один или оба флага. Когда XX прекращен прерыванием, сервисная программа прерывания может проверять состояние флагов.

Другой способ прекращения XX - с помощью аппаратного сброса. Поскольку синхрогенератор продолжает работать, аппаратный сброс должен поддерживаться в активном состоянии, только в течение двух машинных циклов (24 периодов колебаний).

Сигнал сброса стирает бит PCON.0. В этот момент CPU возобновляет выполнение программы с самого начала. Как показано на рис. 14, перед началом отработки алгоритма внутреннего запуска могут иметь место два или три машинных цикла выполнения программы. Встроенное в микросхему устройство в это время препятствует доступу к внутреннему ОЗУ, но доступ к выводам порта не ограничен. Чтобы исключить возможность появления неопределенных выходных сигналов на выводах порта, команда, следующая за вызывающей XX, не должна быть командой, записывающей информацию в SFR порта или во внешнее ОЗУ данных. Напомним, что после аппаратного сброса содержимое SFR переопределяется.

### 

### Режим ВНП.

Команда, устанавливающая в 1 бит PCON.1, переводит микро-ЭВМ в режим ВНП. В нем генератор микро-ЭВМ, как отмечалось, останавливается, прекращает функционирование не только CPU, но и таймеры/счетчики, приемопередатчик, система прерываний. При наличии основного или резервного источника питающего напряжения встроенное ОЗУ и регистры SFR сохраняют свое содержимое. Состояние портов не отличается от состояния при переходе в режим XX (см. выше). Однако в отличие от режима XX на выводах ALE и PSEN устанавливаются сигналы с нулевым уровнем.

Единственный способ выйти из этого режима - аппаратный сброс. Он переопределяет содержимое всех SFR, но не меняет содержимого встроенного ОЗУ.В рассматриваемом режиме напряжение питания Vcc может снижаться вплоть до 2 В. Следует, однако, позаботиться, чтобы снижалось оно не раньше перехода микроЭВМ в режиме ВНП и восстанавливалось до прежнего значения прежде, чем она выйдет из этого режима. Сигнал сброса, возвращающий микро-ЭВМ в обычный режим, не должен подаваться раньше, чем Vcc достигнет своего рабочего значения, и должен поддерживаться в активном состоянии достаточно долго, чтобы генератор успел запуститься и его колебания стабилизировались (обычно - не менее 10 мс).

**Система команд**

**Система команд микроконтроллера приведена в приложении 1**

2 Монтаж системы сигнализации на охраняемом объекте

2.1 Планирование разводки кабеля и установка разнотипных датчиков

**Монтаж охранных извещателей.**

Выбор типов охранных и охранно - пожарных извещателей, их количества, определение мест установки и методов монтажа должны определяться в соответствии с требованиями действующих нормативных документов, с учетом физико - химических свойств веществ и материалов, используемых в защищаемом помещении (объекте): видом и значимостью охраняемого объекта, принятой тактикой охраны, объектовой помеховой обстановкой, размерами и конструкцией блокируемых элементов, техническими характеристиками извещателей. При этом должно быть исключено образование непросматриваемых ("мертвых") зон.

Магнитоконтактные извещатели предназначены для блокировки на открывание дверей, окон, люков, витрин и других подвижных конструкций. Их устанавливают, как правило, в верхней части блокируемого элемента, со стороны охраняемого помещения на расстоянии 200 мм от вертикальной или горизонтальной, в зависимости от типа магнитоконтактного извещателя, линии раствора блокируемого элемента. При этом геркон извещателей предпочтительно устанавливать на неподвижной части конструкции (плинтусе, дверной раме), а магнит - на подвижной части (двери, оконной раме).

При блокировке внутренних дверей магнитоконтактные извещатели, в зависимости от типа, должны устанавливаться с внутренней стороны дверей, а при необходимости - с обеих сторон, с включением извещателей в разные шлейфы сигнализации.

Выключатели путевые конечные предназначены для блокировки на открывание строительных конструкций, имеющих значительные массу и линейные размеры (ворота, погрузочно - разгрузочные люки и т.п.). Выключатели следует устанавливать на наиболее массивных деталях блокируемой конструкции на кронштейнах.

Корпуса или основания выключателей должны быть заземлены. Крепление выключателей на заземленных металлических панелях не освобождает от необходимости присоединения заземляющего провода.

Поверхностные ударно - контактные извещатели предназначены для блокировки остекленных конструкций, расположенных не ближе 5 м от проезжей части улицы.

Монтаж извещателей следует производить со стороны охраняемого помещения. Места расположения составных частей извещателей определяются количеством, взаимным расположением и площадью блокируемых стеклянных полотен.

Крепление извещателя к поверхности стеклянного полотна производится клеем.

Блокировка остекленных конструкций алюминиевой фольгой производится при наличии на охраняемом объекте вибрационных нагрузок или автотранспортных помех. Фольгу следует наклеивать по периметру блокируемого стеклянного полотна с внутренней стороны обвязки масляной краской, лаком, грунтом. Блокировка фольгой должна обеспечивать защиту конструкций как от разрушения стекла, так и от извлечения стекла из обвязки (либо его поворота в обвязке) без разрушения.

При блокировке проемов из профилированного стекла или стеклоблоков фольгу следует приклеивать через середину стеклоблока параллельно контурным линиям проема с шагом не более 200 мм. Приклейка фольги к поверхности стекла должна производиться при положительных температурах окружающего воздуха. Соединение фольги со шлейфом сигнализации следует выполнять гибкими проводниками.

После приклеивания фольги на нее необходимо нанести краску, при этом полоса краски должна выступать за края фольги не менее чем на 3 мм.

"П"-образная наклейка фольги (только верхняя и боковые стороны обвязки) не допускается.

После выполнения всех монтажных работ по наклейке фольги на остекленные конструкции следует с помощью омметра проверить ее целостность.

При блокировке некапитальных строительных конструкций "на пролом" провод ПЭЛ, ПЭВ или аналогичный диаметром 0,18 - 0,25 мм должен прокладываться с внутренней стороны конструкций по всей площади параллельно контурным линиям и крепиться скобами с шагом крепления 200 мм. Расстояние между длинными сторонами блокирующего провода при открытом или скрытом способе прокладки должно быть не более 200 мм.

При открытом способе прокладки провод должен быть защищен от механических повреждений фанерой, оргалитом или другими аналогичными материалами.

При скрытом способе прокладки провод должен укладываться в штробы с последующей клеевой шпаклевкой и закрашиванием. Глубина и ширина штроба должны быть не менее двух диаметров прокладываемого провода.

Блокировку зарешеченных проемов следует выполнять обвиванием предварительно окрашенных горизонтальных и вертикальных прутьев двойным гибким проводом для исключения возможности закорачивания блокированных участков. Прокладываемые провода должны повторять конфигурацию решетки. После блокировки провода и решетка окрашиваются вновь.

Переход провода с одного прута решетки на другой следует производить по деревянной обвязке рамы скрытым способом.

Ячейки более 200 x 100 мм и решетки из прутьев диаметром менее 10 мм блокировать указанным способом не допускается.

Монтаж емкостных, радиоволновых, ультразвуковых, оптико - электронных и комбинированных извещателей должен производиться на жестких, устойчивых к вибрации опорах (капитальные стены, колонны, столбы и т.п.), с помощью юстировочных узлов, кронштейнов или подставок и исключать возможность ложного срабатывания извещателей по этой причине.

В защищаемой зоне, а также вблизи ее на расстояниях, указанных в технической документации, не должно быть посторонних предметов, изменяющих зону чувствительности извещателей. При установке в одном помещении нескольких оптико - электронных или радиоволновых извещателей необходимо применять извещатели, имеющие разные частотные литеры. Примеры установки извещателей приведены на рисунках 14, 16, 17, 18, 19.

