

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Кафедра радиотехнических систем

## **МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА**

Методическое пособие  
к курсовому проектированию  
для студентов радиотехнических специальностей  
всех форм обучения

Минск БГУИР 2009

УДК 004.31(076)  
ББК 32.973.26-018.2я73  
М59

**Рецензент:**  
доцент кафедры сетей и устройств телекоммуникаций БГУИР,  
канд. техн. наук И. И. Астровский

**Авторы:**  
В. Н. Левкович, О. В. Шабров, А. А. Казека, А. В. Мартинович

**Микропроцессорные** устройства: метод. пособие к курс. проект.  
М59 для студ. радиотех. спец. всех форм обуч. / В. Н. Левкович [и др.]. –  
Минск : БГУИР, 2009. – 40 с. : ил.  
ISBN 978-985-488-438-7

В пособии сформулированы цели и задачи курсового проектирования, приведены типовые темы курсовых работ, требования к содержанию и оформлению пояснительной записки, а также пример выполнения основных разделов курсовой работы на тему «Цифровой тахометр».

УДК 004.31(076)  
ББК 32.973.26-018.2я73

ISBN 978-985-488-438-7

© УО «Белорусский государственный университет  
информатики и радиоэлектроники», 2009

## 1 ЦЕЛИ И ЗАДАЧИ КУРСОВОГО ПРОЕКТИРОВАНИЯ

В настоящее время микропроцессоры и микроконтроллеры применяются в радиоэлектронике как универсальная элементная база, позволяющая быстро, надежно и экономично реализовывать функции преобразования информации, ее хранения и передачи, цифровой обработки и формирования сигналов, управления техническими объектами, человеко-машинных интерфейсов и многое другое. Для современного инженера по радиоэлектронике знание микропроцессорной техники и умение решать на ее базе технические задачи является обязательным условием его успешной работы в избранном направлении.

Начальные знания и навыки по архитектуре, принципам организации и функционирования, проектированию микропроцессорных устройств и систем студенты получают в рамках лекционных, практических и лабораторных занятий. Но твердые знания и уверенные навыки приходят к специалистам только в результате самостоятельной деятельности. Курсовое проектирование является разновидностью самостоятельной управляемой работы студентов.

*Целью* курсового проектирования в рамках изучения дисциплины «Микропроцессорные устройства» является развитие и закрепление практических навыков проектирования специализированных вычислителей, устройств управления, передачи данных, формирования и обработки сигналов на базе микропроцессоров и микроконтроллеров.

*Задачей* курсового проекта является разработка функционально законченного микропроцессорного устройства. Подтверждение работоспособности проектируемого устройства должно осуществляться моделированием и экспериментальными исследованиями. Проектирование предлагается проводить на базе высокопроизводительного, простого в освоении, популярного среди разработчиков аппаратуры, обеспеченного бесплатной эффективной интегрированной средой разработки и отладки программ, дешевого и доступного на рынке семейства однокристальных микроконтроллеров PICMicro фирмы Microchip. Сочетание перечисленных качеств этих микроконтроллеров уже в рамках учебного процесса позволяет студентам пройти полный цикл – от постановки задачи по созданию устройства до работающего микропроцессорного устройства.

Как показывает практика, именно самостоятельное получение законченного результата придает студентам уверенность в собственных силах, пробуждает интерес к дальнейшему творчеству и дает реальные шансы на получение оплачиваемой работы по выбранной специальности уже во время учебы.

## 2 ТЕМАТИКА КУРСОВЫХ РАБОТ

Ниже приводятся типичные темы для курсового проектирования. В пределах каждой темы задаются возможные варианты основных параметров и функциональных возможностей проектируемых устройств. Комбинируя варианты, можно сформулировать достаточное количество тем для обеспечения всего потока студентов неповторяющимися темами.

Особенностью тематики является то, что студенту выдаются требования только на основополагающие параметры и функциональные возможности проектируемого микропроцессорного устройства. Все производные параметры и возможности будущего изделия студент должен выбрать самостоятельно, исходя из технических возможностей заданного микроконтроллера и собственных способностей. Сложность решенной задачи и глубина ее проработки и определяют итоговую оценку работы студента.

### 1. Цифровой вольтметр

Вид измеряемого напряжения: постоянное, переменное частотой 50 Гц.

Пределы измерения: 0...1 В, 0...5 В, 0...30 В, 0...100 В.

Метод аналого-цифрового преобразования: последовательного приближения, поразрядного приближения, следящего типа, двойного интегрирования.

### 2. Цифровой амперметр

Вид измеряемого тока: постоянный, переменный частотой 50 Гц.

Пределы измерения: 0...10 мА, 0...30 мА, 0...50 мА, 0...100 мА, 0...1 А.

Метод АЦП: последовательного приближения, поразрядного приближения, следящего типа, двойного интегрирования.

### 3. Цифровой генератор сигналов

Форма выходного сигнала: прямоугольные импульсы, синусоидальный, треугольные импульсы, пилообразные импульсы.

Регулируемые параметры: частота, амплитуда, длительность (для импульсов).

Режимы работы: непрерывный, одиночный по запуску.

### 4. Электронные часы

Варианты отображения: часы, минуты; часы, минуты, секунды.

Режимы работы: 12-часовой цикл, 24-часовой цикл.

Наличие будильника: да, нет.

#### 5. Электронный таймер

Пределы выдержек: 0...10 с, 1...60 с, 1...60 мин, 1...12 ч, 1...24 ч.

#### 6. Электронный частотомер

Пределы измерений: 0...10 Гц, 0...100 Гц, 0...1 кГц, 0...10 кГц,

#### 7. Цифровой кодовый замок

Варианты кодовых последовательностей: двоичная, троичная, десятичная, шестнадцатеричная.

Длина кодовой последовательности (в символах): 3, 4, 5, 6, 8, 10.

Защита от взлома путем перебора комбинаций: есть, нет.

#### 8. Цифровой термометр

Пределы измерения: 0...50 °С, минус 50...плюс 50 °С, 0...120 °С.

Первичный датчик температуры: терморезистор, интегральный с цифровым кодовым выходом, интегральный с импульсным выходом.

#### 9. Калькулятор

Выполняемая операция: сложение, умножение, вычитание, деление.

Обрабатываемые данные:

целые десятичные числа без знака от одного до трех разрядов,

правильные десятичные дроби от одного до четырех разрядов.

#### 10. Спидометр

Варианты назначения: для пешехода, для велосипеда.

Измеряемые и отображаемые параметры: скорость движения, пройденный путь, скорость движения и пройденный путь.

#### 11. Пульт дистанционного управления по инфракрасному лучу

Количество команд: 2, 4, 8.

Варианты битового кодирования: логические уровни, широтно-импульсная манипуляция, фазоимпульсная манипуляция, амплитудно-импульсная манипуляция.

#### 12. Приемник команд дистанционного управления по инфракрасному лучу

Количество команд: 2, 4, 8.

Варианты битового кодирования: логические уровни, широтно-импульсная манипуляция, фазоимпульсная манипуляция, амплитудно-импульсная манипуляция.

13. Фазовый регулятор мощности электронагревателя

Количество ступеней регулирования: 2, 4, 8, 10, 16.

14. Источник питания с регулируемым напряжением

Пределы регулирования выходного напряжения: 0...6 В, 4...15 В, 10...30 В.

15. Электронные весы

Первичный датчик: тензорезистор, преобразователь давление/частота.

Варианты аналого-цифрового преобразователя (АЦП) (для тензорезистора): двойного интегрирования, следящего типа с цифроаналоговым преобразователем (ЦАП) на основе широтно-импульсной модуляции (ШИМ).

16. Контроллер электропривода миксера

Функциональные возможности:

- «мягкий старт»,
- управление оборотами,
- «мягкий старт» и управление оборотами.

17. Устройство передачи информации по последовательному каналу

Алфавит: десятичные цифры с символом разделения слов, шестнадцатеричные цифры с символом разделения слов, символы алфавита русского языка.

Протокол передачи: байт-асинхронный бит-синхронный, MicroLan.

18. Устройство приема информации по последовательному каналу

Алфавит: десятичные цифры с символом разделения слов, шестнадцатеричные цифры с символом разделения слов, символы алфавита русского языка.

Протокол передачи: байт-асинхронный бит-синхронный, MicroLan.

19. Контроллер УКВ радиоприемника

Реализовать функции поиска радиостанции, запоминания кода настройки и включение радиостанции по сохраненному коду настройки.

Варианты количества запоминаемых настроек: 1, 2, 4, 8, 16.

## 20. Устройство охранной сигнализации

Варианты назначения: охрана автомобиля, охрана закрытого помещения, охрана открытой зоны.

Варианты формирования сигналов оповещения: локальный акустический, передача сообщения по радиоканалу.

Варианты постановки на охрану и снятия с охраны: по коду с клавиатуры, по коду с электронного ключа.

Библиотека БГУИР

### 3 ЭТАПЫ ПРОЕКТИРОВАНИЯ

Для успешного решения задачи на микрокомпьютере (микроконтроллере) разработчик должен пройти следующие семь этапов:

- 1) постановка задачи;
- 2) выбор приемлемого алгоритма;
- 3) определение типов входных и выходных данных;
- 4) распределение аппаратных ресурсов микроконтроллера, т. е. портов ввода/вывода и периферийных устройств на кристалле;
- 5) проектирование и анализ решения, в том числе составление схем, описаний и пр.;
- 6) кодирование алгоритма на языке программирования;
- 7) проверка и отладка программы.

Процесс решения задачи носит, как правило, итерационный характер. Это означает, что, получив решение, разработчик часто бывает вынужден вернуться вновь к третьему, второму и даже первому этапу.

По-видимому, первые рассмотренные выше четыре этапа относятся к тому, что принято называть «искусством разработки». Здесь успех в основном определяется опытом разработчика, его знанием объекта разработки, поэтому дать конкретные рекомендации не представляется возможным.

Однако можно подробнее остановиться на алгоритмах и методах их описаний.

Современные компьютеры способны выполнять широкий круг задач по получению, передаче, хранению, переработке информации, принятию решений и т. д. Но все эти действия должны быть заранее подготовлены, запрограммированы человеком.

