РЕФЕРАТ

Пояснительная записка к дипломному проекту: 88 страниц, 15 рисунков, 21 таблица, 24 источника, 5 приложений, 3 листа чертежей формата А1.

Объект исследований: разработка музыкального звонка с двумя режимами работы: автономный и от сети.

Предмет исследования: электромузыкальный звонок.

В первом разделе рассмотрены общие принципы разработки устройств на микроконтроллерах.

Во втором разделе рассматриваются вопросы практической разработки электромузыкального звонка. Разрабатываются структурная, функциональная и принципиальная схемы. Составляется алгоритм и программа для микроконтроллера.

В третьем разделе выполнен экономический расчет объекта анализа, производится сравнительная характеристика с устройствами-аналогами.

В четвертом разделе проведены расчеты вентиляции, природного и искусственного освещения, уровня шума. Полученные значения сопоставлены с нормативными.

Данное устройство может быть рекомендовано к внедрению в производство.

АЛГОРИТМ, БЛОК ПИТАНИЯ, КНОПКА, МИКРОКОНТРОЛЛЕР, МИКРОПРОЦЕССОРНАЯ СИСТЕМА, ПРОГРАММА, СВЕТОДИОД

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ, ЕДИНИЦ, СОКРАЩЕНИЙ И ТЕРМИНОВ 3

ВВЕДЕНИЕ

1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ

2 РАЗРАБОТКА МУЗЫКАЛЬНОГО ЗВОНКА С ДВУМЯ РЕЖИМАРАБОТЫ: АВТОНОМНЫЙ И ОТ СЕТИ

2.1 Постановка задачи

2.2 Разработка структурной схемы устройства и функциональной спецификации

2.3 Аппаратные средства микроконтроллеров серии ATtiny2313

2.4 Разработка функциональной схемы устройства

2.5 Разработка алгоритма работы устройства

2.6 Разработка программного обеспечения микроконтроллера

2.6.1 Описание программы

2.6.2 Процедура вычисления адреса

2.6.3 Текст программы

2.6.4 Особенности программы

2.6.5 Подпрограмма формирования задержки

2.6.6 Программа на языке СИ

2.6.7 Описание программы (листинг 2

2.7 Выбор, описание и расчеты элементной базы

2.8 Разработка схемы принципиальной

3 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ОБЪЕКТА РАЗРАБОТКИ

3.1 Расчет расходов на ПО для микроконтроллера

3.2 Расчет расходов на создание ПО

3.3 Расчет стоимости разработки конструкторской документации и сборки устройства

3.4 Расчет расходов на стадии производства изделия

3.5 Анализ устройств-аналогов

4 ОХРАНА ТРУДА

4.1 Требования к производственным помещениям

4.1.1 Окраска и коэффициенты отражения

.4.1.2 Освещение

4.1.3 Параметры микроклимата

4.1.4 Шум и вибрация

4.1.5 Электромагнитное и ионизирующее излучения

4.2 Эргономические требования к рабочему месту

4.3 Режим труда

4.4 Расчет освещенности

4.5. Расчет вентиляции

4.6 Расчет уровня шума

ВЫВОДЫ

ПРИЛОЖЕНИЯ

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ,

ЕДИНИЦ, СОКРАЩЕНИЙ И ТЕРМИНОВ

АЦП – аналого-цифровой преобразователь

КМОП – комплементарная логика на транзисторах металл-оксид-полупроводник

МК - микроконтроллер

МПС – микропроцессорная система

ЦПУ – центральное процессорное устройство

ШИМ – широтно импульсная модуляция

ВВЕДЕНИЕ

Разработка систем управления и контроля с использованием однокристальных микроконтроллеров в настоящее время переживает настоящий бум. Системы на базе микроконтроллеров используются практически во всех сферах жизнедеятельности человека, и каждый день появляются все новые и новые области применения этих устройств. В последнее время в связи с бурным развитием электроники и схемотехники расширились возможности и самих микроконтроллеров, позволяющие выполнять многие задачи, ранее недоступные для реализации, такие, например, как обработка аналоговых сигналов. Одним из наиболее ранних микроконтроллеров, появившихся на рынке, является микроконтроллер ATtiny, разработанный фирмой Intel более двадцати лет назад. Несмотря на столь приличный возраст, классический ATtiny и его клоны в настоящее время остаются одними из наиболее популярных при разработке систем управления и контроля. Хорошо продуманная архитектура и интуитивно понятная система команд оказывают решающее влияние на выбор многих разработчиков аппаратно-программных систем.

Однокристальные (однокорпусные) микроконтроллеры представляют собой приборы, конструктивно выполненные в виде БИС и включающие в себя следующие составные части: микропроцессор, память программ и память данных, а также программируемые интерфейсные схемы для связи с внешней средой.

Мировая промышленность выпускает огромную номенклатуру микроконтроллеров. По области применения их можно разделить на два класса: специализированные, предназначенные для применения в какой-либо одной конкретной области (контроллер для телевизора, контроллер для модема) и универсальные, которые не имеют конкретной специализации и могут применяться в самых различных областях микроэлектроники, с помощью которых можно создать как любое из перечисленных выше устройств, так и принципиально новое устройство.

Тема данной работы - «Разработка музыкального звонка с двумя режимами работы: автономным и от сети», которая является предметом исследования.

Объектом исследования является устройство (бытовой электромузыкальный звонок), предназначенное для воспроизведения ранее запрограммированных мелодий, при нажатии и удержании кнопки. Устройство должно содержать минимум компонентов, быть простым в изготовлении и эксплуатации, иметь возможность работать в режимах: автономном и от сети.

Данная тема является актуальной, т.к. электромузыкальные звонки пользуются повышенным спросом у населения.