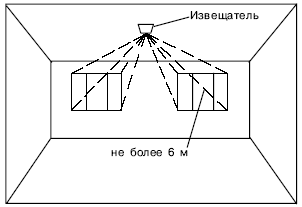


Рисунок 15 – Правильная установки извещателя на потолке

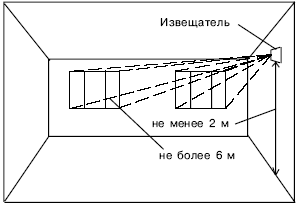


Рисунок 16 – Неправильная установка извещателя на боковой стене

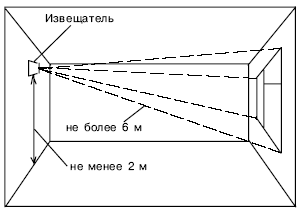


Рисунок 17 – Установка извещателя на противоположной стене

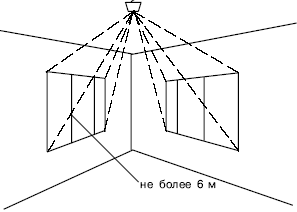


Рисунок 18 – Установки извещателя на потолке для блокировки окон в соседних стенах

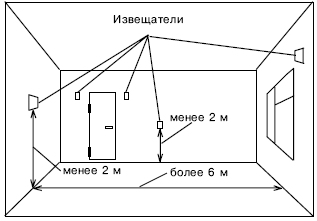


Рисунок 19 – Не рекомендуемые места установки извещателей

Монтаж поверхностных пьезоэлектрических извещателей, предназначенных для блокировки потолочных перекрытий, полов и стен помещений от пролома молотком, ломом или другим тяжелым предметом, производится в местах, защищенных от механических повреждений и доступа посторонних лиц из расчета 75 - 100% охвата охраняемой площади. При этом должно учитываться количество находящихся в охраняемом помещении ценностей.

При монтаже извещателей, блокирующих оконные и дверные проемы в деревянной обвязке, следует применять, как правило, скрытую их установку. (В строго обоснованных случаях допускаются отступления от данного правила.)

**Монтаж электропроводок технических средств сигнализации.**

Монтаж электропроводок технических средств сигнализации должен выполняться в соответствии с проектом (актом обследования), типовыми проектными решениями и с учетом требований СНиП 2.04.09-84, СНиП 3.05.06-85\*, ПУЭ, ВСН 600-81, "Общей инструкции по строительству линейных сооружений городских телефонных сетей", "Инструкции по монтажу сооружений и устройств связи, радиовещания и телевидения".

Соединения и ответвления проводов и кабелей должны производиться в соединительных или распределительных коробках способом пайки или с помощью винтов.

Прокладка незащищенных проводов и кабелей через помещения, которые не подлежат защите, должна производиться скрытым способом или в металлических тонкостенных трубах.

При прокладке скрытым способом провода и кабели сигнализации должны быть проложены в отдельной штробе.

Прокладка проводов и кабелей по стенам внутри охраняемых зданий должна производиться на расстоянии не менее 0,1 м от потолка и, как правило, на высоте не менее 2,2 м от пола. При прокладке проводов и кабелей на высоте менее 2,2 м от пола должна быть предусмотрена их защита от механических повреждений.

Технические средства сигнализации должны быть заземлены

Устройства заземления (зануления) должны выполняться в соответствии с требованиями СНиП 3.05.06-85, ПУЭ, технической документации предприятий – изготовителей.

3. Разработка программного обеспечения для микроконтроллера

3.1 Выбор инструментальных средств (среда Ассемблер)

**Язы́к ассе́мблера** (в отечественных источниках, также автокод) — язык программирования низкого уровня, мнемонические команды которого (за редким исключением) соответствуют инструкциям процессора вычислительной системы. Трансляция программы в исполняемый машинный код производится ассемблером (от англ. assembler - сборщик) - программой-транслятором, которая и дала языку ассемблера его название.

**Содержание языка Ассемблер.**

Команды языка ассемблера один к одному соответствуют командам процессора, фактически, они представляют собой более удобную для человека символьную форму записи (мнемокод) команд и их аргументов. При этом одной команде языка ассемблера может соответствовать несколько команд процессора.

Кроме того, язык ассемблера позволяет использовать символические метки вместо адресов ячеек памяти, которые при ассемблировании заменяются на автоматически рассчитываемые абсолютные или относительные адреса, а также так называемые директивы (команды, не переводящиеся в процессорные инструкции, а выполняемые самим ассемблером).

Директивы ассемблера позволяют, в частности, включать блоки данных, задать ассемблирование фрагмента программы по условию, задать значения меток, использовать макроопределения с параметрами.

Каждая модель (или семейство) процессоров имеет свой набор команд и соответствующий ему язык ассемблера. Наиболее популярные синтаксисы — Intel-синтаксис и AT&T-синтаксис.

Существуют компьютеры, реализующие в качестве машинного язык программирования высокого уровня (Forth, Lisp, Эль-76); фактически, в них он является языком ассемблера.

**Достоинства языка Ассемблер.**

При достаточной квалификации программиста, язык ассемблера позволяет писать самый быстрый и компактный код. Возможно, даже лучше, чем генерируемый трансляторами языков более высокого уровня.

Если код программы достаточно большой, данные, которыми он оперирует, не помещаются целиком в регистрах процессора, то есть частично или полностью находятся в оперативной памяти, то искусный программист, как правило, способен значительно оптимизировать программу по сравнению с высокоуровневыми трансляторами по одному или нескольким параметрам: скорость работы (за счёт оптимизации вычислений и/или более рационального обращения к ОП, перераспределения данных), объём кода (в том числе за счёт эффективного использования промежуточных результатов).

Обеспечение максимального использования специфических возможностей конкретной платформы, что также позволяет создавать более эффективные программы с меньшими затратами ресурсов.

При программировании на языке ассемблера возможен непосредственный доступ к аппаратуре, в том числе портам ввода-вывода, регистрам процессора и др.

Язык ассемблера применяется для создания драйверов оборудования и ядра операционной системы.

Язык ассемблера используется для создания «прошивок» BIOS.

С помощью языка ассемблера создаются компиляторы и интерпретаторы языков высокого уровня, а также реализуется совместимость платформ.

Существует возможность исследования других программ с отсутствующим исходным кодом с помощью дизассемблера.

**Недостатки языка Ассемблер.**

В силу машинной ориентации («низкого» уровня) языка ассемблера человеку сложнее читать и понимать программу на нём по сравнению с языками программирования высокого уровня; программа состоит из слишком «мелких» элементов — машинных команд, соответственно, усложняются программирование и отладка, растёт трудоёмкость, велика вероятность внесения ошибок.

Требуется высокая квалификация программиста. Код на ассемблере, написанный неопытным программистом, обычно оказывается хуже сгенерированного компилятором.

Как правило, меньшее количество доступных библиотек по сравнению с современными индустриальными языками программирования.

Отсутствует переносимость программ на компьютеры с другой архитектурой и системой команд.

**Применение языка Ассемблер.**

Исторически можно рассматривать язык ассемблера как второе поколение языков программирования ЭВМ (если первым считать машинный код). Недостатки языка ассемблера, сложность разработки на нём больших программных комплексов привели к появлению языков третьего поколения — языков программирования высокого уровня (Фортран, Лисп, Кобол, Паскаль, Си и др.). Именно языки программирования высокого уровня и их наследники в основном используются в настоящее время в индустрии информационных технологий. Однако языки ассемблера сохраняют свою нишу, обуславливаемую их уникальными преимуществами в части эффективности и возможности полного использования специфических средств конкретной платформы.

На языке ассемблера пишутся программы или фрагменты программ, для которых критически важны:

быстродействие (драйверы, игры);

объем используемой памяти (загрузочные секторы, встраиваемое (англ. embedded) программное обеспечение, программы для микроконтроллеров и процессоров с ограниченными ресурсами, вирусы, программные защиты).

С использованием программирования на языке ассемблера производятся:

Оптимизация критичных к скорости участков программ, написанных на языке высокого уровня, таком как C++. Это особенно актуально для игровых приставок, имеющих фиксированную производительность, и для мультимедийных кодеков, которые стремятся делать менее ресурсоёмкими и более популярными.

Создание операционных систем (ОС). ОС часто пишут на Си, языке, который специально был создан для написания одной из первых версий UNIX. Аппаратно зависимые участки кода, такие как загрузчик ОС, уровень абстрагирования от аппаратного обеспечения (hardware abstraction layer) и ядро, часто пишутся на языке ассемблера. Ассемблерного кода в ядрах Windows или Linux совсем немного, поскольку авторы стремятся к переносимости и надёжности, но, тем не менее, он присутствует. Некоторые любительские ОС, такие как MenuetOS, целиком написаны на языке ассемблера. При этом MenuetOS помещается на дискету и содержит графический многооконный интерфейс.