Создание программы основывается на алгоритме решения задачи, в соответствии с ним создается последовательность команд, которая и составляет программу.

Под алгоритмом понимается конечный набор действий для выполнения некоторой процедуры, удовлетворяющий трем основным требованиям: массовости, детерминированности и результативности.

Требование массовости предполагает, что предписание должно обеспечивать выполнение не одной конкретной процедуры, а быть пригодным для реализации класса однородных процедур. Бессмысленно писать программу для получения суммы двух констант, но имеет смысл программа, определяющая сумму двух переменных, которые могут принимать множество значений в некотором диапазоне.

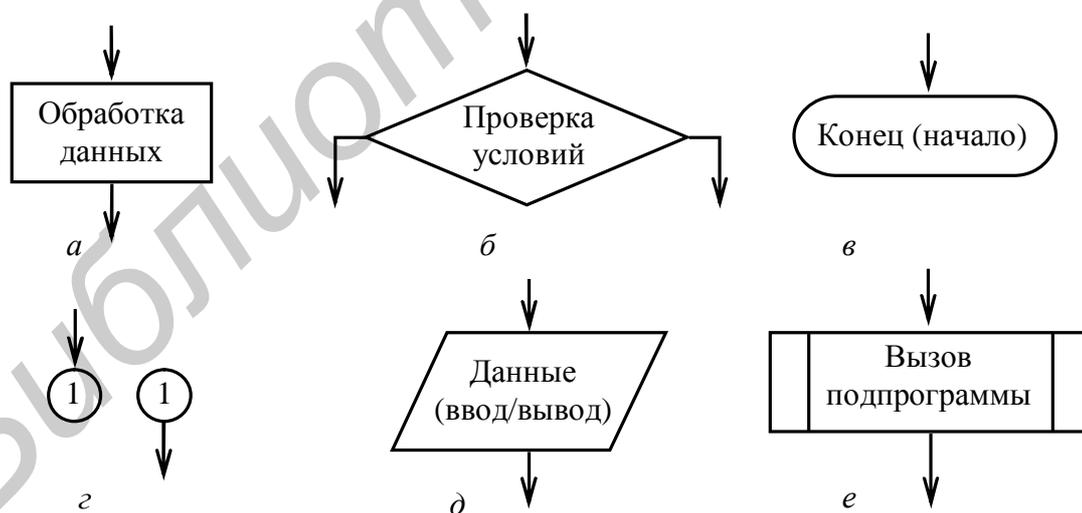
Детерминированность означает, что действия, образующие алгоритм, должны быть однозначно понимаемы, т. е. при одинаковых исходных данных независимо от исполнителя должна обеспечиваться одинаковость результатов.

Результативность обеспечивает конечность применения указаний. Результат должен быть получен за конечное число шагов либо за конечное число шагов должно быть получено указание о неприменимости данного алгоритма для решения поставленной задачи.

Алгоритм может быть описан в виде набора предложений, отражающих последовательность действий исполнителя, или в виде схемы, состоящей из ряда блоков, обозначающих действия исполнителя. Последний вариант хотя и имеет большую трудоемкость, но отличается большей наглядностью.

Изображение алгоритма решения задачи в виде схемы – важный этап подготовки задачи к решению на вычислительной машине. Схема позволяет разработчику адекватно представить работу программы. Кроме этого, схема алгоритма является одной из важных частей документации на разрабатываемую систему или устройство.

Схема алгоритма составляется из отдельных операторов. Различают шесть типов операторов, каждый из которых имеет один или несколько входов или один или несколько выходов (рисунок 3.1). Стрелками обозначают направление хода действий.



*a* – оператор обработки данных; *б* – оператор проверки условия;  
*в* – оператор начала или конца алгоритма; *г* – оператор соединения;  
*д* – оператор ввода или вывода данных; *е* – оператор вызова подпрограммы

Рисунок 3.1 – Типы операторов

Оператор в форме прямоугольника (рисунок 3.1, а) символизирует выполнение каких-либо операций по обработке данных; текст внутри прямоугольника является кратким описанием этого процесса обработки. Например, если в схеме алгоритма содержится оператор «Очистка аккумулятора», то это означает, что на данном этапе работы машины аккумулятор должен быть обнулен. В таком случае для выполнения этой операции микроконтроллеру требуется одна машинная команда. Однако могут быть заданы сложные действия, для которых требуется целый набор команд.

Оператор, имеющий форму ромба (рисунок 3.1, б), используется для символического обозначения проверки выполнения какого-либо условия с целью принятия решения о направлении последующего хода вычислений. Внутри ромба описывается условие, подлежащее проверке в той точке схемы алгоритма, где размещается данный оператор. Возможные результаты проверки указываются на линиях, выходящих из ромба. Для проведения подобной проверки требуется использование одной или нескольких команд микроконтроллера.

Оператор овальной формы используется для символического обозначения начала или конца алгоритма (рисунок 3.1, в). Текст внутри овала, как правило, состоит из одного слова – «Начало», «Конец», «Выход», «Возврат» или имени подпрограммы.

В тех случаях, когда необходимо «разорвать» линию потока вычислений, идущую от одного оператора к другому, применяются так называемые соединители в виде окружности с указанной внутри нее цифрой или буквой (рисунок 3.1, г). Наличие другого идентичного соединителя (с той же цифрой или буквой) означает, что прерванная в месте расположения первого соединителя линия продолжается с того места, где находится второй подобный соединитель. Использование соединителей упрощает внешний вид схемы алгоритма, что позволяет избежать пересечения линий и даёт возможность размещать схему алгоритма на нескольких страницах.

Для обозначения процедур ввода или вывода применяется оператор, имеющий форму параллелограмма (рисунок 3.1, д). Внутри параллелограмма указывают обычно переменные, подлежащие вводу или выводу.

Оператор, изображенный на рисунке 3.1, е, используется для обозначения вызова подпрограммы. Внутри него обычно помещается имя вызываемой подпрограммы.

Алгоритмы в зависимости от порядка следования операций бывают трех типов: линейные, ветвящиеся и циклические.

Схемы линейных алгоритмов не содержат операторов проверки условий, в них операторы располагаются строго друг за другом.

Ветвящиеся алгоритмы состоят из двух и более параллельных линейных ветвей, при этом ветвления реализуются через операторы проверки условий.

Циклический алгоритм предписывает многократные циклические проходы групп операторов.

Реальные схемы алгоритмов, как правило, представляют собой комбинации из трех перечисленных выше типов схем алгоритмов.

Библиотека БГУИР

## 4 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Пояснительная записка к курсовой работе должна содержать:

- задание на курсовое проектирование,
- титульный лист,
- введение,
- технические требования к разрабатываемому устройству,
- функциональную и структурную схемы устройства,
- принципиальную схему устройства,
- алгоритм работы управляющей программы,
- текст или листинг управляющей программы,
- результаты моделирования и испытания устройства,
- заключение,
- содержание,
- список литературы,
- приложения (при необходимости).

Общий объем пояснительной записки должен быть, как правило, в пределах от 15 до 30 страниц, оформленных в соответствии с требованиями, изложенными в разделе 5.

*Задание* на курсовое проектирование выдается преподавателем на бланке в течение 2-х недель с начала учебного семестра. Поощряются предложения тем для курсового проектирования со стороны самих студентов. Поскольку в этом случае присутствует личный интерес студента в успешном завершении работы, то сам процесс проектирования становится увлекательным, захватывающим и плодотворным.

*Титульный лист* должен содержать:

- названия министерства, учебного заведения, факультета, кафедры;
- заготовку для отметки о допуске работы к защите;
- название документа «Пояснительная записка...» и тему работы;
- фамилию, имя, отчество и номер учебной группы автора работы;
- заготовку для оценки и даты защиты работы;
- название города и календарный год выполнения работы.

Во *введении* необходимо отразить положительный эффект, который достигается при решении технических задач с помощью микропроцессоров и микроконтроллеров, а также актуальность темы курсовой работы и цель проектирования.

В задании на курсовое проектирование приводятся обычно только несколько основных технических параметров к разрабатываемому устройству. В разделе «*Технические требования...*» необходимо развить и дополнить заданные технические параметры и характеристики, функциональные возможности таким образом, чтобы получился целостный образ будущего устройства.

Содержание следующих пяти разделов записки понятно из их названий. Но надо помнить, что недостаточно привести схемы, они должны быть должным образом описаны. Для большей наглядности описаний полезно использовать временные диаграммы и математические выражения. Элементы принципиальной схемы должны быть применены в соответствии с рекомендациями производителей, выбраны или рассчитаны исходя из условий работы.

В *заключении* обычно приводится краткий перечень решенных задач, выполненных работ, отмечаются области применения, пути и направления совершенствования полученных результатов.

## **5 ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

### **5.1 Общие положения**

Пояснительную записку (ПЗ) выполняют рукописным способом или с применением печатающих и графических устройств вывода ЭВМ с соблюдением требований ГОСТ 2.105-95.

При рукописном способе используют шариковую ручку с пастой синего или фиолетового цвета. Высота букв и цифр должна быть не менее 3,5 мм. Текст ПЗ печатается с количеством знаков в строке 60–75. При применении текстовых редакторов ЭВМ печать производится шрифтом 13–14 пунктов с межстрочным интервалом, позволяющим разместить  $40 \pm 3$  строки на странице.

Текст располагают на одной стороне листа формата А4 с соблюдением размеров полей и интервалов, указанных на рисунке 5.1. Абзацы в тексте начинают отступом, равным 15–17 мм при выполнении записи рукописным способом или пяти знакам при применении печатающего устройства вывода ЭВМ.

Все части ПЗ необходимо излагать только на одном языке – русском или белорусском, или на одном из иностранных языков, например английском или немецком. Сокращение русских и белорусских слов и словосочетаний в записке выполняется по СТБ 7.12-94.

Описки и графические неточности, обнаруженные в тексте ПЗ, выполненной рукописным способом, допускается исправлять закрашиванием белой краской и нанесением на том же месте исправленного текста. Помарки и следы не полностью удаленного прежнего текста не допускаются.