РАЗДЕЛ 1 ТЕОРЕТИЧЕСКИЕ ОСНОВЫ РАЗРАБОТКИ

Микропроцессорная система (МПС) на основе микроконтроллера (МК) используются чаще всего в качестве встроенных систем для решения задач управления некоторым объектом . Важной особенностью данного применения является работа в реальном времени, т.е. обеспечение реакции на внешние события в течение определенного временного интервала. Такие устройства получили название контроллеров.

Перед разработчиком МПС стоит задача реализации полного цикла проектирования, начиная от разработки алгоритма функционирования и заканчивая комплексными испытаниями в составе изделия. Методология проектирования контроллеров может быть представлена так, как показано на рис. 1.1.

В техническом задании формулируются требования к контроллеру с точки зрения реализации определенной функции управления. Техническое задание включает в себя набор требований, который определяет, что пользователь хочет от контроллера и что разрабатываемый прибор должен делать.

На основании требований пользователя составляется функциональная спецификация, которая определяет функции, выполняемые контроллером для пользователя после завершения проектирования, уточняя тем самым, насколько устройство соответствует предъявляемым требованиям. Она включает в себя описания форматов данных, как на входе, так и на выходе, а также внешние условия, управляющие действиями контроллера.

Этап разработки алгоритма управления является наиболее ответственным, поскольку ошибки данного этапа обычно обнаруживаются только при испытаниях законченного изделия и приводят к необходимости дорогостоящей переработки всего устройства. Разработка алгоритма обычно сводится к выбору одного из нескольких возможных вариантов алгоритмов, отличающихся соотношением объема программного обеспечения и аппаратных средств.

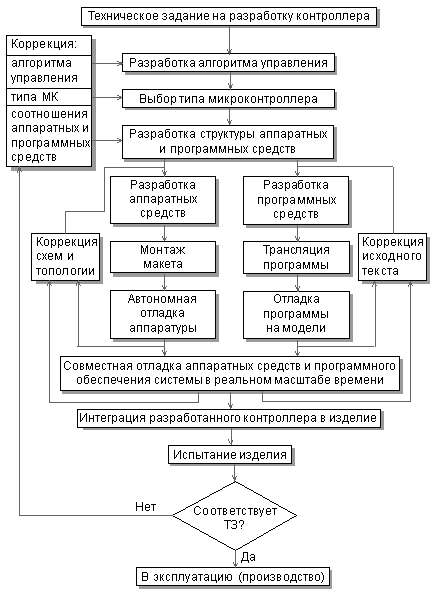


Рисунок 1.1- Основные этапы разработки контроллера

При этом необходимо исходить из того, что максимальное использование аппаратных средств упрощает разработку и обеспечивает высокое быстродействие контроллера в целом, но сопровождается, как правило, увеличением стоимости и потребляемой мощности. При выборе типа МК учитываются следующие основные характеристики:

- разрядность;

- быстродействие;

- набор команд и способов адресации;

- требования к источнику питания и потребляемая мощность в различных режимах;

- объем ПЗУ программ и ОЗУ данных;

- возможности расширения памяти программ и данных;

- наличие и возможности периферийных устройств, включая средства поддержки работы в реальном времени (таймеры, процессоры событий и т.п.);

- возможность перепрограммирования в составе устройства;

- наличие и надежность средств защиты внутренней информации;

- возможность поставки в различных вариантах конструктивного исполнения;

- стоимость в различных вариантах исполнения;

- наличие полной документации;

- наличие и доступность эффективных средств программирования и отладки МК;

- количество и доступность каналов поставки, возможность замены изделиями других фирм.

Список этот не является исчерпывающим.

Номенклатура выпускаемых в настоящее время МК исчисляется тысячами типов изделий различных фирм. Современная стратегия модульного проектирования обеспечивает потребителя разнообразием моделей МК с одним и тем же процессорным ядром. Такое структурное разнообразие открывает перед разработчиком возможность выбора оптимального МК, не имеющего функциональной избыточности, что минимизирует стоимость комплектующих элементов.

РАЗДЕЛ 2 РАЗРАБОТКА МУЗЫКАЛЬНОГО ЗВОНКА С ДВУМЯ РЕЖИМАМИ РАБОТЫ: АВТОНОМНЫЙ И ОТ СЕТИ

2.1 Постановка задачи

Требуется разработать устройство, предназначенное для воспроизведения простых одноголосых мелодий, записанных в память программ на этапе программирования. Устройство должно иметь семь различных мелодий, которые включаются по желанию. Каждой из кнопок должна соответствовать своя мелодия. Мелодия воспроизводится при нажатии и удержании кнопки. При отпускании кнопки воспроизведение мелодий прекращается.

Питание данного устройства должно осуществляться в двух режимах: автономно и от сети.

Данное устройство рекомендуется использовать как электромузыкальный звонок.

2.2 Разработка структурной схемы устройства и функциональной

спецификации

Структурная схема разрабатываемого электромузыкального звонка приведена на рис. 2.1.

S8

Мелодия 4

S4

Мелодия 5

S5

Э/кл.

Звуковой излучатель

Мелодия 6

S6

Мелодия 7

S7

**ИП**

220 В

Рисунок 2.1- Структурная схема электромузыкального звонка

Функциональная спецификация представляет собой:

1. Входы

а. 7 кнопок выбора мелодий (S1-S7);

b. Кнопка запуска электромузыкального звонка (S8);

с. Источник бесперебойного электропитания звонка (ИП).

1. Выходы

а. Электронный ключ (Э/кл);

b. Звуковой динамик (Звуковой излучатель).

1. Функции

а. Запись мелодии в память, при нажатии кнопки S8;

1. Воспроизведение мелодии из памяти;
2. Осуществление бесперебойного электропитания в двух режимах: автономном и от сети.

2.3 Аппаратные средства микроконтроллеров серии ATtiny2313

В разработке электромузыкального звонка предлагается использовать, широко распространенный, относительно недорогой и надежный в эксплуатации 8 битный AVR микроконтроллер серии ATtiny2313 с 2 КБ программируемой в системе *Flash* памяти.