Программирование микроконтроллеров (МК) и других встраиваемых процессоров. По мнению профессора Таненбаума, развитие МК повторяет историческое развитие компьютеров новейшего времени. На сегодняшний день для программирования МК весьма часто применяют язык ассемблера. В МК приходится перемещать отдельные байты и биты между различными ячейками памяти. Программирование МК весьма важно, так как, по мнению Таненбаума, в автомобиле и квартире современного цивилизованного человека в среднем содержится 50 микроконтроллеров.

Создание драйверов. Некоторые участки драйверов, взаимодействующие с аппаратным обеспечением, программируют на языке ассемблера. Хотя в целом в настоящее время драйверы стараются писать на языках высокого уровня в связи с повышенными требованиями к надёжности. Надёжность для драйверов играет особую роль, поскольку в Windows NT и Linux драйверы работают в режиме ядра. Одна ошибка может привести к краху системы.

Создание антивирусов и других защитных программ.

Написание трансляторов языков программирования.

Связывание программ на разных языках.

Поскольку на языке ассемблера часто разрабатываются только фрагменты программ, их необходимо связывать с остальными частями программной системы, написанными на других языках программирования.

Это достигается двумя основными способами:

На этапе компиляции — вставка в программу ассемблерных фрагментов (англ. inline assembler) с помощью специальных директив языка (в частности, данный способ поддерживается языками программирования Си и Delphi), в том числе написание функций на языке ассемблера. Способ удобен для несложных преобразований данных, но полноценного ассемблерного кода, с данными и подпрограммами, включая подпрограммы с множеством входов и выходов, не поддерживаемых высокоуровневыми языками, с его помощью сделать нельзя.

На этапе компоновки при раздельной компиляции. Для взаимодействия скомпонованных модулей достаточно, чтобы связующие функции (определённые в одних модулях и использующиеся в других) поддерживали нужные соглашения вызова (англ. calling conventions) и типы данных. Написаны же отдельные модули могут быть на любых языках, в том числе и на языке ассемблера.

**Синтаксис.**

**Синтаксис языка ассемблера определяется системой команд конкретного процессора.**

Типичными командами языка ассемблера являются (большинство примеров даны для Intel-синтаксиса архитектуры x86):

Команды пересылки данных (mov, lea и др.)

Арифметические команды (add, sub, imul и др.)

Логические и побитовые операции (or, and, xor, shr и др.)

Команды управления ходом выполнения программы (jmp, loop, ret и др.)

Команды вызова прерываний (иногда относят к командам управления): int, into

Команды ввода/вывода в порты (in, out)

Для микроконтроллеров и микрокомпьютеров характерны также команды, выполняющие проверку и переход по условию, например:

cbne — перейти, если не равно

dbnz — декрементировать, и если результат ненулевой, то перейти

cfsneq — сравнить, и если не равно, пропустить следующую команду

Инструкции.

Типичный формат записи команд:

[метка:] опкод [операнды] [;комментарий]

где опкод (код операции) — непосредственно мнемоника инструкции процессору. К ней могут быть добавлены префиксы (повторения, изменения типа адресации и пр.).

В качестве операндов могут выступать константы, адреса регистров, адреса в оперативной памяти и пр. Различия между синтаксисом Intel и AT&T касаются в основном порядка перечисления операндов и указания различных методов адресации.

Используемые мнемоники обычно одинаковы для всех процессоров одной архитектуры или семейства архитектур (среди широко известных — мнемоники процессоров и контроллеров x86, ARM, SPARC, PowerPC, M68k). Они описываются в спецификации процессоров. Возможные исключения:

если ассемблер использует кроссплатформенный AT&T-синтаксис (оригинальные мнемоники приводятся к синтаксису AT&T);

если изначально существовало два стандарта записи мнемоник (система команд была наследована от процессора другого производителя).

Например, процессор Zilog Z80 наследовал систему команд Intel i8080, расширил её и поменял мнемоники (и обозначения регистров) на свой лад. Процессоры Motorola Fireball наследовали систему команд Z80, несколько её урезав. Вместе с тем, Motorola официально вернулась к мнемоникам Intel и в данный момент половина ассемблеров для Fireball работает с мнемониками Intel, а половина — с мнемониками Zilog.

**Директивы Ассемблера.**

Программа на языке ассемблера может содержать директивы: инструкции, не переводящиеся непосредственно в машинные команды, а управляющие работой компилятора. Набор и синтаксис их значительно разнятся и зависят не от аппаратной платформы, а от используемого транслятора (порождая диалекты языков в пределах одного семейства архитектур). В качестве «джентельменского набора» директив можно выделить следующие:

-определение данных (констант и переменных),

-управление организацией программы в памяти и параметрами выходного файла,

-задание режима работы компилятора,

-всевозможные абстракции (то есть элементы языков высокого уровня) — от оформления процедур и функций (для упрощения реализации парадигмы процедурного программирования) до условных конструкций и циклов (для парадигмы структурного программирования),

-макросы.

Пример программы

Примеры программы Hello, world! для разных платформ и разных диалектов:

Пример программы для Windows на диалекте MASM

.386

.model flat, stdcall

option casemap:none

include \masm32\include\windows.inc

include \masm32\include\kernel32.inc

includelib \masm32\lib\kernel32.lib

.data

msg db "Hello, world", 13, 10

len equ $-msg

.data?

written dd ?

.code

start:

push -11

call GetStdHandle

push 0

push offset written

push len

push offset msg

push eax

call WriteFile

push 0

call ExitProcess

end start

**Пример программы для Linux/x86 на диалекте NASM.**

SECTION .data

msg: db “Hello, world”,10

len: equ $-msg

SECTION .text

global \_start

\_start: ove ax, len

ove ax, msg

mov ebx, 1 ; stdout

ove ax, 4 ; write(2)

int 0x80

mov ebx, 0

ove ax, 1 ; exit(2)

**Пример программы для MS-DOS на диалекте TASM.**

.MODEL TINY

CODE SEGMENT

ASSUME CS:CODE, DS:CODE

ORG 100h

START:

mov ah,9

mov dx,OFFSET Msg

int 21h

int 20h

Msg DB 'Hello World',13,10,'$'

CODE ENDS

END START int 0x80

**Происхождение и критика термина «язык ассемблера».**

Данный тип языков получил своё название от названия транслятора (компилятора) с этих языков — ассемблера (англ. assembler — сборщик). Название обусловлено тем, что программа «автоматически собиралась», а не вводилась вручную покомандно непосредственно в кодах. При этом наблюдается путаница терминов: ассемблером нередко называют не только транслятор, но и соответствующий язык программирования («программа на ассемблере»).

Использование термина «язык ассемблера» также может вызвать ошибочное мнение о существовании некоего единого языка низкого уровня или хотя бы стандартов на такие языки. При именовании языка ассемблера желательно уточнять, ассемблер для какой архитектуры имеется в виду.

В СССР язык ассемблера ранее называли «автокод».

Ассе́мблер (от англ. assembler — сборщик) — компьютерная программа, компилятор исходного текста программы, написанной на языке ассемблера, в программу на машинном языке.

Как и сам язык (ассемблер), ассемблеры, как правило, специфичны конкретной архитектуре, операционной системе и варианту синтаксиса языка. Вместе с тем существуют мультиплатформенные или вовсе универсальные (точнее, ограниченно-универсальные, потому что на языке низкого уровня нельзя написать аппаратно-независимые программы) ассемблеры, которые могут работать на разных платформах и операционных системах. Среди последних можно также выделить группу кросс-ассемблеров, способных собирать машинный код и исполняемые модули (файлы) для других архитектур и ОС.

Ассемблирование может быть не первым и не последним этапом на пути получения исполняемого модуля программы. Так, многие компиляторы с языков программирования высокого уровня выдают результат в виде программы на языке ассемблера, которую в дальнейшем обрабатывает ассемблер. Также результатом ассемблирования может быть не исполняемый, а объектный модуль, содержащий разрозненные и непривязанные друг к другу части машинного кода и данных программы, из которого (или из нескольких объектных модулей) в дальнейшем с помощью программы-компоновщика («линкера») может быть скомпонован исполнимый файл.

**Архитектура x86.**

**Ассемблеры для DOS.**

Наиболее известными ассемблерами для операционной системы DOS являлись Borland Turbo Assembler (TASM), Microsoft Macro Assembler (MASM) и Watcom Assembler (WASM). Также в своё время был популярен простой ассемблер A86. Интерфейс представлен на рисунке 22.