### **5.2 Рубрикация, заголовки и содержание**

Текст ПЗ разделяют на логически сопряженные части – разделы и подразделы. Как разделы, так и подразделы могут состоять из одного или нескольких пунктов.

Разделы должны иметь порядковые номера в пределах всей ПЗ, обозначаемые арабскими цифрами без точки и записанные с абзацного отступа. Подразделы нумеруют в пределах раздела, которому они подчинены. Причем цифровой индекс подраздела должен состоять из порядкового номера раздела и отделенного от него точкой порядкового номера подраздела. Номер подраздела также записывают с абзацного отступа и точку в конце номера не ставят.

Иногда внутри подраздела необходимо выделить более мелкие смысловые подразделения – пункты, например: характеристики устройств и функциональных элементов технической системы; показатели качества технической системы

в различных режимах ее работы и т. п. В подобных случаях пункты нумеруют в пределах подраздела.

*Лист формата А4 (210×297 мм)*

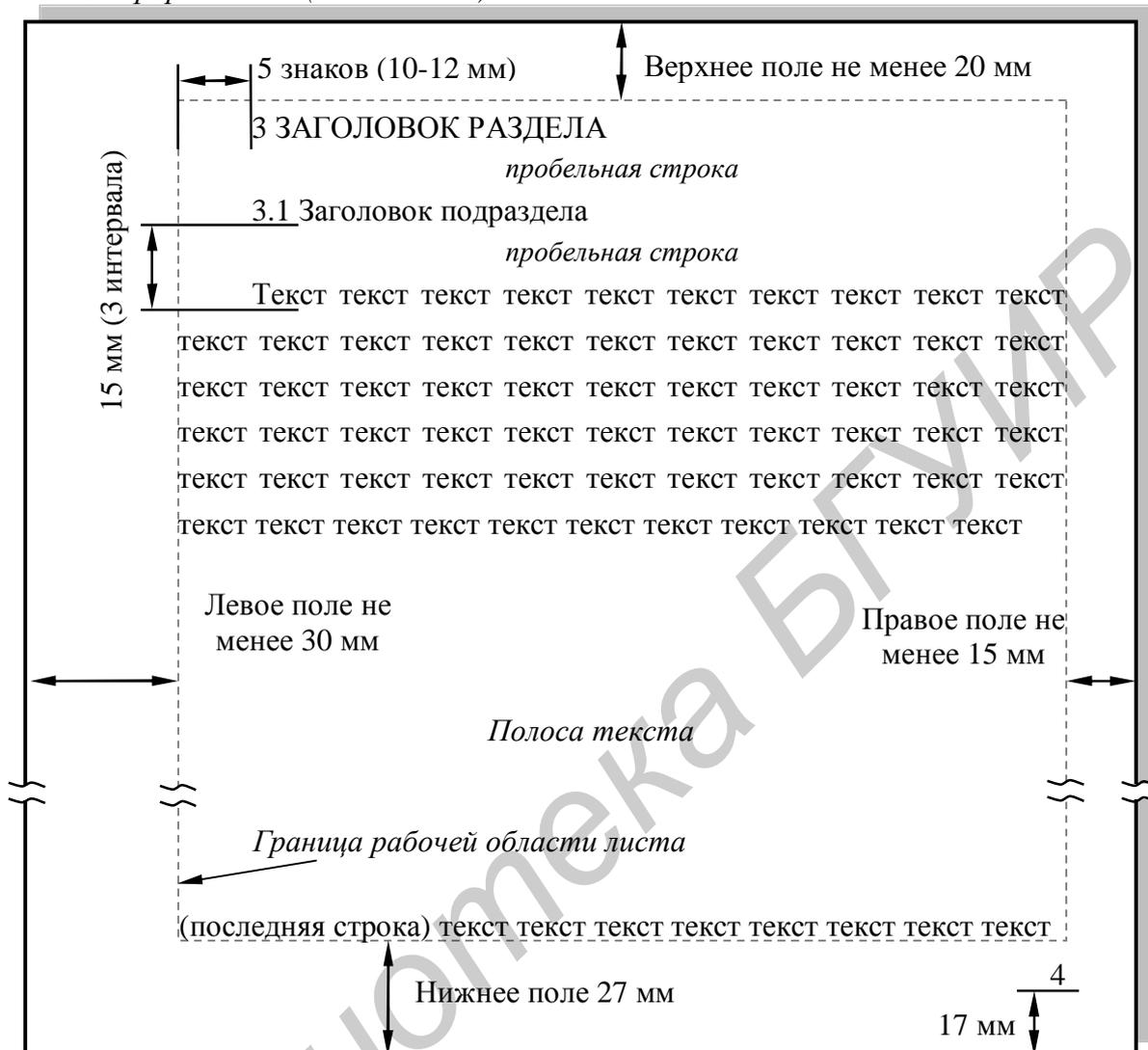


Рисунок 5.1 – Расположение текста на листе

Цифровой индекс пункта должен состоять из номеров раздела, подраздела и пункта, разделенных точками, и записан с абзацного отступа.

Если в ПЗ выделены только разделы, то пункты нумеруют в пределах раздела. Цифровой индекс пункта должен состоять из номера раздела и порядкового номера пункта, разделенных точкой. Номер пункта записывают с абзацного отступа.

Каждый раздел и подраздел должен иметь краткий и ясный заголовок. Пункты, как правило, заголовков не имеют.

Заголовки разделов записывают прописными буквами высотой не менее 5 мм (14 пт) без точки в конце заголовка. Заголовки подразделов записывают строчными буквами, начиная с первой прописной буквы. Заголовки не

подчеркивают. Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой.

Каждый раздел ПЗ рекомендуется начинать с новой страницы. Расстояние между заголовком раздела и заголовком подраздела должно составлять 8 мм при рукописном или полтора интервала печатающего устройства при машинном способе выполнения ПЗ (см. рисунок 5.1).

Между заголовком подраздела и текстом оставляют пробел шириной 15 мм при рукописном способе или 3–4 интервала – при машинном способе выполнения записки, что соответствует пропуску одной строки, называемой пробельной строкой. Если в ПЗ отсутствуют подразделы, то между заголовком раздела и текстом также оставляют промежуток, равный одной пробельной строке.

Перечень всех разделов и подразделов, включающий их порядковые номера и заголовки, оформляют в виде содержания – обязательного элемента ПЗ. Содержание помещают непосредственно за техническим заданием на проектирование и включают в общую нумерацию страниц.

Пример построения содержания показан на рисунке 5.2.

*Лист формата А4 (210×297 мм)*

Содержание	
Введение .....	2
1. Выбор и обоснование технических требований к устройству.....	3
2. Разработка структурной схемы.....	4
3. Разработка электрической принципиальной схемы.....	5
4. Разработка алгоритма работы программы .....	9
5. Листинг программы .....	15
Заключение .....	21
Литература .....	22

3

Рисунок 5.2 – Пример построения содержания

Страницы ПЗ нумеруют арабскими цифрами в правом нижнем углу. Титульный лист и лист технического задания включают в общую нумерацию, но номер на них не ставят. В общую нумерацию страниц включают все приложения.

### 5.3 Основные правила изложения текста

Текст ПЗ должен быть кратким, четким и не допускать различных толкований.

Текст излагают с соблюдением правил орфографии и пунктуации языка, на котором составляется ПЗ. Чаще всего эти правила нарушают при делении текста на абзацы, при перечислениях, при употреблении чисел, символов и размерностей.

Абзацами выделяют примерно равные и обособленные по смыслу части текста.

Перечисления в ПЗ обычно приводят в тех случаях, когда необходимо назвать состав технической системы или отдельной ее части, указать предъявляемые требования, установить порядок экспериментальных исследований, испытаний, наладки и т. п.

В тексте ПЗ, за исключением формул, таблиц и рисунков, не допускается:

– применять математический знак минус (–) перед отрицательными значениями величин (следует писать слово «минус»);

– применять знак « $\varnothing$ » (следует писать слово «диаметр»). В виде исключения на чертежах, помещенных в тексте документа, перед размерным числом следует писать знак « $\varnothing$ »;

– применять без числовых значений математические знаки, например:  $>$  (больше),  $<$  (меньше),  $=$  (равно),  $\geq$  (больше или равно),  $\leq$  (меньше или равно),  $\neq$  (не равно), а также знаки № (номер), % (процент).

Отвлеченные числа до девяти (числа без обозначения физических величин, числа счета) пишут только словами, свыше девяти – цифрами. Всегда пишут цифрами дроби и все цифры с размерностями. Дробные числа необходимо приводить в виде десятичных дробей.

Перед числами с размерностями не рекомендуется ставить предлог «в» или знак тире «–», например «конденсатор емкостью в 100 мкФ» следует писать «конденсатор емкостью 100 мкФ».

Если в ПЗ приводят диапазон числовых значений физической величины, выраженных в одной и той же единице физической величины, то обозначение единицы физической величины следует указывать после последнего числового значения диапазона.

В ПЗ следует применять единицы физических величин, их наименования и обозначения в соответствии с ГОСТ 8.417-81.

Дробные числа необходимо приводить в виде десятичных дробей, за исключением размеров в дюймах, которые следует записывать, например 1/2".

## **5.4 Основные требования к оформлению иллюстраций**

Виды иллюстраций (чертежи, схемы, графики, осциллограммы, фотографии и др.), их количество в ПЗ определяет автор работы. Допускается располагать иллюстрацию в конце ПЗ в виде приложения. Все иллюстрации независимо от их вида и содержания в технической литературе принято называть рисунками.

Каждую иллюстрацию сопровождают подрисуночной подписью, которую располагают симметрично полю, занимаемому иллюстрацией. Подпись должна содержать слово «Рисунок» без сокращения и порядковый номер иллюстрации арабскими цифрами, например «Рисунок 3» при сквозной нумерации или «Рисунок 3.3» при индексной нумерации иллюстраций по разделам ПЗ.

Подпись иллюстраций, расположенных в приложениях, должна содержать слово «Рисунок», обозначение приложения и порядковый номер иллюстрацией в приложении, например «Рисунок А.2». Если в приложении помещена одна иллюстрация, ее обозначают «Рисунок А.1».