ATtiny2313 - низкопотребляющий 8 битный КМОП микроконтроллер с AVR RISC архитектурой. Выполняя команды за один цикл, ATtiny2313 достигает производительности 1 MIPS при частоте задающего генератора 1 МГц, что позволяет разработчику оптимизировать отношение потребления к производительности.

AVR ядро объединяет богатую систему команд и 32 рабочих регистра общего назначения. Все 32 регистра непосредственно связаны с арифметико-логическим устройством (АЛУ), что позволяет получить доступ к двум независимым регистрам при выполнении одной команды. В результате эта архитектура позволяет обеспечить в десятки раз большую производительность, чем стандартная CISC архитектура.

ATtiny2313 имеет следующие характеристики: 2 КБ программируемой в системе Flash память программы, 128 байтную EEPROM память данных, 128 байтное SRAM (статическое ОЗУ), 18 линий ввода - вывода общего применения, 32 рабочих регистра общего назначения, однопроводный интерфейс для встроенного отладчика, два гибких таймера/счетчика со схемами сравнения, внутренние и внешние источники прерывания, последовательный программируемый USART, универсальный последовательный интерфейс с детектором стартового условия, программируемый сторожевой таймер со встроенным генератором и три программно инициализируемых режима пониженного потребления. В режиме Idle останавливается ядро, но ОЗУ, таймеры/счетчики и система прерываний продолжают функционировать. В режиме Power-down регистры сохраняют свое значение, но генератор останавливается, блокируя все функции прибора до следующего прерывания или аппаратного сброса. В Standby режиме задающий генератор работает, в то время как остальная часть прибора бездействует. Это позволяет очень быстро запустить микропроцессор, сохраняя при этом в режиме бездействия мощность.

Прибор изготовлен по высокоплотной энергонезависимой технологии изготовления памяти компании Atmel. Встроенная ISP Flash позволяет перепрограммировать память программы в системе через последовательный SPI интерфейс или обычным программатором энергонезависимой памяти. Объединив в одном кристалле 8- битное RISC ядро с самопрограммирующейся в системе Flash памятью, ATtiny2313 стал мощным микроконтроллером, который дает большую гибкость разработчика микропроцессорных систем.

ATtiny2313 поддерживается различными программными средствами и интегрированными средствами разработки, такими как компиляторы C, макроассемблеры, программные отладчики/симуляторы, внутрисхемные эмуляторы и ознакомительные наборы.

Характеристики микроконтроллера ATtiny2313

AVR RISC архитектура

AVR - высококачественная и низкопотребляющая RISC архитектура

120 команд, большинство которых выполняется за один тактовый цикл

32 8 битных рабочих регистра общего применения

Полностью статическая архитектура

ОЗУ и энергонезависимая память программ и данных

2 КБ самопрограммируемой в системе Flash памяти программы, способной выдержать 10 000 циклов записи/стирания

128 Байт программируемой в системе EEPROM памяти данных, способной выдержать 100 000 циклов записи/стирания

128 Байт встроенной SRAM памяти (статическое ОЗУ)

Программируемая защита от считывания Flash памяти программы и EEPROM памяти данных

Характеристики периферии:

Один 8- разрядный таймер/счетчик с отдельным предделителем

Один 16-разрядный таймер/счетчик с отдельным предделителем, схемой сравнения, схемой захвата и двумя каналами ШИМ

Встроенный аналоговый компаратор

Программируемый сторожевой таймер со встроенным генератором

USI - универсальный последовательный интерфейс

Полнодуплексный UART

Специальные характеристики микроконтроллера :

Встроенный отладчик debugWIRE

Внутрисистемное программирование через SPI порт

Внешние и внутренние источники прерывания

Режимы пониженного потребления Idle, Power-down и Standby

Усовершенствованная схема формирования сброса при включении

Программируемая схема обнаружения кратковременных пропаданий питания

Встроенный откалиброванный генератор

Порты ввода - вывода и корпусное исполнение

18 программируемых линий ввода - вывода

20 выводной PDIP, 20 выводной SOIC и 32 контактный MLF корпуса

Диапазон напряжения питания:

от 1.8 до 5.5 В

Рабочая частота:

0 - 16 МГц

Потребление

Активный режим:

300 мкА при частоте 1 МГц и напряжении питания 1.8 В

20 мкА при частоте 32 кГц и напряжении питания 1.8 В

Режим пониженного потребления

0.5 мкА при напряжении питания 1.8 В

Блок- схема ATtiny2313 представлена на рисунке 2.2.

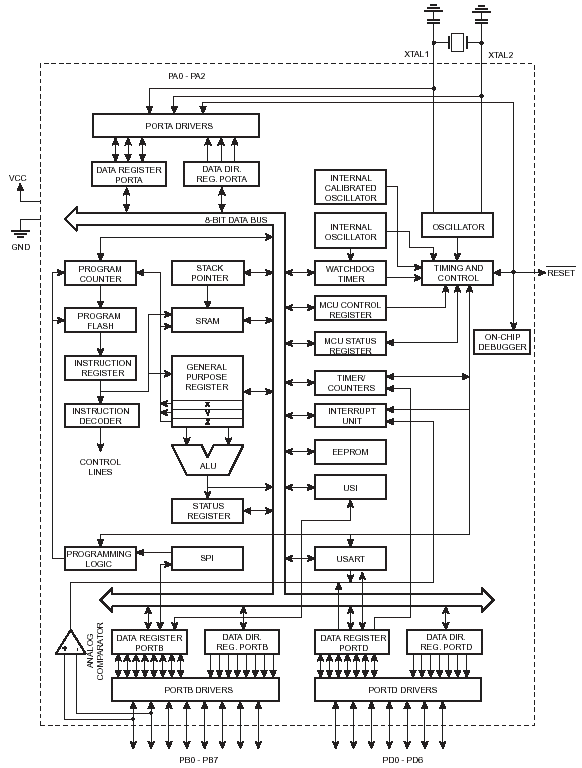


Рисунок 2.2 - Блок-схема микроконтроллера ATtiny2313

Расположение выводов МК ATtiny2313 приведено на рисунке 2.3.