**Windows.** (рисунок 23)

При появлении операционной системы Windows появилось расширение TASM, именуемое TASM32, позволившее создавать программы для выполнения в среде Windows. Последняя известная версия TASM — 5.3, поддерживающая инструкции MMX, на данный момент включена в Turbo C++ Explorer. Но официально развитие программы полностью остановлено.

Microsoft поддерживает свой продукт под названием Microsoft Macro Assembler. Она продолжает развиваться и по сей день, последние версии включены в наборы DDK. Но версия программы, направленная на создание программ для DOS, не развивается. Кроме того, Стивен Хатчессон создал пакет для программирования на MASM под названием «MASM32».

**GNU и GNU/Linux.**

В состав операционной системы GNU входит пакет binutils, включающий в себя ассемблер gas (GNU Assembler), использующий AT&T-синтаксис, в отличие от большинства других популярных ассемблеров, которые используют Intel-синтаксис.

На рисунке 24 представленGNU Assembler.

**Переносимые ассемблеры.**

Также существует открытый проект ассемблера, версии которого доступны под различные операционные системы и который позволяет получать объектные файлы для этих систем. Называется этот ассемблер NASM (Netwide Assembler).

Yasm — это переписанная с нуля версия NASM под лицензией BSD (с некоторыми исключениями).

flat assembler (fasm) — молодой ассемблер под модифицированной для запрета перелицензирования (в том числе под GNU GPL) BSD-лицензией. Есть версии для KolibriOS, Linux, DOS и Windows; использует Intel-синтаксис и поддерживает инструкции x86-64.

**Архитектуры RISC.**

MCS-51 (Intel 8051) — классическая архитектура микроконтроллера. Для неё существует кросс-ассемблер ASM51, выпущенный корпорацией MetaLink.

Кроме того, многие фирмы — разработчики программного обеспечения, такие как IAR или Keil, представили свои варианты ассемблеров. В ряде случаев применение этих ассемблеров оказывается более эффективным благодаря удобному набору директив и наличию среды программирования, объединяющей в себе профессиональный ассемблер и язык программирования Си, отладчик и менеджер программных проектов.

**AVR.**

На данный момент существуют 2 компилятора производства Atmel (AVRStudio 3 и AVRStudio4). Вторая версия — попытка исправить не очень удачную первую.

В рамках проекта AVR-GCC (он же WinAVR) существует компилятор avr-as (это портированный под AVR ассемблер GNU as из GCC).

Также существует свободный минималистический компилятор avra

Ассемблирование и компилирование

Процесс трансляции программы на языке ассемблера в объектный код принято называть ассемблированием. В отличие от компилирования, ассемблирование — более или менее однозначный и обратимый процесс. В языке ассемблера каждой мнемонике соответствует одна машинная инструкция, в то время как в языках программирования высокого уровня за каждым выражением может скрываться большое количество различных инструкций. В принципе, это деление достаточно условно, поэтому иногда трансляцию ассемблерных программ также называют компиляцией.

3.2 Подпрограмма для микроконтроллера

**Листинг программы необходимой для функционирования системы.**

**См. Приложение 2.**

4 Результаты функционирование системы охранной сигнализации промышленного объекта

4.1 Пакет программного обеспечения для функционирования микроконтроллера

Для функционирования МК необходимо следующее программное обеспечение:

**ОС(Операцио́нная систе́ма).**

Операцио́нная систе́ма, сокр. ОС (англ. operating system) — комплекс управляющих и обрабатывающих программ, которые, с одной стороны, выступают как интерфейс между устройствами вычислительной системы и прикладными программами, а с другой стороны — предназначены для управления устройствами, управления вычислительными процессами, эффективного распределения вычислительных ресурсов между вычислительными процессами и организации надёжных вычислений. Это определение применимо к большинству современных ОС общего назначения.

В логической структуре типичной вычислительной системы ОС занимает положение между устройствами с их микроархитектурой, машинным языком и, возможно, собственными (встроенными) микропрограммами — с одной стороны — и прикладными программами с другой.

Разработчикам программного обеспечения ОС позволяет абстрагироваться от деталей реализации и функционирования устройств, предоставляя минимально необходимый набор функций (см. интерфейс программирования приложений).

В большинстве вычислительных систем ОС являются основной, наиболее важной (а иногда единственной) частью системного ПО. С 1990-х наиболее распространёнными операционными системами являются ОС семейства Microsoft Windows и системы класса UNIX (особенно Linux)

**Функции операционных систем.**

Выполнение по запросу программ тех достаточно элементарных (низкоуровневых) действий, которые являются общими для большинства программ и часто встречаются почти во всех программах (ввод и вывод данных, запуск и остановка других программ, выделение и освобождение дополнительной памяти и др.).

Загрузка программ в оперативную память и их выполнение.

Стандартизованный доступ к периферийным устройствам (устройства ввода-вывода).

Управление оперативной памятью (распределение между процессами, организация виртуальной памяти).

Управление доступом к данным на энергонезависимых носителях (таких как жёсткий диск, оптические диски и др.), организованным в той или иной файловой системе.

Обеспечение пользовательского интерфейса.

Сетевые операции, поддержка стека сетевых протоколов.

Дополнительные функции:

Параллельное или псевдопараллельное выполнение задач (многозадачность).

Эффективное распределение ресурсов вычислительной системы между процессами.

Разграничение доступа различных процессов к ресурсам.

Организация надёжных вычислений (невозможности одного вычислительного процесса намеренно или по ошибке повлиять на вычисления в другом процессе), основана на разграничении доступа к ресурсам.

Взаимодействие между процессами: обмен данными, взаимная синхронизация.

Защита самой системы, а также пользовательских данных и программ от действий пользователей (злонамеренных или по незнанию) или приложений.

Многопользовательский режим работы и разграничение прав доступа (см. аутентификация, авторизация).

Операционные системы могут быть классифицированы по базовой технологии (UNIX-подобные, пост-UNIX/потомки UΝΙΧ), типу лицензии (проприетарная или открытая), развивается ли в настоящее время (устаревшие или современные), по назначению (универсальные, ОС встроенных систем, ОС PDA, ОС реального времени, для рабочих станций или для серверов), а также по множеству других признаков.

Примеры ОС непосредственно для пк в целом:

Acorn(Arthur, ARX, MOS, RISC OS, RISCiX), Atari ST(TOS, MultiTOS,

MiNT), Apple(Apple DOS, GS/OS, Mac OS, IOS), IBM(IBSYS, OS/2, DOS/360, DOS/VSE, PC DOS — OEM-версия MS-DOS, впоследствии дорабатывалась самостоятельно.), Microsoft(MSX-DOS

MS-DOS, Windows 9x, Windows NT, Windows 2000, Windows XP, Windows Server, Windows Vista, Windows 7 (интерфейс представлен на рисунке 20), Windows CE), Novell(NetWare, UnixWare, openSUSE — дистрибутив свободной операционной системы Linux, разработчиком которого является корпорация Novell, Inc.) и другие.

Примеры непосредственно для микроконтроллеров, встраиваемые и ОС реального времени:

AMX OS KADAK, Contiki, eCos, FreeRTOS, INTEGRITY, ITRON, LynxOS, Montavista Linux (рисунок 21), Nucleus, QNX, OS-9 — от Microware, OS-9000 — от Microware, OSA[7] — для микроконтроллеров PIC (Microchip) и AVR (Atmel), OSE от ENEA, OSEK, RDOS, RTEMS первоначальная разработка велась по заказу МО США, сейчас opensource (GPL-like лицензия), RTOS, ThreadX, TRON OS разработчик — Ken Sakamura,

uC/OS-II для микроконтроллеров, scmRTOS[8] — для микроконтроллеров, μClinux, VxWorks, Salvo[9] — для микроконтроллеров.

Для функционирования МК конкретная ос не нужна, необходимо чтобы она поддерживала функционирование нашего микроконтроллера. Примером подходящей ОС может быть Windows, Linux, Unix, Apple.