При необходимости иллюстрациям можно давать наименования, которые записывают после номера рисунка через «тире» с прописной буквы. Точки после номера рисунка и после наименования не ставят, например «Рисунок 3.1 – Структурная схема разрабатываемого устройства».

Допускается выносить в подрисуночную подпись расшифровку условных обозначений, нумерованных частей и деталей иллюстрации. Все пояснительные данные помещают между рисунком и его обозначением.

Расшифровки пишут в подбор, отделяя их друг от друга точкой с запятой. Цифры, буквы, другие условные обозначения позиций на рисунке пишут без скобок, отделяя их от расшифровок знаками тире, например «1 – транзисторы; 2 – навесные конденсаторы; 3 – пленочные резисторы; 4 – направленные ответвители; 5 – ...». Стандартные буквенные позиционные обозначения, приведенные на рисунке, не расшифровывают. Если иллюстрации разъясняются в тексте ПЗ, то расшифровки в подрисуночных подписях не допускаются.

В тексте ПЗ должны быть даны ссылки на все иллюстрации без исключения.

## **5.5 Основные требования к оформлению таблиц**

Таблицы применяют с целью упростить изложение текста, содержащего достаточно большой по объему фактический материал, придать ему

компактную форму для анализа и расчетов, повысить обоснованность и достоверность принимаемых решений.

В виде таблицы обычно оформляют:

- сведения справочного характера;
- значения функций, используемые при графических методах расчета;
- данные экспериментальных исследований функциональных элементов и устройств;
- результаты математического моделирования и др.

Каждую таблицу в зависимости от ее размера рекомендуется помещать непосредственно за абзацем, в котором впервые дана на нее ссылка, либо на следующей странице. При необходимости допускается оформлять таблицу в виде приложения к ПЗ.

Все таблицы в тексте должны быть снабжены нумерацией арабскими цифрами и текстовыми заголовками. Номер таблицы и заголовок пишут в подбор, разделяя их знаком тире, и располагают над таблицей. Слово «Таблица» начинают писать с левой границы поля, занимаемого таблицей. После слова «таблица», порядкового номера и заголовка точки не ставят. Заголовок пишут с прописной буквы. Таблицы рекомендуется нумеровать в соответствии с принятой системой нумерации формул и рисунков, например: «Таблица 2» при сквозной нумерации или «Таблица 1.2» при индексной нумерации по разделам ПЗ.

Таблицы в каждом приложении снабжают отдельной нумерацией с обязательным указанием обозначения приложения на первой позиции номерного индекса, например «Таблица Б.2».

Заголовок должен быть кратким, точно отражать содержание таблицы и давать возможность читать таблицу, не обращаясь к тексту ПЗ. Строки с заголовком не должны выходить за пределы границ поля, занимаемого таблицей.

Таблицы фактического материала, как правило, оформляют в соответствии с рисунком 5.3. Высота строк таблицы должна быть не менее 8 мм.

Текст ПЗ и заголовок таблицы должны быть разделены пробельной строкой. Пробельной строкой отделяют от таблицы следующий за ней текст. Заголовок и фактический материал пробельной строкой не разделяют.

Таблица 1 – Функции сигнальных линий RS-232

Номер контакта	Обозначение	Назначение
2	TD (TXD)	Передаваемые данные
3	RD (RXD)	Принимаемые данные
7	SG	Сигнальная земля

Графы

Рисунок 5.3 – Пример типичного оформления таблицы

Если в конце страницы ПЗ таблица прерывается, то ее продолжение размещают на следующей странице. Продолжение оформляют, начиная с повторения головки, над левым углом которой пишут слова «Продолжение таблицы...» и указывают только ее номер.

### 5.6 Оформление библиографического указателя «СПИСОК ЛИТЕРАТУРЫ»

Ссылки на литературу, нормативно-техническую и другую документацию, иные источники, использованные при составлении текста ПЗ, помещают в конце ПЗ перед листом приложения в виде указателя СПИСОК ЛИТЕРАТУРЫ, записываемого прописными буквами с новой строки симметрично тексту.

В тексте ПЗ все ссылки на анализируемые опубликованные сведения, заимствованные положения, формулы, таблицы, иллюстрации, методики и т. п. нумеруют арабскими цифрами в квадратных скобках в возрастающем порядке.

В указателе «Литература» библиографические ссылки располагают и нумеруют в той последовательности, в какой расположены и пронумерованы ссылки в тексте ПЗ.

Без ссылок в тексте ПЗ разрешается использовать сведения, полученные на лекциях, семинарских, практических и лабораторных занятиях. Однако использованные учебные, учебно-методические материалы и пособия должны быть приведены и расположены в конце указателя.

Библиографические описания в указателе «Список литературы» должны быть выполнены в соответствии с правилами, установленными ГОСТ 7.1-2003.

## 6 ПРИМЕРЫ ВЫПОЛНЕНИЯ ОСНОВНЫХ РАЗДЕЛОВ КУРСОВОЙ РАБОТЫ НА ТЕМУ «ЦИФРОВОЙ ТАХОМЕТР»

### 6.1 Структурная схема

Структурная схема разрабатываемого цифрового тахометра показана на рисунке 6.1. Она включает формирователь импульсов (ФИ), микроконтроллер (МК) и цифровое табло (ЦТ).

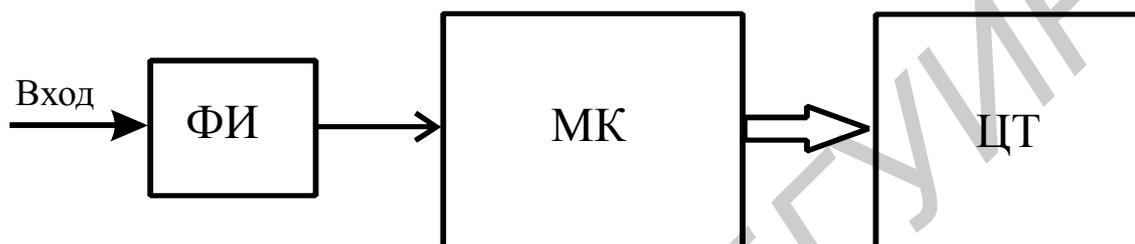


Рисунок 6.1 – Структурная схема цифрового тахометра

Формирователь импульсов предназначен для преобразования импульсов, поступающих на вход схемы от датчика вращения к логическим уровням, с которыми работает микроконтроллер. Дополнительно ФИ защищает входной порт микроконтроллера от перегрузок, вызванных электростатическими разрядами или внешними наводками.

Микроконтроллер программно реализует алгоритм работы устройства в целом. Он периодически производит подсчет количества импульсов, поступивших от датчика вращения вала в течение интервала измерения, преобразует это число в единицы измерения частоты вращения – обороты в минуту – и формирует сигналы управления цифровым табло.

Цифровое табло предназначено для оптического отображения измеренного значения частоты вращения вала двигателя.

### 6.2 Схема принципиальная электрическая

Принципиальная электрическая схема цифрового тахометра, разработанная в соответствии со структурной схемой, приведена на рисунке 6.2.

В качестве микроконтроллера выбран PIC16F628, отличающийся высоким быстродействием, низкой стоимостью, высокой нагрузочной способностью

выходных портов. Внешние элементы, подключаемые к микроконтроллеру, выбраны в соответствии с рекомендациями производителя.

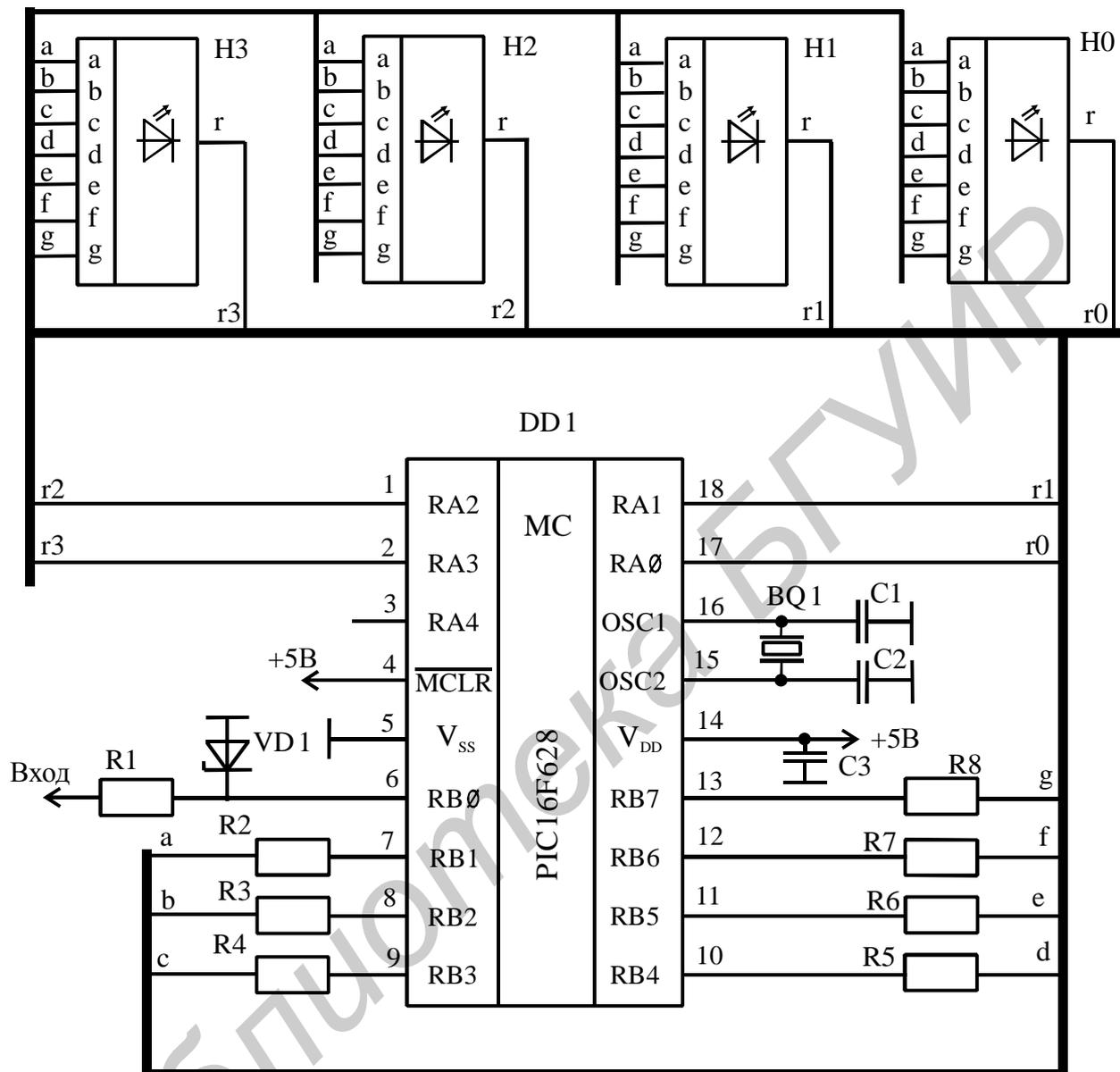


Рисунок 6.2 – Принципиальная электрическая схема цифрового тахометра

Кварцевый резонатор ВQ1 имеет резонансную частоту 4 МГц. Конденсаторы С1 и С2 керамические емкостью 20 пФ служат для предотвращения возбуждения тактового генератора на гармониках кварцевого резонатора.