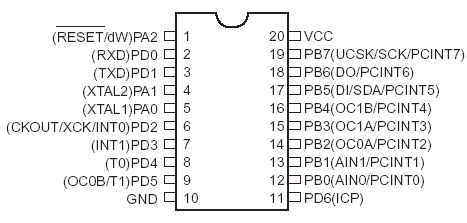


Рисунок 2.3 - Расположение выводов микроконтроллера ATtiny2313

2.4 Разработка функциональной схемы устройства

В проектируемом устройстве можно выделить следующие функциональные блоки: микроконтроллер ATtiny 2313, кнопка запуска электромузыкального звонка, кнопки выбора мелодии, кварцевый резонатор, электронный ключ, звуковой излучатель, источник питания. Функциональная схема электромузыкального звонка приведена на рисунке 2.4.

При нажатии на кнопку S8 (дверная кнопка) и любой выбранной кнопке S1-S7 (кнопки выбора мелодии) производится запуск работы устройства, собранного на МК ATtiny 2313. Кварцевый резонатор служит для стабилизации частоты кварцевого генератора. Электронный ключ служит для развязки выхода микроконтроллера с низкоомным входом громкоговорителя. Источник питания служит для электропитания всего устройства. Источник питания работает в двух режимах: от сети (схема бестрансформаторного питания) и автономно (от элементов постоянного тока).

Кнопки

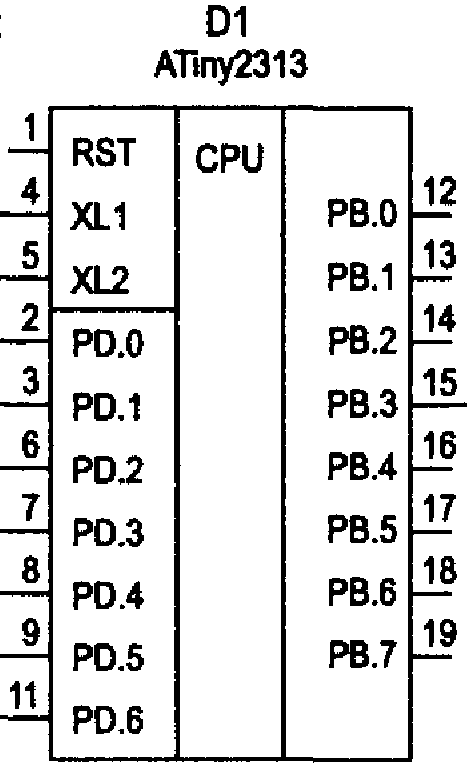
Выбора

мелодии

Кварцевый

резонатор

Эл. ключ



**20**

Источник питания

Рисунок 2.4 - Функциональная схема электромузыкального звонка

2.5 Разработка алгоритма работы устройства

Для начала нам нужно придумать, как мы будем хранить мелодии в памяти. Для того, чтобы в памяти можно было что-либо хранить, нужно сначала это что-то каким-либо способом закодировать. Любая мелодия состоит из нот. Каждая нота имеет свой тон (частоту) и длительность звучания. Для того, чтобы закодировать тон ноты, можно просто все ноты пронумеровать по порядку. Удобнее нумеровать, начиная с самого низкого тона.

Известно, что весь музыкальный ряд делится на октавы.В современном музыкальном ряду каждая октава делится на 12 нот. Семь основных нот и пять дополнительных.

Деление на основные и дополнительные ноты сложилось исторически. В настоящее время используется музыкальный строй, в котором все 12 нот одной октавы равнозначны. Частоты любых двух соседних нот отличаются друг от друга в одинаковое количество раз. При этом частоты одноименных нот в двух соседних октавах отличаются ровно в два раза.

Для нас же важно то, что коды всем этим нотам мы должны присваивать в порядке возрастания частоты. Начнем мы с ноты «До» первой октавы. Для электромузыкального звонка более низкие ноты не нужны. В таблице 2.1показаны коды для всей первой октавы. Следующая, вторая октава продолжает первую и по кодировке, и по набору частот. Так нота «До» второй октавы будет иметь код 13, а частоту *f12*= *fo*∙ *2*. А нота «Ре» второй октавы будет иметь код 14 и частоту *f13*=*f1* ∙ *2*. И так далее.

Музыкальная длительность тоже легко кодируется. В музыке применяют не произвольную длительность, а длительность, выраженную долями от целой (см. таблицу 2.2).В зависимости от темпа реальная длительность целой ноты меняется. Для сохранения мелодии необходимо соблюдать лишь соотношения между длительностями. Поэтому нам необходимо закодировать лишь семь вариантов длительности. Присвоим им коды от 0 до 6. Например так, как это показано в графе «Код» таблицы 2.2.Назначение графы «Коэффициент деления» мы пока опустим.

Таблица 2.1-Кодировка нот первой октавы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код | Нота | Частота | Код | Нота | Частота |
| 1 | До | *fo* | 7 | Фа# | *f6=f5/K* |
| 2 | До# | *f1=fo/K* | 8 | Соль | *f7=f6/K* |
| 3 | Ре | *f2=f1/K* | 9 | Соль\* | *f8=f7/K* |
| 4 | Ре# | *f3=f2/K* | 10 | Ля | *f9=f8/K* |
| 5 | Ми | *f4=f3/K* | 11 | Ля# | *f10=f9/K* |
| 6 | Фа | *f5=f4/K* | 12 | Си | *f11=f10/K* |

Для справки:



Таблица 2.2 - Кодирование музыкальных длительностей

|  |  |  |
| --- | --- | --- |
| Код | Длительность | Коэффициент деления |
| 0 | 1 (целая) | 64 |
| 1 | 1/2(половинная) | 128 |
| 2 | 1/4 (четверть) | 256 |
| 3 | 1/8 (восьмая) | 512 |
| 4 | 1/16 (шестнадцатая) | 1024 |
| 5 | 1/32 (тридцать вторая) | 2048 |
| 6 | 1/64 (шестьдесят четвертая) | 4096 |

Кроме нот, любая мелодия обязательно содержит музыкальные паузы.