Рисунок 20 – Интерфейс Windows 7



Рисунок 21 – Интерфейс Linux

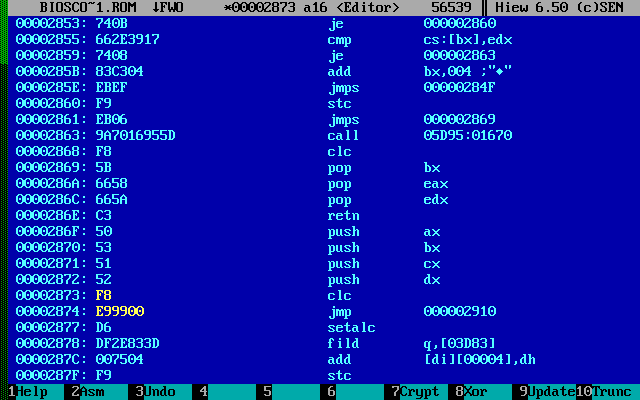


Рисунок 22 – Интерфейс Ассемблер Dos

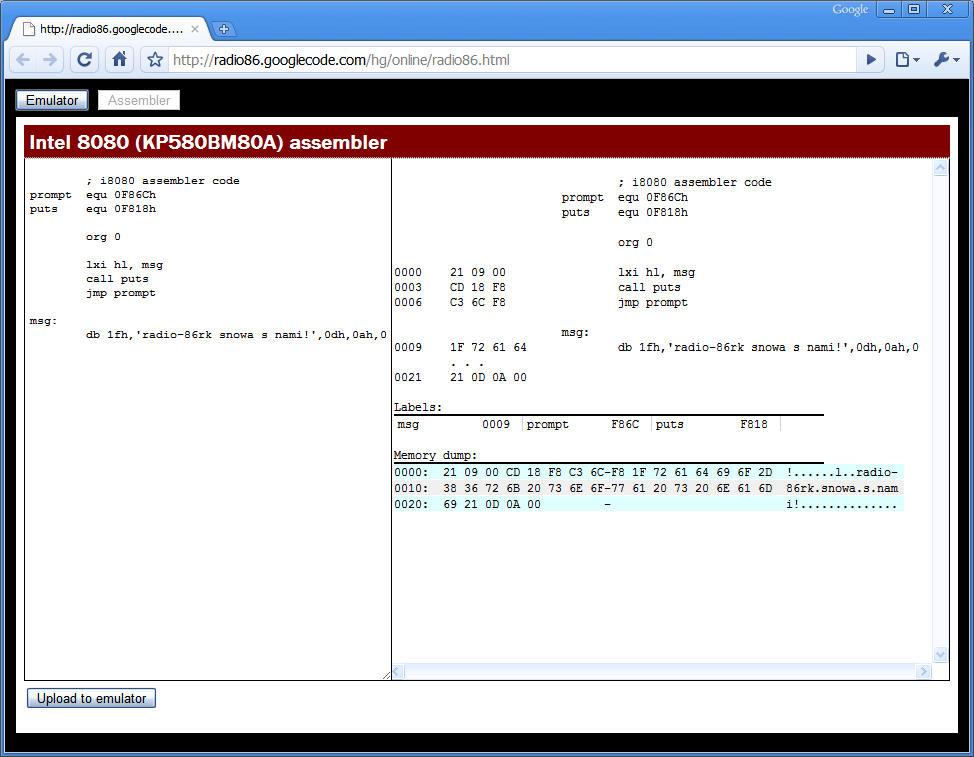


Рисунок 23 – Интерфейс Ассемблер Windows

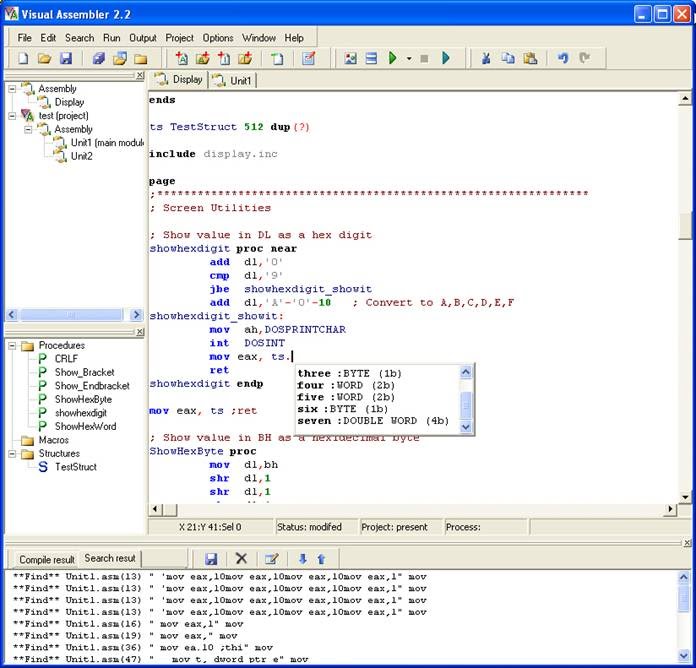


Рисунок 24 – Интерфейс Visual Ассемблер

4.2 Экономическое обоснование

Себестоимость промышленной продукции - это текущие затраты предприятия на про¬изводство и реализацию продукции, выраженные в денежной форме. В России действует по¬становление правительства о составе затрат по производству и реализации продукции (работ, услуг, включаемых в их себестоимость) и о порядке формирования финансовых результатов, учитываемых при налогообложении прибыли.

Наибольший удельный вес во всех расходах предприятия занимают затраты на произ¬водство продукции. Они называются издержками производства. К ним относятся выражен¬ные в стоимостной форме:

– потребляемые в процессе производства средства и предметы труда (амортизация основных

фондов; сырье, материалы, топливо, энергия и т.п.);

– используемые в производстве покупные изделия и полуфабрикаты, а также производствен¬ные услуги других предприятий;

– часть стоимости живого труда (заработная плата).

Многие из этих затрат можно планировать и учитывать в натуральном выражении, т.е. в кг., метрах, тоннах, штуках и т.п. Однако, для полного подсчета всех расходов, их приводят к единому измерению и представляют в денежном выражении (рублях).

Предприятия имеют также затраты по реализации (сбыту) продукции, т.е. осуществляют непроизводственные (коммерческие) расходы (транспортировка, упаковка, хранение, рек¬лама и т.п.).

Вместе производственная себестоимость и коммерческие расходы составляют полную (ком¬мерческую) себестоимость продукции.

Не все издержки предприятия включают в себестоимость выпускаемой продукции. Сюда, например, не относятся расходы непромышленных подразделений предприятия (поликлини¬ка, столовая, клуб и т.п.).

Все затраты предприятия можно разбить на прямые и косвенные:

Прямые затраты - это затраты непосредственно связанные с производством продукции (например затраты на материалы, полуфабрикаты, заработную плату производственного пер¬сонала, расходы по содержанию и эксплуатации оборудования и т.п.).

Эти затраты могут быть непосредственно отнесены на конкретный вид изделий и рас¬пределены между ними на основании технико-экономических расчетов (например, по нор¬мативам).

Косвенные затраты - это затраты, которые не могут быть прямо распределены по объек¬там отнесения затрат (например, административные и управленческие расходы, расходы на освещение, отопление, страхование имущества и т.п.)

При учете и анализе затрат на производство важное значение имеет связь затрат с объе¬мом производства. По этому признаку выделяют постоянные и переменные издержки пред¬приятия.

Постоянные издержки (затраты) остаются неизменными и осуществляются независи¬мо от объема производства. Постоянные издержки иначе называются накладные расходы (арен¬дная плата, оплата административно-управленческого персонала и т.п.).

Переменные издержки (затраты) находятся в прямо пропорциональной зависимости от объема производства (затраты на сырье, материалы, оплата производственного персонала и т.п.),и труда, качество управления, маркетинга и т.п. Она выступает как исходная база формирова¬ния цен. а также оказывает непосредственное влияние на величину прибыли и уровень рента¬бельности.

Расчёт затрат на монтажные, электромонтажные и общестроительные работы произведен с использованием – территориальных единичных расценок (ТЕР); затраты на оборудование и материалы не учтенные ценником рассчитывались исходя из цен по прайс-листам; затраты на пусконаладочные работы рассчитывались по Государственные элементные сметные нормы на пусконаладочные работы (ГЭСНп).

Установка охранной сигнализации хоть и стоит не дёшево, но она даёт вам уверенность в том что, при проникновении нарушителя на охраняемую территорию, сработает сигнализация и по сигналу, поданному на пульт вневедомственной охраны, прибудет наряд милиции. Затраты на её установку могут полностью окупится.

Заключение о представленной охранной системе сигнализации

Была разработана интеллектуальная система охранной сигнализации стационарного промышленного объекта. При этом мы обеспечили ей высокие охранные свойства, небольшую стоимость, надежность, простоту в использовании и минимальные затраты на эксплуатацию. Все перечисленные качества делают нашу систему вполне конкурентоспособной.