Конденсатор С3 керамический емкостью 0,22 мкФ предназначен для подавления импульсных помех, распространяющихся по цепи питания.

В качестве формирователя импульсов использована цепочка из резистора R1 сопротивлением 10 кОм и стабилитрона VD1 с напряжением пробоя 5,1 В.

Она ограничивает уровень входных сигналов в пределах от минус 0,5 до плюс 5,5 В, что является безопасным для интегральных микросхем.

Резисторы R2...R8 номиналом 1 кОм служат для ограничения токов на уровне 3 мА, протекающих через сегменты светодиодного табло. Суммарный ток от сегментов одного разряда, замыкающийся через выводы порта А микроконтроллера, не будет превышать 21 мА при максимальном допустимом токе вывода микроконтроллера 24 мА. Поэтому в схеме обошлись без дополнительных разрядных усилителей.

Цифровое табло подключено по схеме динамического управления. Это позволило упростить аппаратную часть прибора за счет усложнения управляющей программы.

### 6.3 Схемы алгоритмов управляющей программы

Алгоритм управляющей программы построен по иерархическому принципу. Все действия сгруппированы в функционально законченные процедуры, которые оформлены в виде подпрограмм. В основной программе производится только настройка режимов работы функциональных блоков и узлов микроконтроллера, после чего она переходит в режим ожидания наступления значимых событий, которые и инициируют затем вызов цепочки подпрограмм. Схема алгоритма основной программы показана на рисунке 6.3.

Список подпрограмм и описание их назначения приводится ниже.

**Int.** Подпрограмма обработки прерываний. Определяет причину прерывания и вызывает соответствующую подпрограмму – Taim (если прерывание от TMR0) или Sum (если прерывание от входа INT). Схема алгоритма подпрограммы Int приведена на рисунке 6.4.

**Taim.** Производит коррекцию начального значения TMR0 для обеспечения требуемой задержки в следующем цикле. Затем вызывает программу Indic, производящую смену подсвечиваемого (текущего) разряда табло. Увеличивает содержимое программного таймера TIM на 1 и контролирует истечение интервала измерения 1 с. По истечении интервала 1 с вызывает подпрограмму Result. Схема алгоритма подпрограммы Taim показана на рисунке 6.5.

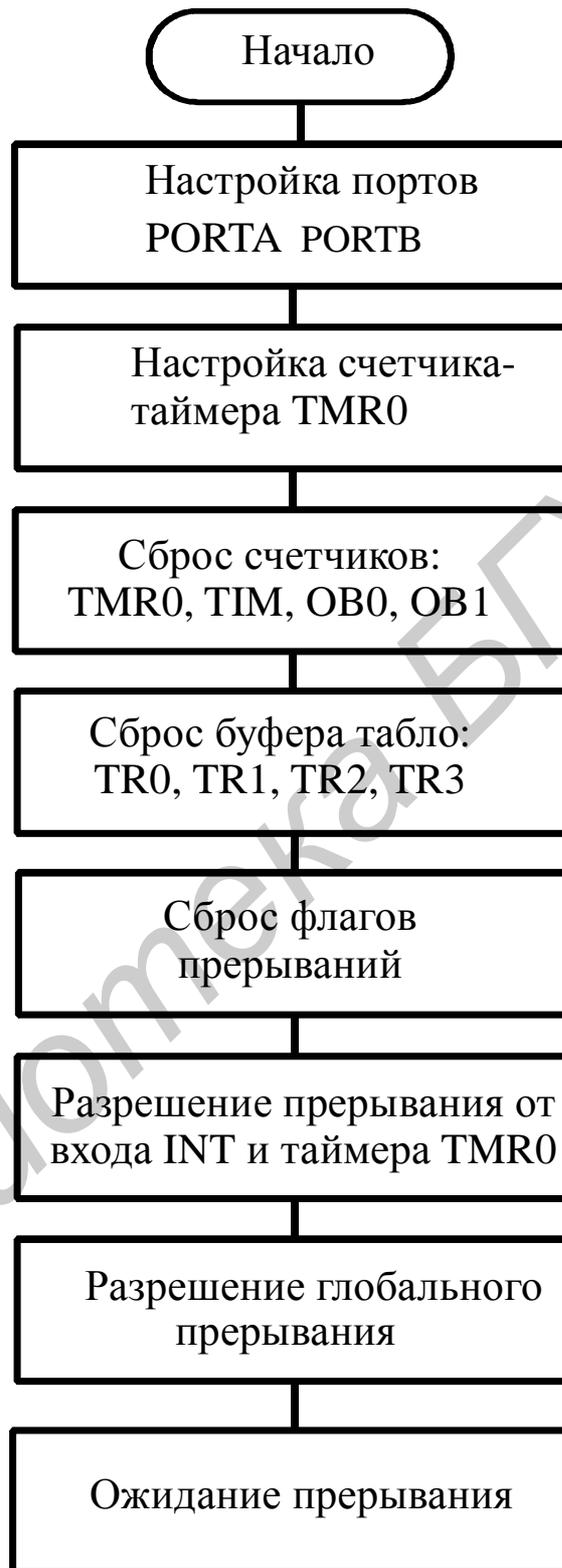


Рисунок 6.3 – Схема алгоритма основной программы

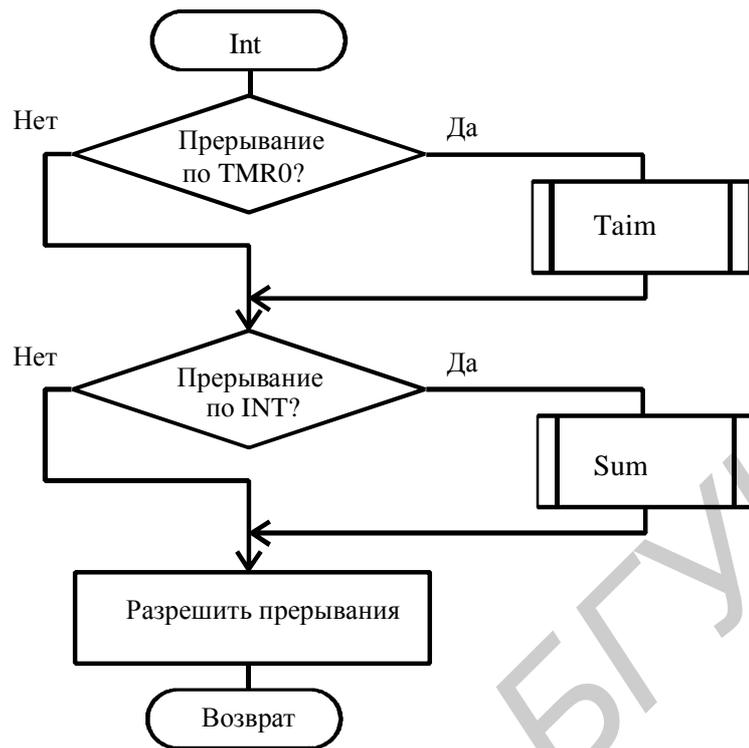


Рисунок 6.4 – Схема алгоритма подпрограммы Int

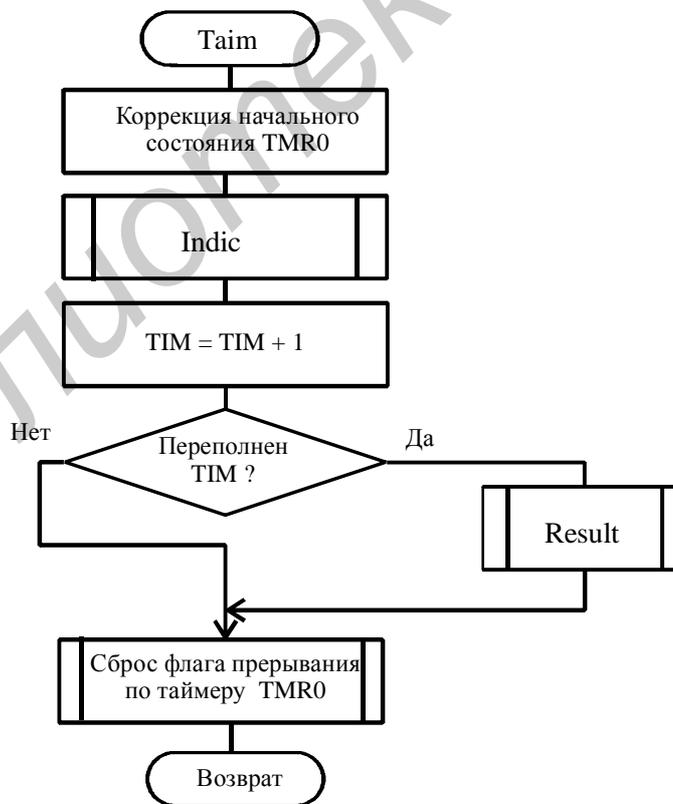


Рисунок 6.5 – Схема алгоритма подпрограммы Taim

**Sum.** Производит счет входных импульсов от датчика оборотов. При этом в счетчик оборотов (OB0, OB1) добавляется на каждый входной импульс константа 60 для обеспечения показаний «обороты в минуту» сразу без дополнительной процедуры преобразования единиц измерения. Счет ведется в двоично-десятичном коде. Обнуление счетчика оборотов производится по сбросу и по таймеру после считывания результата. Схема алгоритма подпрограммы Sum показана на рисунке 6.6.