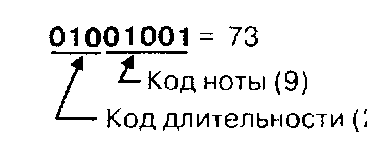
Определение. Паузы— это промежутки времени, когда ни один звук не звучит. Длительность музыкальных пауз принимает точно такие же значения, как и длительность нот.

В связи с этим удобно представить паузу как еще одну ноту. Ноту без звука. Такой ноте логично присвоить нулевой код.

Кодируем мелодии.

Для экономии памяти удобнее каждую нотукодировать одним байтом. Договоримся, что три старших бита мы будем использовать для кодирования длительности ноты, а оставшиеся пять битов — для кодирования ее тона. Пятью битами можно закодировать до 32 разных нот, что вполне хватит для электромузыкального звонка.

Итак, если использовать приведенный выше способ кодирования, то код ноты ля первой октавы длительностью 1/4 в двоичном виде будет равен:



Теперь мы можем приступать к кодированию мелодий.Для того, чтобы закодировать мелодию, нам нужна ее нотная запись.Используя нотную запись, мы должны присвоить каждой ноте и каждой музыкальной паузе свой код.

Цепочка таких кодов и будет представлять собой закодированную мелодию. По условиям задачи наш электромузыкальный звонок должен уметь воспроизводить семь разных мелодий. Коды всех семи мелодий мы разместим в программной памяти микроконтроллера.

Как определить конец каждой мелодии?Для того, чтобы компьютер знал, где заканчивается каждая мелодия, используем код 255 в качестве признака конца.

Теперь нам нужно придумать, как микроконтроллер будет находить начало каждой мелодии. Все мелодии имеют разную длину, а в памяти они будут записаны одна за другой. Поэтому адрес начала каждой мелодии зависит от длины всех предыдущих. Удобнее всего просто по факту определить адрес начала каждой мелодии и поместить все семь адресов в специальную таблицу.

Кроме этой таблицы нам еще понадобится таблица коэффициентов деления для всех 32 нот и таблица, хранящая константы задержки для всех используемых нами музыкальных длительностей.

Алгоритм работы электромузыкального звонка (Рисунок 2.5):

1. Просканировать и определить номер нажатой кнопки.

2. Извлечь из таблицы начал мелодий значение элемента, номер которого соответствует только что определенному номеру нажатой кнопки. Это значение будет равно адресу в программной памяти, где начинается нужная нам мелодия.

3. Начать цикл воспроизведения мелодии. Для этого поочередно извлекать коды нот из памяти, начиная с адреса, который мы определили в пункте 2 алгоритма.

4. Каждый код ноты разложить на код тона и код длительности.

5. Если код тона равен нулю, отключить звук и перейти к формированию задержки (к п. 9 настоящего алгоритма).

6. Если код тона не равен нулю, извлечь из таблицы коэффициентов деления значение элемента с номером, равным коду тона.

7. Записать коэффициент деления, который мы нашли в пункте 6 настоящего алгоритма, в регистр совпадения таймера Т1.

8. Включить звук (подключить вывод ОС1А к выходу таймера Т1).

9. Извлечь из таблицы длительностей задержки значение элемента с номером, равным коду длительности.

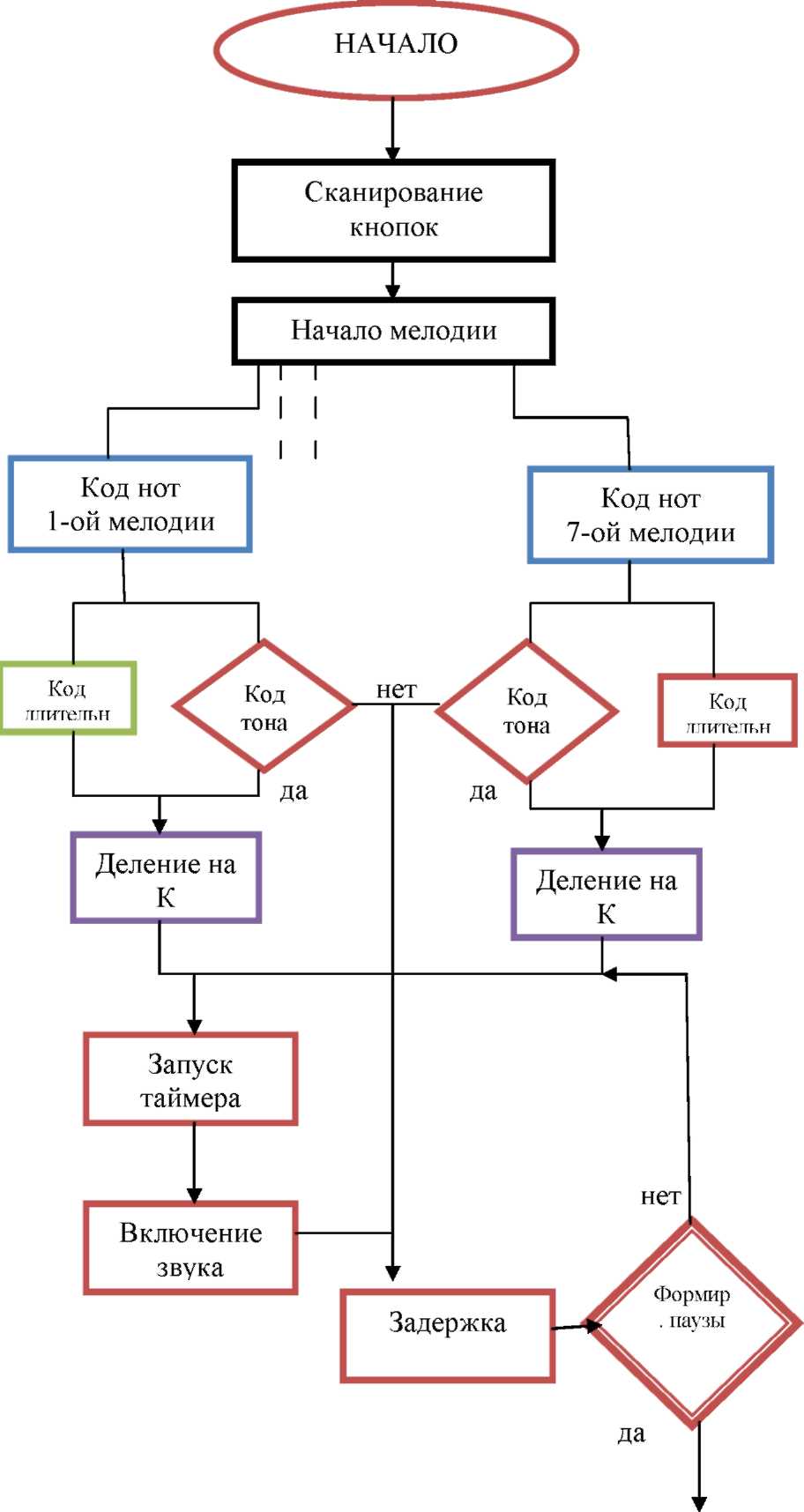
10. Сформировать паузу с использованием константы задержки, которую мы нашли в пункте 9 настоящего алгоритма.