Список используемых источников

1. Граф Р., Шиитс В. Энциклопедия электронных схем. - М.: ДМК, 2001 - 385с.;
2. Угрюмов Е. Цифровая схемотехника. – СПБ.: БХВ-Петербург, 2002. – 528с.;
3. Степаненко И.П. Основы микроэлектроники: учеб. пособие для вузов.-2-е изд. перераб. и доп. - М.: Лаборатория базовых знаний, 2000 – 488 с.;
4. [http://www.condor.com.ru](http://www.rmcip.ru)
5. <http://www.micom.net.ru>
6. http://ru.wikipedia.org
7. Андреев Д.В. Программирование микроконтроллеров MCS-51.: Учеб. Пособие – Ульяновск: УлГТУ, 2000 – 88с.
8. Логинов А.В. Программирование микро-ЭВМ семейства МК51: Учеб. Пособие. – СПб.: Балт. гос. тех. ун-т, 1996 – 72с.;
9. Горфинкель В.Я., Купрякова Е.М. "Экономика предприятия". М.: «Юнит», 1996г
10. ГОСТ Р 50775-95 "Системы тревожной сигнализации. Общие требования. Общие положения".
11. ГОСТ Р 50776-95 "Системы тревожной сигнализации. Общие требования. Руководство по проектированию, монтажу и тех обслуживанию".
12. ГОСТ Р 51241-99 "Средства и системы контроля и управления доступом. Общие технические требования и методы испытаний".
13. Инструкция по применению и испытанию средств защиты, используемых в Эл. установках. СПб.: 2004
14. ПУЭ Правила устройства электроустановок. М.: «Энергоатомиздат», 1998.

15 А. Д. Шишкин. Методическое пособие для выполнения дипломного и курсового проектирования. СПб,-2008. изд. РГГМУ.