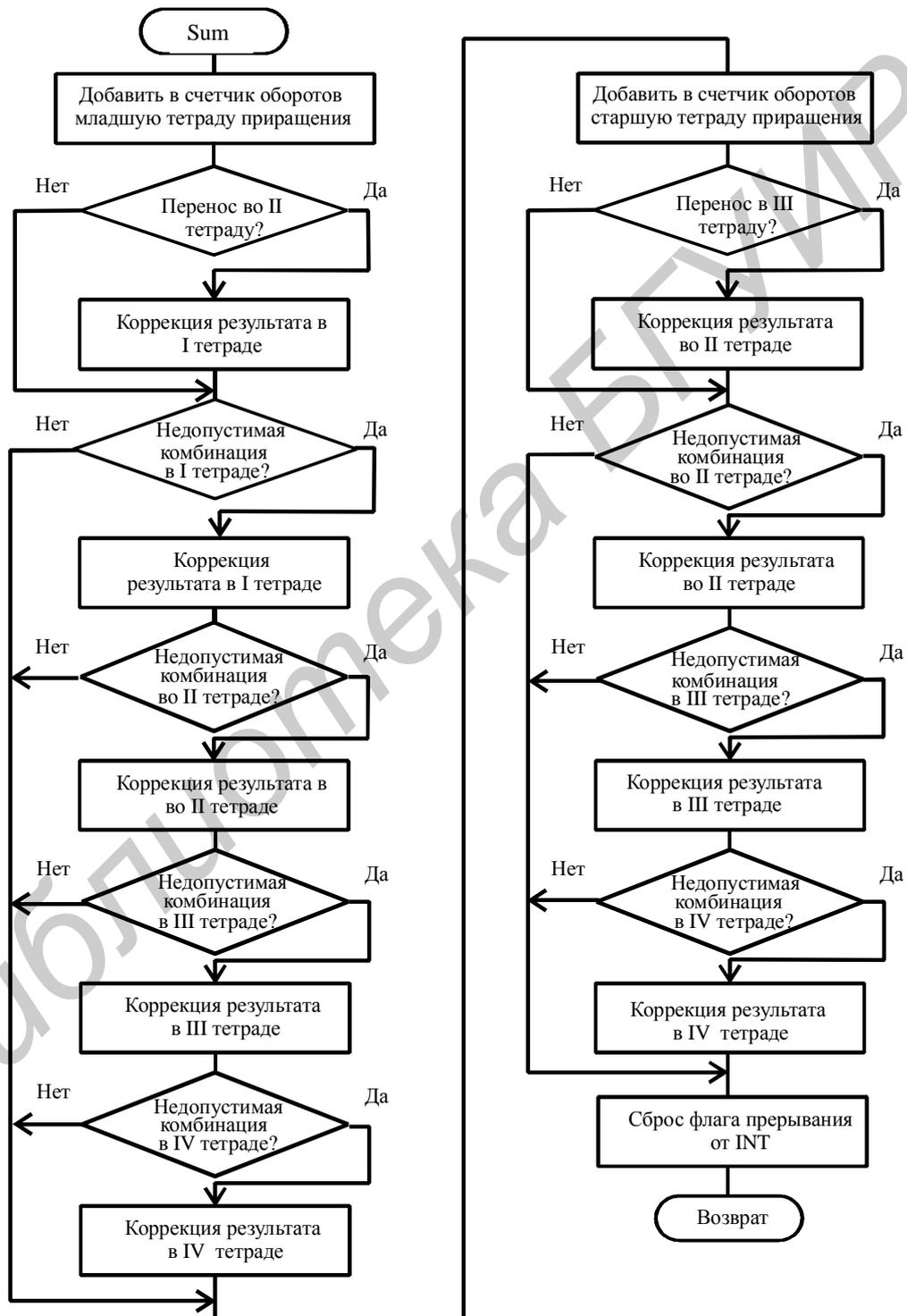


Рисунок 6.6 – Схема алгоритма подпрограммы Sum

**Indic.** Подпрограмма обслуживания табло. При каждом вызове преобразует отображаемую цифру текущего разряда в ее семисегментный код, выводит этот код на порт В и производит подсветку текущего разряда, затем смещает указатель текущего разряда на единицу влево в пределах 0...3. Схема алгоритма подпрограммы Indic показана на рисунке 6.7.

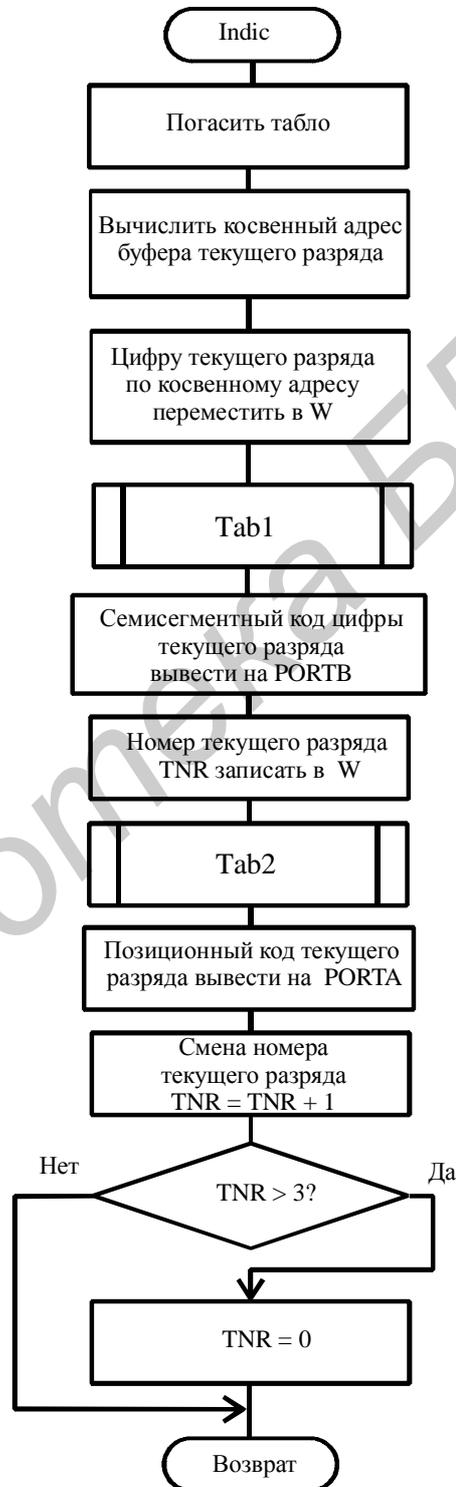


Рисунок 6.7 – Схема алгоритма подпрограммы Indic

**Result.** Выполняет съем накопленного на интервале наблюдения 1 с количества входных импульсов и перенос их в буфер табло. Схема алгоритма подпрограммы Result показана на рисунке 6.8.

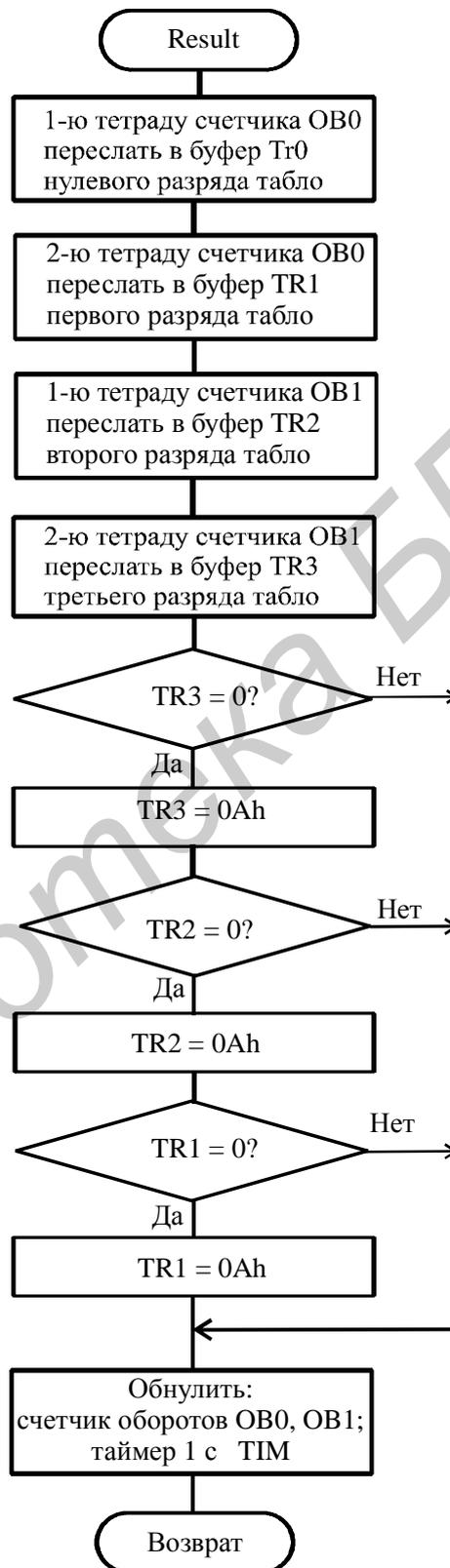


Рисунок 6.8 – Схема алгоритма подпрограммы Result

**Tab1.** Подпрограмма производит преобразование двоично-десятичной цифры в семисегментный код для отображения на цифровом табло. Перед вызовом подпрограммы преобразуемую цифру необходимо загрузить в регистр W микроконтроллера, результат преобразования после выполнения подпрограммы также помещается в W. Схема алгоритма подпрограммы Tab1 показана на рисунке 6.9.