11. По окончании паузы выключить звук (отключить ОС1А от выхода таймера).

12. Повторять цикл (пункты 4—11 настоящего алгоритма) до тех пор, пока нажата соответствующая кнопка.

13. Если очередной код ноты окажется равным 255, перейти к началу текущей мелодии, то есть вернуться к п. 3 настоящего алгоритма

Рисунок 2.5 – Алгоритм прогаммы для электромузыкального звонка



Выход

2.6 Разработка программного обеспечения микроконтроллера

Возможный вариант программы на языке Ассемблер приведен в листинге 1 (См. Приложение А).

2.6.1 Описание программы

Описание программы удобнее начать с конца. Начиная со строки 136 программы располагается описание так называемых таблиц данных. На самом деле каждая из этих "таблиц" представляет собой цепочку кодов, записываемых в программную память микроконтроллера и предназначенных для кодирования того либо иного вида данных. Для описания этих данных используются как операторы db, так и операторы dw.

Первая таблица содержит коэффициенты задержки для формирования всех вариантов музыкальной длительности. Таблица начинается с адреса, соответствующего метке tabz. Вся таблица занимает одну строкупрограммы (строка 136).Так как в нашей программе мы будем применять лишь семь вариантов длительности, таблица имеет **7** элементов. Каждый элемент записывается в память как двухбайтовое слово.

В строках 137—140описывается таблица коэффициентов деления для всех нот. Начало таблицы соответствует метке tabkd. Каждый элемент этой таблицы также имеет размер в два байта. Первый элемент таблицы равен нулю. Это неиспользуемый элемент. Ноты номер ноль у нас не существует. Ноль мы использовали для кодирования паузы.

В паузе не формируется звуковой сигнал, поэтому и коэффициент деления там не имеет смысла. Поэтому значение нулевого элемента массива несущественно. Описание таблицы разбито на строки. Для удобства каждая строка описывает коэффициенты деления для одной октавы. Нулевая нота выделена в отдельную строку. Последняя октава неполная, так как наш электромузыкальный звонок будет использовать всего 32 ноты.

В строках 143—200описана таблица мелодий.Вернее, это не одна таблица, а семь таблиц (своя таблица для каждой из мелодий). Каждая таблица помечена своей отдельной меткой (те 11, те 12 — те 17). Значение каждой метки — это адрес начала соответствующей мелодии. Каждое значение таблицы мелодий записывается в память в виде одного байта. Поэтому все строки, кроме последней, для каждой таблицы имеют четное число значений.

В строках 141,142 описана таблица начал всех мелодий.Начало этой таблицы отмечено меткой tabm. Таблица используется для того, чтобы программа могла найти адрес начала нужной мелодии по ее номеру. В качестве элементов массива выступают удвоенные значения меток mell, mel2 —mel7. Применение удвоенных значений обусловлено необходимостью перевода адресов из основной адресации в альтернативную. При трансляции программы вместо меток в память будут записаны конкретные адреса.

2.6.2 Процедура вычисления адреса

Большое количество таблиц в нашей программе заставляет позаботиться о процедуре вычисления адреса.

Однотипные вычисления удобно оформить в виде подпрограммы. Эта подпрограмма занимает строки 78—84. Вызов подпрограммы производится по имени addw. Подпрограмма получает номер элемента таблицы и адрес ее начала. Номер элемента передается в подпрограмму при помощи регистра *YL*, а адрес — через регистровую пару Z.

Используя эти данные, подпрограмма вычисляет адрес нужного элемента.Для этого она сначала удваивает номер элемента (строка79). Затем дополняет полученное значение до шестнадцатиразрядного путем записи в *YH* нулевого байта (строка80). И, наконец, производит сложение двух шестнадцатиразрядных величин, находящихся к этому моменту в регистровых парах *Y* и *Z* (строки81, 82). Результат вычислений при этом попадает, в регистровую пару *Z*.