Приложение А

# Система команд микроконтроллера семейства 8051.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Название команды | Мнемокод | КОП | Т | Б | Ц | Операция |
| Пересылка в аккумулятор из регистра (n=0÷7) | MOV A, Rn | 11101rrr | 1 | 1 | 1 | (A) ← (Rn) |
| Пересылка в аккумулятор прямоадресуемого байта | MOV A, ad | 11100101 | 3 | 2 | 1 | (A) ← (ad) |
| Пересылка в аккумулятор байта из РПД (i=0,1) | MOV A, @Ri | 1110011i | 1 | 1 | 1 | (A) ← ((Ri)) |
| Загрузка в аккумулятор константы | MOV A, #d | 01110100 | 2 | 2 | 1 | (A) ← #d |
| Пересылка в регистр из аккумулятора | MOV Rn, A | 11111rrr | 1 | 1 | 1 | (Rn) ← (A) |
| Пересылка в регистр прямоадресуемого байта | MOV Rn, ad | 10101rrr | 3 | 2 | 2 | (Rn) ← (ad) |
| Загрузка в регистр константы | MOV Rn, #d | 01111rrr | 2 | 2 | 1 | (Rn) ← #d |
| Пересылка по прямому адресу аккумулятора | MOV ad, A | 11110101 | 3 | 2 | 1 | (ad) ← (A) |
| Пересылка по прямому адресу регистра | MOV ad, Rn | 10001rrr | 3 | 2 | 2 | (ad) ← (Rn) |
| Пересылка прямоадресуемого байта по прямому адресу | MOV add, ads | 10000101 | 9 | 3 | 2 | (add) ← (ads) |
| Пересылка байта из РПД по прямому адресу | MOV ad, @Ri | 1000011i | 3 | 2 | 2 | (ad) ← ((Ri)) |
| Пересылка по прямому адресу константы | MOV ad, #d | 01110101 | 7 | 3 | 2 | (ad) ← #d |
| Пересылка в РПД из аккумулятора | MOV @Ri, A | 1111011i | 1 | 1 | 1 | ((Ri)) ← (A) |
| Пересылка в РПД прямоадресуемого байта | MOV @Ri, ad | 0110011i | 3 | 2 | 2 | ((Ri)) ← (ad) |
| Пересылка в РПД константы | MOV @Ri, #d | 0111011i | 2 | 2 | 1 | ((Ri)) ← #d |
| Загрузка указателя данных | MOV DPTR, #d16 | 10010000 | 13 | 3 | 2 | (DPTR) ← #d16 |
| Пересылка в аккумулятор байта из ПП | MOVC A, @A+DPTR | 10010011 | 1 | 1 | 2 | ← ((A) +(DPTR)) |
| Пересылка в аккумулятор байта из ПП | MOVC A, @A+PC | 10000011 | 1 | 1 | 2 | (PC) ← (PC)+1,  (A) ← ((A)+(PC)) |
| Название команды | Мнемокод | КОП | Т | Б | Ц | Операция |
| Пересылка в аккумулятор байта из ВПД | MOVX A, @Ri | 1110001i | 1 | 1 | 2 | (A) ← ((Ri)) |
| Пересылка в аккумулятор байта из расширенной ВПД | MOVX A,@DPTR | 11100000 | 1 | 1 | 2 | (A) ← ((DPTR)) |
| Пересылка в ВПД из аккумулятора | MOVX @Ri, A | 1111001i | 1 | 1 | 2 | ((Ri)) ← (A) |
| Пересылка в расширенную ВПД из аккумулятора | MOVX @DPTR,A | 11110000 | 1 | 1 | 2 | ((DPTR)) ← (A) |
| Загрузка в стек | PUSH ad | 11000000 | 3 | 2 | 2 | (SP) ← (SP) + 1, ((SP)) ← (ad) |
| Извлечение из стека | POP ad | 11010000 | 3 | 2 | 2 | (ad) ← (SP),  (SP) ← (SP) - 1 |
| Обмен аккумулятора с регистром | XCH A, Rn | 11001rrr | 1 | 1 | 1 | (A) ↔ (Rn) |
| Обмен аккумулятора с прямоадресуемым байтом | XCH A, ad | 11000101 | 3 | 2 | 1 | (A) ↔ (ad) |
| Обмен аккумулятора с байтом из РПД | XCH A, @Ri | 1100011i | 1 | 1 | 1 | (A) ↔ ((Ri)) |
| Обмен младших тетрад аккумулятора и байта РПД | XCHD A, @Ri | 1101011i | 1 | 1 | 1 | (A0…3) ↔((Ri)0…3) |
| Сложение аккумулятора с регистром (n=0÷7) | ADD A, Rn | 00l01rrr | 1 | 1 | 1 | (A) ← (A) + (Rn) |
| Сложение аккумулятора с прямоадресуемым байтом | ADD A, ad | 00100101 | 3 | 2 | 1 | (A) ← (A) + (ad) |
| Сложение аккумулятора с байтом из РПД (i = 0,1) | ADD A, @Ri | 0010011i | 1 | 1 | 1 | (A) ← (A) + ((Ri)) |
| Сложение аккумулятора с константой | ADD A, #d | 00100100 | 2 | 2 | 1 | (A) ← (A) + #d |
| Сложение аккумулятора с регистром и переносом | ADDC A, Rn | 00111rrr | 1 | 1 | 1 | (A) ← (A) + (Rn) + (C) |
| Сложение аккумулятора с прямоадресуемым байтом и переносом | ADDC A, ad | 00110101 | 3 | 2 | 1 | (A) ← (A) + (ad) + (C) |
| Сложение аккумулятора с байтом из РПД и переносом | ADDC A, @Ri | 0011011i | 1 | 1 | 1 | (A) ← (A) + ((Ri)) + (C) |
| Сложение аккумулятора с константой и переносом | ADDC A, #d | 00110100 | 2 | 2 | 1 | (A) ← (A) + # d + (C) |
| Десятичная коррекция аккумулятора | DA A | 11010100 | 1 | 1 | 1 | Если (А0…3)>9 или ((AC)=1),то(А0…3)← (А0…З) + 6, затем если (А4…7) >9 или ((С)=1), то (А4…7) ← (А4…7) + 6 |
| Вычитание из аккумулятора регистра и заёма | SUBB A, Rn | 10011rrr | 1 | 1 | 1 | (A) ← (A) - (C) - (Rn) |
| Вычитание из аккумулятора прямоадресуемого байта и заема | SUBB A, ad | 10010101 | 3 | 2 | 1 | (A) ← (A) - (C) - ((ad)) |
| Вычитание из аккумулятора байта РПД и заема | SUBB А, @Ri | 1001011i | 1 | 1 | 1 | (A) ← (A) - (C) - ((Ri)) |
| Вычитание из аккумулятора константы и заема | SUBB А, d | 10010100 | 2 | 2 | 1 | (A) ← (A) - (C) - #d |
| Инкремент аккумулятора | INC А | 00000100 | 1 | 1 | 1 | (A) ← (A) + 1 |
| Инкремент регистра | INC Rn | 00001rrr | 1 | 1 | 1 | (Rn) ← (Rn) + 1 |
| Инкремент прямоадресуемого байта | INC ad | 00000101 | 3 | 2 | 1 | (ad) ← (ad) + 1 |
| Инкремент байта в РПД | INC @Ri | 0000011i | 1 | 1 | 1 | ((Ri)) ← ((Ri)) + 1 |
| Инкремент указателя данных | INC DPTR | 10100011 | 1 | 1 | 2 | (DPTR ) ← (DPTR) + 1 |
| Декремент аккумулятора | DEC A | 00010100 | 1 | 1 | 1 | (A) ← (A) – 1 |
| Декремент регистра | DEC Rn | 00011rrr | 1 | 1 | 1 | (Rn) ← (Rn) – 1 |
| Декремент прямоадресуемого байта | DEC ad | 00010101 | 3 | 2 | 1 | (ad) ← (ad) – 1 |
| Декремент байта в РПД | DEC @Ri | 0001011i | 1 | 1 | 1 | ((Ri)) ← ((Ri)) - 1 |
| Умножение аккумулятора на регистр В | MUL AB | 10100100 | 1 | 1 | 4 | (B)(A) ← (A)\*(В) |
| Деление аккумулятора на регистр В | DIV AB | 10000100 | 1 | 1 | 4 | (B).(A) ← (A)/(В) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Логическое И аккумулятора и регистра | ANL A, Rn | 01011rrr | 1 | 1 | 1 | (A) ← (A) AND (Rn) |
| Логическое И аккумулятора и прямоадресуемого байта | ANL A, ad | 01010101 | 3 | 2 | 1 | (A) ← (A) AND (ad) |
| Логическое И аккумулятора и байта из РПД | ANL A, @Ri | 0101011i | 1 | 1 | 1 | (A) ← (A) AND ((Ri)) |
| Логическое И аккумулятора и константы | ANL A, #d | 01010100 | 2 | 2 | 1 | (A) ← (A) AND #d |
| Логическое И прямоадресуемого байта и аккумулятора | ANL ad, A | 01010010 | 3 | 2 | 1 | (ad) ← (ad) AND (A) |
| Логическое И прямоадресуемого байта и константы | ANL ad, #d | 01010011 | 7 | 3 | 2 | (ad) ← (ad) AND #d |
| Логическое ИЛИ аккумулятора и регистра | ORL A, Rn | 01001rrr | 1 | 1 | 1 | (A) ← (A) OR (Rn) |
| Логическое ИЛИ аккумулятора и прямоадресуемого байта | ORL A, ad | 01000101 | 3 | 2 | 1 | (A) ← (A) OR (ad) |
| Логическое ИЛИ аккумулятора и байта из РПД | ORL A, @Ri | 0100011i | 1 | 1 | 1 | (A) ← (A) OR ((Ri)) |
| Логическое ИЛИ аккумулятора и константы | ORL A, #d | 01000100 | 2 | 2 | 1 | (A) ← (A) OR #d |
| Логическое ИЛИ прямоадресуемого байта и аккумулятора | ORL ad, A | 01000010 | 3 | 2 | 1 | (ad) ← (ad) OR (A) |
| Логическое ИЛИ прямоадресуемого байта и константы | ORL ad, #d | 01000011 | 7 | 3 | 2 | (ad) ← (ad) OR #d |
| Исключающее ИЛИ аккумулятора и регистра | XRL A, Rn | 01101rrr | 1 | 1 | 1 | (A) ← (A) XOR (Rn) |
| Исключающее ИЛИ аккумулятора и прямоадресуемого байта | XRL A, ad | 01100101 | 3 | 2 | 1 | (A) ← (A) XOR (ad) |
| Исключающее ИЛИ аккумулятора и байта из РПД | XRL A, @Ri | 0110011i | 1 | 1 | 1 | (A) ← (A) XOR ((Ri)) |
| Исключающее ИЛИ аккумулятора и константы | XRL A, #d | 01100100 | 2 | 2 | 1 | (A) ← (A) XOR #d |
| Исключающее ИЛИ прямоадресуемого байта и аккумулятора | XRL ad, A | 01100010 | 3 | 2 | 1 | (ad) ← (ad) XOR (A) |
| Исключающее ИЛИ прямоадресуемого байта и константы | XRL ad, #d | 01100011 | 7 | 3 | 2 | (ad) ← (ad) XOR #d |
| Сброс аккумулятора | CLR A | 11100100 | 1 | 1 | 1 | (A) ← 0 |
| Инверсия аккумулятора | CPL A | 11110100 | 1 | 1 | 1 | (A) ← NOT(A) |
| Сдвиг аккумулятора влево циклический | RL A | 00100011 | 1 | 1 | 1 | (An+1) ← (An), n=0÷6, (A0) ← (A7) |
| Сдвиг аккумулятора влево через перенос | RLC A | 00110011 | 1 | 1 | 1 | (An+1) ← (An), n=0÷6  (A0) ← (C), (C) ←(A7) |
| Сдвиг аккумулятора вправо циклический | RR A | 00000011 | 1 | 1 | 1 | (An) ← (An+1), n=0÷6, (A7) ← (A0) |
| Сдвиг аккумулятора вправо через перенос | RRC A | 00010011 | 1 | 1 | 1 | (An) ← (An+1), n=0÷6  (A7) ← (C), (C) ←(A0) |
| Обмен местами тетрад в аккумуляторе | SWAP A | 11000100 | 1 | 1 | 1 | (A0…3) ↔ (A4…7) |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Сброс переноса | CLR С | 11000011 | 1 | 1 | 1 | (C) ← 0 |
| Сброс бита | CLR bit | 11000010 | 4 | 2 | 1 | (b) ← 0 |
| Установка переноса | SETB С | 11010011 | 1 | 1 | 1 | (C) ← 1 |
| Установка бита | SETB bit | 11010010 | 4 | 2 | 1 | (b) ← 1 |
| Инверсия переноса | CPL С | 10110011 | 1 | 1 | 1 | (C) ← NOT(C) |
| Инверсия бита | CPL bit | 10110010 | 4 | 2 | 1 | (b) ← NOT(b) |
| Логическое И бита и переноса | ANL С, bit | 10000010 | 4 | 2 | 2 | (C) ← (C) AND (b) |
| Логическое И инверсии бита и переноса | ANL С, /bit | 10110000 | 4 | 2 | 2 | (C) ← (C) AND (NOT(b)) |
| Логическое ИЛИ бита и переноса | ORL С, bit | 01110010 | 4 | 2 | 2 | (C) ← (C) OR (b) |
| Логическое ИЛИ инверсии бита и переноса | ORL С, /bit | 10100000 | 4 | 2 | 2 | (C) ← (C) OR (NOT(b)) |
| Пересылка бита в перенос | MOV С, bit | 10100010 | 4 | 2 | 1 | (C) ← (b) |
| Пересылка переноса в бит | MOV bit, С | 10010010 | 4 | 2 | 2 | (b) ← (C) |
| Длинный переход в полном объеме ПП | LJMP ad16 | 00000010 | 12 | 3 | 2 | (PC) ← ad16 |
| Абсолютный переход внутри страницы в 2 Кб | AJMP ad11 | a10a9a800001 | 6 | 2 | 2 | (PC) ← (PC) + 2, (PC0-10) ← ad11 |
| Короткий относительный переход внутри страницы в 256 байт | SJMP rel | 10000000 | 5 | 2 | 2 | (PC) ← (PC) + 2, (PC) ← (PC) +rel |
| Косвенный относительный переход | JMP @A+DPTR | 01110011 | 1 | 1 | 2 | (PC) ← (A) + (DPTR) |
| Переход, если аккумулятор равен нулю | JZ rel | 01100000 | 5 | 2 | 2 | (PC)←(PC)+2, если (A)=0, то (PC)←(PC)+rel |
| Переход, если аккумулятор не равен нулю | JNZ rel | 01110000 | 5 | 2 | 2 | (PC)←(PC)+2, если (A)≠0, то (PC)←(PC)+rel |
| Переход, если перенос равен единице | JC rel | 01000000 | 5 | 2 | 2 | (PC)←(PC)+2, если (С)=1, то (PC)←(PC)+rel |
| Переход, если перенос равен нулю | JNC rel | 01010000 | 5 | 2 | 2 | (PC)←(PC)+2, если (С)=0, то (PC)←(PC)+rel |
| Переход, если бит равен единице | JB bit, rel | 00100000 | 11 | 3 | 2 | (PC)←(PC)+3, если (b)=l, то (PC)←(PC)+rel |
| Переход, если бит равен нулю | JNB bit, rel | 00110000 | 11 | 3 | 2 | (PC)←(PC)+3, если (b)=0, то (PC)←(PC)+rel |
| Переход, если бит установлен, с последующим сбросом бита | JBC bit, rel | 00010000 | 11 | 3 | 2 | (PC) ← (PC) + 3, если (b)=1, то (b) ← 0 и (PC)← (PC) + rel |
| Декремент регистра и переход, если не нуль | DJNZ Rn, rel | 11011rrr | 5 | 2 | 2 | (PC) ← (PC) + 2, (Rn) ← (Rn) - 1,  если (Rn) ≠ 0, то (PC) ← (PC) + rel |
| Декремент прямоадресуемого байта и переход, если не нуль | DJNZ ad, rel | 11010101 | 8 | 3 | 2 | (PC) ← (PC) + 2, (ad) ← (ad) - 1, если (ad) ≠ 0, то (PC) ← (PC) + rel |
| Сравнение аккумулятора с прямоадресуемым байтом и переход, если не равно | CJNE A, ad, rel | 10110101 | 8 | 3 | 2 | (PC) ← (PC) + 3,если (A) ≠ (ad), то (PC) ← (PC) + rel,если (A) < (ad), то (C) ← 1, иначе (C) ← 0 |
| Сравнение аккумулятора с константой и переход, если не равно | CJNE A, #d, rel | 10110100 | 10 | 3 | 2 | (PC) ← (PC) + 3,если (A) ≠ #d, то (PC) ← (PC) + rel,если (A) < #d, то (C) ← 1, иначе (С) ← 0 |
| Сравнение регистра с константой и переход, если не равно | CJNE Rn, #d, rel | 10111rrr | 10 | 3 | 2 | (PC) ← (PC) + 3,если (Rn) ≠ #d, то (PC) ← (PC) + rel,если (Rn) < #d, то (C) ← 1, иначе (С) ← 0 |
| Сравнение байта в РПД с константой и переход, если не равно | CJNE @Ri,#d,rel | 1011011i | 10 | 3 | 2 | (PC) ← (PC) + 3,если ((Ri)) ≠ #d, то (PC) ← (PC) + rel,если ((Ri)) < #d, то (C) ← 1, иначе (C) ← 0 |
| Длинный вызов подпрограммы | LCALL adl6 | 00010010 | 12 | 3 | 2 | (PC) ← (PC) + 3, (SP) ← (SP) +1,  ((SP)) ← (PC0…7), (SP) ← (SP) + 1,  ((SP)) ← (PC8…15), (PC) ← ad16 |
| Абсолютный вызов подпрограммы в пределах страницы в 2 Кб | ACALL ad11 | a10a9a810001 | 6 | 2 | 2 | (PC) ← (PC) + 2, (SP) ← (SP) + 1,  ((SP)) ← (PC0…7), (SP) ← (SP) + 1,  ((SP)) ← (PC8…15), (PC0-10) ← ad11 |
| Возврат из подпрограммы | RET | 00100010 | 1 | 1 | 2 | (PC8…15) ← ((SP)), (SP) ← (SP) - 1,  (PC0…7) ← ((SP)), (SP) ← (SP) – 1 |
| Возврат из подпрограммы обработки прерывания | RETI | 00110010 | 1 | 1 | 2 | (PC8…15) ← ((SP)), (SP) ← (SP) - 1,  (PC0…7) ← ((SP)), (SP) ← (SP) – 1 |
| Пустая операция | NOP | 00000000 | 1 | 1 | 1 | (PC) ← (PC) + 1 |