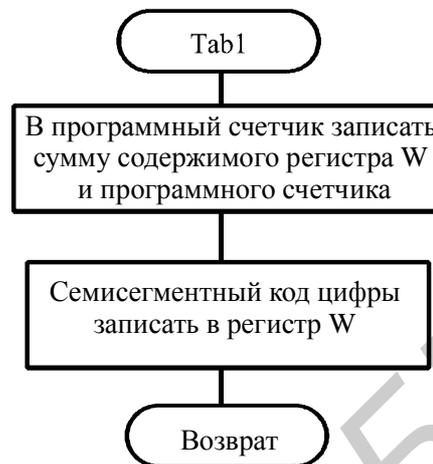


Рисунок 6.9 – Схема алгоритма подпрограммы Tab1

**Tab2.** Производит преобразование двоичного номера текущего разряда в позиционный код для подсветки текущего (отображаемого) разряда цифрового табло. Перед вызовом подпрограммы в W загрузить номер текущего разряда, результат после выполнения программы также помещается в W. Схема алгоритма подпрограммы Tab2 показана на рисунке 6.10.

В схемах алгоритмов и в программе использованы следующие переменные и константы.

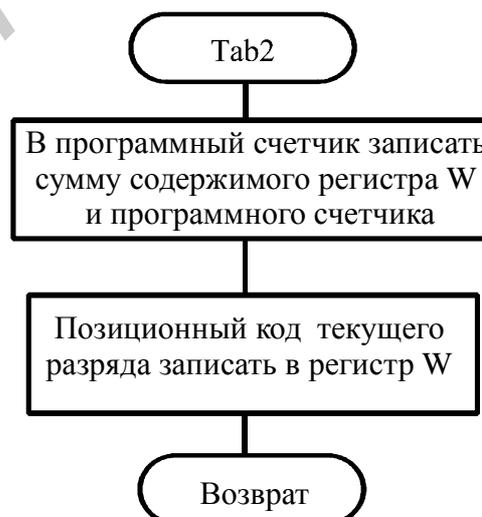


Рисунок 6.10 – Схема алгоритма подпрограммы Tab2

**TIM** – счетчик прерываний по переполнению таймера TMR0. Используется в программе для отсчета секундного интервала, в течение которого производится подсчет количества поступивших на вход INT импульсов от датчика вращения. Коэффициент деления предварительного делителя перед TMR0 и начальное значение TMR0 подбираются таким образом, чтобы при тактовой частоте микроконтроллера 4 МГц переполнение TIM (смена состояний от 0 до 255) происходило через 1 с.

**OB0, OB1** – счетчик оборотов (соответственно младший и старший байты). Подсчет количества поступивших от датчика вращения импульсов производится в двоично-десятичной системе счисления в 4-разрядном десятичном счетчике. При этом младший разряд десятичного счетчика занимает младшую тетраду OB0, а старший разряд – соответственно старшую тетраду OB1. Счетчик реализован программно по прерываниям от входного сигнала INT.

**DOB** – приращение счетчика оборотов на каждый импульс от датчика вращения, равное 60. Используется для получения в счетчике оборотов в конце секундного интервала измерения числа, соответствующего единице измерения частоты вращения «обороты в минуту».

**TNR** – номер текущего (подсвечиваемого) разряда табло, изменяется программно в пределах от 0 до 3.

**TR0, TR1, TR2, TR3** – буферы для хранения отображаемых на табло цифр 4-разрядного десятичного числа. (TR0 – младший разряд, TR3 – старший разряд).

#### 6.4 Текст управляющей программы на языке Ассемблер

```
*****  
;  
; ТАХОМЕТР  
*****  
; Программа реализует измерение частоты вращения вала  
; в пределах 0...9999 об./мин с погрешностью +/- 60 об./мин.  
; Используется датчик вращения, дающий один импульс на оборот. Импульсы  
; подаются на вход внешнего прерывания INT (порт RB0) и обрабатываются  
; программой по прерываниям.  
; Измерение частоты вращения вала производится путем подсчета количества  
; поступивших от датчика вращения импульсов на интервале 1 с.  
; Частота тактового генератора микроконтроллера 4 МГц.  
; Цифровое табло 4-разрядное, работает в динамическом режиме.  
; Младший правый разряд считается нулевым, старший левый – третьим.
```

; Светодиоды сегментов табло соединены по схеме с общим катодом.  
 ; Общие выводы разрядов табло обслуживаются выводами порта A:  
 ; разряд 0 – RA0, разряд 1 – RA1, разряд 2 – RA2, разряд 3 – RA3.  
 ; Сегменты разрядов обслуживаются выводами порта B:  
 ; a – RB1, b – RB2, c – RB3, d – RB4, e – RB5, f – RB6, g – RB7.  
 ; Для зажигания сегментов табло на порт B надо выводить единицы,  
 ; а на порт A – нули.  
 ; Подпрограмма обслуживания табло вызывается по прерываниям  
 ; от таймера TMR0.

-----  
 List p=16f628, f=inhx8m ; задание типа контроллера  
 ; и формата исполнимого файла .hex  
 #include p16f628.inc; подключение файла описания имен  
 ; и адресов внутренних регистров контроллера.  
 \_\_config \_CP\_OFF & \_DATA\_CP\_OFF & \_BODEN\_ON & \_PWRTE\_ON &  
 \_WDT\_OFF & \_LVP\_OFF & \_MCLRE\_ON & \_HS\_OSC  
 ; Директивой \_\_config заданы следующие режимы работы:  
 ; защита памяти программ – выключена,  
 ; защита памяти данных – выключена,  
 ; сброс по снижению уровня питания – включен,  
 ; таймер питания – включен,  
 ; сторожевой таймер – выключен,  
 ; низковольтное программирование – выключено,  
 ; вход внешнего сигнала СБРОС – включен,  
 ; тактовый генератор – HS (кварцевый высокочастотный).

-----  
 ;\*\*\*\*\***Распределение памяти**\*\*\*\*\*

;-----буфер табло-----

TNR EQU H'20' ; номер текущего (отображаемого) разряда

TR0 EQU H'21' ; буфер разряда 0

TR1 EQU H'22' ; буфер разряда 1

TR2 EQU H'23' ; буфер разряда 2

TR3 EQU H'24' ; буфер разряда 3

;-----программно реализованный счетчик-таймер интервалов 1 с -----

TIM EQU H'25' ;

;---- Счетчик входных импульсов от датчика вращения, 1 импульс на оборот

OB0 EQU H'26' ; младший байт (1-я и 2-я тетрады) – 0-й и 1-й разряды табло

OB1 EQU H'27' ; старший байт (3-я и 4-я тетрады) – 2-й и 3-й разряды табло

DOB EQU H'3C' ; приращение счетчика импульсов на один оборот,  
; для перевода показаний в об./мин.

-----  
;\*\*\*\*\*Векторы сброса и прерывания\*\*\*\*\*

```
org 0 ; Вектор сброса
goto Begin ; переход на начало основной программы
org 4 ; Вектор прерывания
goto Int ; переход на начало подпрограммы обработки прерывания
```

-----  
;\*\*\*\*\*Подпрограммы\*\*\*\*\*

### Tab1

; Преобразование двоично-десятичной цифры в семисегментный код.  
; Перед вызовом подпрограммы преобразуемую цифру загрузить в W,  
; результат после выполнения программы помещается в W.

```
addwf PCL
retlw B'01111110' ; код цифры 0
retlw B'00001100' ; код цифры 1
retlw B'10110110' ; код цифры 2
retlw B'10011110' ; код цифры 3
retlw B'11001100' ; код цифры 4
retlw B'11011010' ; код цифры 5
retlw B'11111010' ; код цифры 6
retlw B'00001110' ; код цифры 7
retlw B'11111110' ; код цифры 8
retlw B'11011110' ; код цифры 9
retlw B'00000000' ; пусто, разряд погашен
```

-----  
;\*\*\*\*\*Подпрограммы\*\*\*\*\*

### Tab2

; Преобразование двоичного номера текущего разряда в позиционный код.  
; Перед вызовом подпрограммы в W загрузить номер текущего разряда,  
; результат после выполнения программы – в W.

```
addwf PCL
retlw B'00001110' ; подсветка 0-го разряда
retlw B'00001101' ; подсветка 1-го разряда
retlw B'00001011' ; подсветка 2-го разряда
retlw B'00000111' ; подсветка 3-го разряда
```

-----  
;\*\*\*\*\*Подпрограммы\*\*\*\*\*

### Indic

; Обслуживание табло.  
 ; При каждом вызове преобразует отображаемую цифру текущего разряда  
 ; в ее семисегментный код, выводит этот код на порт В, производит подсветку  
 ; текущего разряда, затем смещает указатель текущего разряда  
 ; на 1 влево в пределах 0...3.

```

    bcf     STATUS,RP0      ; банк 0
    movlw  0xFF            ; погасить табло
    movwf  PORTA          ;
    movlw  TR0
    addwf  TNR,0          ; вычисление косвенного адреса текущего разряда
    movwf  FSR
    movf   INDF,0         ; код числа разряда – в W
    call   Tab1           ; получение 7-сегментного кода текущего разряда
    movwf  PORTB          ; семисегментный код – на табло
    movf   TNR,0
    call   Tab2           ;
    movwf  PORTA          ; подсветка текущего разряда
    incf   TNR,1          ; TNR = TNR + 1
    btfsc  TNR,H'2'       ;
    clrf   TNR            ;
    return
  
```

## Sum

; Производит счет входных импульсов и нормировку для  
 ; обеспечения показаний в оборотах в минуту.  
 ; Счет ведется в двоично-десятичном коде.  
 ; Обнуление счетчика при включении и по таймеру после  
 ; считывания результата.

```

    bcf     STATUS,RP0      ; банк 0
  