2.6.3 Текст программы

Теперь рассмотрим текст программы с самого начала. В строках 3...10расположен модуль описания переменных (рабочих регистров). В строках 13...31располагается модуль переопределения векторов прерываний, в строках 32...41— модуль команд инициализации.

2.6.4 Особенности программы

Процедура, расположенная в строках45—52 программы, сканирует клавиатуру и находит код первой из нажатых кнопок. Найденый код находится в регистре count. Затем управление переходит к строке 53. С этого места начинается процедура выбора мелодии (строки53—58). Суть процедуры — прочитать из таблицы tabm значение адреса начала этой мелодии. То есть прочитать элемент таблицы, номер которого равен коду нажатой кнопки.

Прежде чем прочитать элемент, необходимо найти его адрес**.** Для вычисления адреса используем подпрограмму addw.Перед тем, как вызвать подпрограмму, подготовим все данные. Номер нажатой кнопки помещаем в регистр *YL* (строка 53).Адрес начала таблицы записываем в регистровую пару *Z* (строки 54, 55). И лишь затем в строке 56вызывается подпрограмма addw.

После выхода из подпрограммы в регистровой nape *Z* находится результат вычислений — адрес нужного нам элемента таблицы tabm.Следующие две команды (строки 57 и 58)извлекают тот элемент (адрес начала мелодии) и помещают его в регистровую пару *X*. Там этот адрес будет храниться все время, пока воспроизводится именно эта мелодия.

Следующий этап— воспроизведение мелодии. Воспроизведением мелодии занимается процедура, расположенная в строках 59—77.Для последовательного воспроизведения нот нам понадобится указатель текущей ноты. В качестве указателя текущей ноты используется регистровая пара *Z*. В самом начале процедуры воспроизведения мелодии в регистровую пару *Z* помещается адрес начала мелодии их регистровой пары *X* (строки 59, 60).

Затем начинается цикл воспроизведения (строки 61—77).В этом цикле программа извлекает код ноты по адресу, на который указывает наш указатель, выделяет из кода ноты код тона и код длительности, воспроизводит ноту, а затем увеличивает значение указателя на единицу. Затем весь цикл повторяется.

Этот процесс происходит до тех пор, пока код очередной ноты не окажется равным 255 (метка конца мелодии). Прочитав этот код, программа передает управление на строку 62,где в регистр *Z* снова записывается адрес начала мелодии. Воспроизведение мелодии начнется сначала. Этот процесс должен прерваться лишь в одном случае — при отпускании управляющей кнопки *S8*.

Для проверки состояния кнопок в цикл воспроизведения мелодии включена специальная процедура (строки 61—63).Процедура упрощенно проверяет состояние сразу всех кнопок. Она считывает содержимое порта *PD* (строка 61)и сравнивает его с кодом *0* x *7F* (строка 62).Прочитанное из порта значение может быть равно *0*x*7F* только в одном случае — если все кнопки отпущенны. Если хотя бы одна кнопка нажата, то при чтении порта мы получим другое значение.

Проверкой вышеописанного условия занимается оператор breq в строке 63. Если все кнопки оказались отпущены, этот оператор завершает цикл воспроизведения мелодии и передает управление на метку ml, то есть на самое начало основного цикла программы. Там происходит выключение звука, а затем новое сканирование клавиатуры.

Если хотя бы одна кнопка окажется нажатой, то цикл воспроизведения звука продолжается дальше, и управление переходит к строке 64, где происходит извлечение кода ноты. Так как адрес этой ноты находится в регистровой паре *Z* (указатель текущей ноты), то для извлечения ноты просто используется команда 1pm.

В строке 65 происходит проверка признака конца мелодии. Только что прочитанный код ноты сравнивается с кодом *0*x*FF*. Оператор breq в строке66 передает управление по метке т4,если мелодия действительно закончилась (условие выполняется). Если код ноты не равен *0*x*FF*, перехода не происходит, и управление переходит к строке 67.

В строках 67—75происходит обработка кода ноты. То есть из кода ноты выделяется код тона и код длительности. Сначала на код ноты накладывается маска, которая оставляет пять младших разрядов, а три старших сбрасывает (строка 67). Под действием маски в регистре temp остается код тона, который затем помещается в регистр fnotа (строка68).

Теперь нам нужно найти код длительности ноты.Для этого нам заново придется извлечь код ноты из памяти программ. Так как до этого момента мы не изменяли положение указателя текущей ноты, то для извлечения нет никаких препятствий. В строке69 мы повторно извлекаем код ноты из памяти программ. Но на этот раз значение указателя увеличивается. Теперь можно приступать к выделению кода длительности. Длительность кодируется тремя младшими битами кода ноты. Для выделения этих битов нам также нужно использовать маску.Но одной маской нам не обойтись. Нам нужно не просто выделить три старших разряда, а сделать их младшими, как это показано на Рисунке 2.6.

Процедура выделения кода длительности занимает строки 70—74.