Приложение Б

Подпрограмма основного опроса датчиков, передача сигнала на последовательный порт (модем) адреса датчика при его ;срабатывании.

OPROS:

в P1.0..P1.3 записывается адресующий код мультиплексора – код

проверяемого датчика

ADDRESS\_BIT0 EQU P1.0 ;адрес бита порта P1.0

ADDRESS\_BIT1 EQU P1.1 ;адрес бита порта P1.1

ADDRESS\_BIT2 EQU P1.2 ;адрес бита порта P1.2

ADDRESS\_BIT3 EQU P1.3 ;адрес бита порта P1.3

TEMP EQU R1 ;адрес регистра R1

P0.0, P0.1 – биты выходного сигнала мультиплексоров №1 и №2 соответственно

BIT\_Q\_MUX1 EQU P0.0 ;адрес бита порта P0.0

BIT\_Q\_MUX2 EQU P0.1 ;адрес бита порта P0.1

CLR A ;сброс аккумулятора

CLR P1.4 ;разрешение работы мультиплексора №1

CLR P1.5 ;разрешение работы мультиплексора №2

CALL INIT ;вызов подпрограммы инициализации последовательного ;порта

START\_OPROS:

MOV C,ACC.0 ;запись адреса проверяемого ;датчика

MOV ADDRESS\_BIT0,С

в порты P1.0..P1.3 через бит ;переноса

MOV C,ACC.1

MOV ADDRESS\_BIT1,C

MOV C,ACC.2

MOV ADDRESS\_BIT2,C

MOV C,ACC.3

MOV ADDRESS\_BIT3,C

JNB BIT\_Q\_MUX1,NOT\_ERR

если не сработал датчик

подсоединенный к 1-му

мультиплексору, то проверить

датчик по такому же адресу 2-го

мультиплексора

CLR TEMP ;очищаем регистр, для временного хранения адреса датчика

MOV TEMP,A ;заносим адрес сработавшего датчика

ORL A,#00010000B

4-й бит аккумулятора обозначает 1-й мультиплексор

CALL SP\_OUT ;вызов подпрограммы передачи ;данных о сработавшем датчике через последовательный порт

MOV A,TEMP ;восстанавливаем значение адреса ;датчика

NOT\_ERR:

JNB BIT\_Q\_MUX2,END\_OPROS

если не сработал датчик ;подсоединенный ко 2-му ;мультиплексору, то продолжить сканирование

CLR TEMP ;очищаем регистр, для временного ;хранения адреса датчика

MOV TEMP,A ;заносим адрес сработавшего ;датчика

ORL A,#00100000B

5-й бит аккумулятора обозначает ;2-й мультиплексор

CALL SP\_OUT ;вызов подпрограммы передачи ;данных о сработавшем датчике через последовательный порт

MOV A,TEMP ;восстанавливаем значение адреса ;датчика

END\_OPROS:

CJNE A,#00001111B,NOT\_EXIT

если была опрошена последняя ;пара датчиков, то

RET ;возвращение из подпрограммы ;опроса датчиков

NOT\_EXIT:

INC A ;увеличение аккумулятора на 1, для опроса следующей пары датчиков

JMP START\_OPROS

продолжаем опрос

Подпрограмма инициализации последовательного порта в режиме 3 со скоростью передачи данных 2400 бод.

INIT: MOV SCON,#00100010B

MOV TMOD,#00100000B

;настройка Т/С1

MOV PCON,#0 ;сброс бита SMOD

MOV TH1,#<.NOT.13+1

K=f/(12\*32\*0.0024)= 13, где ;f=12МГц есть частота синхронизации микроконтроллера

SET TR1 ;включить Т/С1

RET ;вернуться из подпрограммы

Пподпрограмма передачи данных и контрольного бита данных через последовательных порт

SP\_OUT: MOV C,P ;пересылка контрольного бита

MOV TB8,C ;из флага паритета SCON.3

M1: JNT T1,M1 ;синхронизация передачи

CLR T1 ;сброс бита SCON.1

MOV SBUF,A ;передача контрольного бита и содержимого аккумулятора через последовательный порт

RET