```

; Задание приращения счетчика импульсов в двоично-десятичном  
 ; формате (две тетрады)

```

    movlw  0x60            ; !!! десятичное число 60 !!!!!
    movwf  DOB             ;-----
    movlw  0x0F           ;
    andwf  DOB,0          ;
    addwf  OB0,1           ; добавить младшую тетраду приращения
    btfss  STATUS,DC      ; проверка наличия переноса
  
```

```

; из I во II тетраду
goto      M_2      ;-----
movlw     0x06     ;
addwf    OB0,1    ; коррекция при переносе из I во II тетраду
goto      M_3      ;-----
M_2 movlw     0x06     ; проверка и коррекция
Addwf    OB0,0    ; недопустимой комбинации
btfss    STATUS,DC ; I тетрады
goto      M_4      ; выход в конец процедуры!
movwf    OB0      ;-----
M_3 ;проверка и коррекция недопустимой комбинации II тетрады
movlw    0x60 ;
addwf    OB0,0 ;
btfss    STATUS,C ;
goto      M_4      ; выход в конец процедуры!
movwf    OB0
incf     OB1,1 ;
; проверка и коррекция III тетрады
movlw    0x06
addwf    OB1,0
btfss    STATUS,DC
goto      M_4
movwf    OB1
; проверка и коррекция IV тетрады
movlw    0x60
addwf    OB1,0
btfss    STATUS,C
goto      M_4
movwf    OB1
M_4 ; Добавить старшую тетраду приращения
movlw    0xF0
andwf    DOB,0
addwf    OB0,1
btfss    STATUS,C
goto      M_5
movlw    0x60
addwf    OB0,1
goto      M_6

```

M\_5

```
movlw    0x60
addwf    OB0,0
btfss    STATUS,C
goto     M_7
movwf    OB0
```

M\_6

```
incf     OB1
movlw    0x06
addwf    OB1,0
btfss    STATUS,DC
goto     M_7
movwf    OB1
movlw    0x60
addwf    OB1,0
btfss    STATUS,C
goto     M_7
movwf    OB1
```

M\_7

```
bcf      INTCON,INTF    ; сброс флага прерывания
return
```

-----

### **Taim**

- ; Корректирует начальное значение TMR0.
- ; Вызывает программу Indic.
- ; Увеличивает содержимое программного таймера на единицу.
- ; Следит за реальным временем.
- ; Вызывает программу Result.

```
movlw    0x0C    ; 12 в десятичной
movwf    TMR0    ; коррекция TMR0
call     Indic
bcf      STATUS,RP0    ; банк 0
incfsz   TIM,1
goto     M_0
call     Result
```

M\_0

```
bcf      INTCON,T0IF    ; сброс флага прерывания
return
```

;------

## Result

; Съем показания счетчика оборотов, запись результата в буфер табло.

```
bcf      STATUS,RP0    ; банк 0
movlw   H'0F'
andwf   OB0,0
movwf   TR0
swapf   OB0,0
andlw   0x0F
movwf   TR1
movlw   H'0F'
andwf   OB1,0
movwf   TR2
swapf   OB1,0
andlw   0x0F
movwf   TR3
```

; Гашение незначащих нулей на табло

```
movf    TR3,1
btfss   STATUS,Z
goto    M_1
movlw   0x0A
movwf   TR3
movf    TR2,1
btfss   STATUS,Z
goto    M_1
movlw   0x0A
movwf   TR2
movf    TR1,1
btfss   STATUS,Z
goto    M_1
movlw   0x0A
movwf   TR1
```

; Обнуление счетчика оборотов и таймера 1 секунда.

```
M_1 clrf    OB0
     clrf    OB1
     clrf    TIM
     return
```

;------

## Int

; Подпрограмма обработки прерываний  
; определяет причину прерывания и вызывает  
; соответствующую подпрограмму – Taim (если прерывание от TMR0)  
; или Sum (если прерывание от входа INT).

```
btfsc    INTCON,T0IF  
call     Taim  
btfsc    INTCON,INTF  
call     Sum  
retfie
```

-----

\*\*\*\*\***Основная программа**\*\*\*\*\*

; Выполняет начальные установки регистров и режимов работы.  
; Ожидает прерывания от таймера и сигнального входа INT.

## Begin

```
bcf      STATUS,RP0    ; банк 0
```

; Установка портов в состояния, обеспечивающие режим "табло погашено"

```
clrf     PORTB  
movlw   0xFF  
movwf   PORTA
```

; Отключение компараторов

```
movlw   0x07  
movwf   CMCON
```

; Установки режимов работы портов

```
bsf     STATUS,RP0    ; банк 1
```

```
movlw   0x10  
movwf   TRISA          ; RA0, RA1, RA2 и RA3 – выходы
```

```
movlw   0x01  
movwf   TRISB          ; RB0 – вход, остальные – выходы.
```

; Настройка таймера TMR0

```
bcf     OPTION_REG,PSA ; предварительный делитель перед TMR0
```

```
bcf     OPTION_REG,T0CS ; внутреннее тактирование TMR0
```

```
bcf     OPTION_REG,PS2 ; коэффициент деления предварительного  
; делителя 1:16
```

```
bcf     STATUS,RP0    ; банк 0
```

; Сброс таймеров

```
clrf    TMR0          ; аппаратного
```

```
clrf    TIM           ; программного
```

; Сброс счетчика входных импульсов

clrf OB0

clrf OB1

; Сброс буфера табло

clrf TNR

clrf TR0

clrf TR1

clrf TR2

clrf TR3

movlw 0xB0 ; сброс всех флагов прерывания и

movwf INTCON ; разрешение прерываний от таймера TMR0

; и входа прерываний INT

;

**Wait goto Wait** ; Ожидание прерывания

;

End ; Конец ассемблирования

## ЛИТЕРАТУРА

- 1 Безуглов, Д. А. Цифровые устройства и микропроцессоры : учеб. пособие для вузов / Д. А. Безуглов, И. В. Калиенко. – Ростов н/Д : Феникс, 2008. – 468 с.
- 2 Микропроцессорные системы : учеб. пособие для вузов / Е. К. Александров [и др.], под общ. ред. Д. В. Пузанкова. – СПб. : Политехника, 2002. – 935 с.
- 3 Предко, М. Справочник по PIC-микроконтроллерам / М. Предко ; пер. с англ. – М. : ДМК Пресс, 2004. – 512 с.
- 4 Яценков, В. С. Микроконтроллеры Microchip. Практическое руководство / В. С. Яценков. – М. : Горячая линия – Телеком, 2002. – 296 с.
- 5 Сташин, В. В. Проектирование цифровых устройств на однокристальных микроконтроллерах / В. В. Сташин, А. В. Урусов, О. Ф. Мологонцева. – М. : Энергоатомиздат, 1990. – 224 с.
- 6 Левкович, В. Н. Микропроцессорные устройства : учеб. пособие для студ. радиотех. спец. всех форм обуч. / В. Н. Левкович, О. В. Шабров. – Минск : БГУИР, 2007. – 99 с.
- 7 Левкович, В. Н. Цифровые и микропроцессорные устройства : лаб. практикум для студ. спец. 1-39 01 02 «Радиоэлектронные системы», 1-39 01 03 «Радиоинформатика», 1-39 01 04 «Радиоэлектронная защита информации». В 2 ч. Ч. 1 / В. Н. Левкович. – Минск : БГУИР, 2005. – 38 с.
- 8 Левкович, В. Н. Цифровые и микропроцессорные устройства : лаб. практикум для студ. спец. 1-39 01 02 «Радиоэлектронные системы», 1-39 01 03 «Радиоинформатика», 1-39 01 04 «Радиоэлектронная защита информации». В 2 ч. Ч. 2 / В. Н. Левкович, А. А. Кашеев. – Минск : БГУИР, 2006. – 36 с.
- 9 Левкович, В. Н. Конструирование программ на Ассемблере для микроконтроллеров семейства PICmicro : учеб. пособие по курсу «Цифровые и микропроцессорные устройства» для студ. спец. 1- 39 01 01 «Радиотехника» и 1- 39 01 02 «Радиоэлектронные системы» всех форм обуч. / В. Н. Левкович, А. С. Грицук, И. В. Коваленко. – Минск : БГУИР, 2004. – 80 с.
- 10 Разработка программ для микроконтроллеров фирмы MICROCHIP в интегрированной среде MPLAB IDE 7 : метод. указ. к лаб. работам по курсу «Микропроцессорные устройства» для студ. радиотех. спец. всех форм обуч. / А. А. Казека, А. В. Мартинович, И. Г. Давыдов. – Минск : БГУИР, 2008. – 31 с.
- 11 Предварительный стандарт предприятия. Дипломные проекты (работы). Общие требования. СТП П-01-2008 / сост. А. Т. Доманов, Н. И. Сорока. – Минск : БГУИР, 2009.

## СОДЕРЖАНИЕ

1 Цели и задачи курсового проектирования.....	3
2 Тематика курсовых работ.....	4
3 Этапы проектирования.....	8
4 Требования к содержанию пояснительной записки.....	12
5 Требования к оформлению пояснительной записки.....	14
6 Примеры выполнения основных разделов курсовой работы на тему «Цифровой тахометр».....	21
Литература.....	39

Библиотека БГУИР

Учебное издание

**Левкович** Василий Николаевич  
**Шабров** Олег Васильевич  
**Казека** Александр Анатольевич  
**Мартинovich** Алексей Васильевич

## **МИКРОПРОЦЕССОРНЫЕ УСТРОЙСТВА**

Методическое пособие  
к курсовому проектированию  
для студентов радиотехнических специальностей  
всех форм обучения

Редактор Н. В. Гриневич  
Корректор Н. Е. Батурчик

---

Подписано в печать 1.06.2009.  
Гарнитура «Таймс».  
Уч.-изд. л. 2,1.

Формат 60x84 1/16.  
Печать ризографическая.  
Тираж 150 экз.

Бумага офсетная.  
Усл. печ. л. 2,56.  
Заказ 155.

---

Издатель и полиграфическое исполнение: Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники»  
ЛИ №02330/0494371 от 16.03.2009. ЛП №02330/0494175 от 03.04.2009.  
220013, Минск, П. Бровки, 6