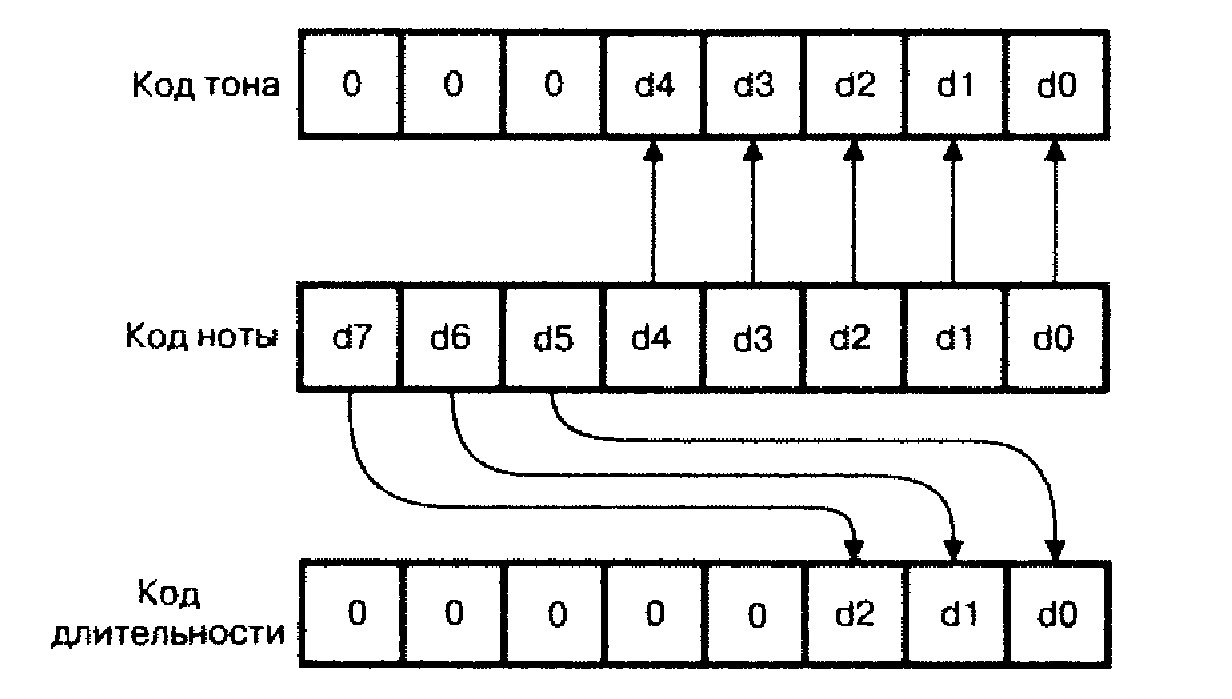


Рисунок 2.6 - Разложение кода ноты

Сначала программа производит многократный циклический сдвиг кода ноты до тех пор, пока три старших разряда не станут тремя младшими. Для сдвига используется команда *rо* 1. Так как сдвиг происходит через ячейку признака переноса, то нам понадобится четыре команды сдвига. Эти команды занимают в программе строки 70—73.

Затем в строке 74на полученное в результате сдвигов число накладывается маска, которая выделяет три младшие бита, а пять старших сбрасывает в ноль. Полученный таким образом код длительности записывается в регистр dnota (строка 75).

Когда код тонаи код длительности определены, производится вызов подпрограммы воспроизведения ноты (строка 76).Оператор *rjmp* в строке 77передает управление на начало цикла воспроизведения мелодии, и цикл повторяется для следующей ноты.

Подпрограмма воспроизведения ноты занимает строки 85—110.Она выполняет следующие действия:

- извлекает из таблицы tabkd коэффициент деления, соответствующий коду ноты;

- программирует таймер и включает звук;

- затем выдерживает паузу и звук выключает.

Если код тона равен нулю (нужно воспроизвести паузу без звука), извлечение коэффициента деления и включение звука не выполняется. Подпрограмма сразу переходит к формированию паузы.

Начинается подпрограмма воспроизведения ноты с сохранения всех используемых регистров (строки 85—88).Затем производится проверка кода ноты на равенство нулю (строка 89).Если код ноты равен нулю, то оператор breq в строке 90передает управление по метке ntl, то есть к строке, где происходит вызов процедуры формирования задержки.

Если код ноты не равен нулю, то программа приступает к извлечению коэффициента деления. Для вычисления адреса элемента таблицы tabkd, где находится этот коэффициент, снова используется подпрограмма addw.

Код тона помещается в регистр *YL* (строка91), а адрес начала таблицы — в регистровую пару *Z* (строки 92, 93).Вызов подпрограммы addw производится в строке 94. В регистровой паре *Z* подпрограмма возвращает адрес элемента таблицы, где находится нужный нам коэффициент деления. В строках 95, 96из таблицы извлекается этот коэффициент. А в строках 97,98он помещается в регистр совпадения таймера. В строках 99,100включается звук.

В строке 104вызывается специальная подпрограмма, предназначенная для формирования задержки. Подпрограмма называется wait и формирует задержку с переменной длительностью. Длительность задержки зависит от значения регистра dnota. По окончании задержки звук выключается (строки 102,103).

На этом можно было бы закончить процесс воспроизведения ноты. Однако это еще не все. Для правильного звучания мелодии между двумя соседними нотами необходимо обеспечить хотя бы небольшую паузу.Если такой паузы не будет, ноты будут звучать слитно. Это исказит мелодию, особенно если подряд идет несколько нот с одинаковым тоном. Формирование паузы между нотами происходит в строках 104,105.

Вспомогательная пауза формируется при помощи уже знакомой нам подпрограммы задержки. В строке 104коду паузы присваивается нулевое значение (выбирается самая минимальная пауза). Затем в строке 105вызывается подпрограмма wait. После окончания паузы остается только восстановить содержимое всех сохраненных регистров из стека (строки 106—109)и выйти из подпрограммы (строка 110).

2.6.5 Подпрограмма формирования задержки

И последнее, что нам еще осталось рассмотреть, — это подпрограмма формирования задержки. Текст подпрограммы занимает строки 111—135.Как и любая другая подпрограмма, подпрограмма wait в начале сохраняет (строки 111—114)**,** а в конце — восстанавливает (строки 131—134)все используемые регистры.

Рассмотрим, как работает эта подпрограмма. Сначала определяется длительность задержки. Для этого извлекается соответствующий элемент из таблицы tabz.Номер элемента соответствует коду задержки, находящемуся в регистре dnota.Извлечение значения из таблицы производится уже знакомым нам образом. Команды, реализующие вычисление адреса нужного элемента таблицы, находятся в строках 115—118.Затем в строках 119 и 120производится чтение элемента таблицы. Прочитанный код задержки помещается в регистровую пару *Y*.

Теперь наша задача:сформировать задержку, пропорциональную содержимому регистровой пары *Y.* Так как микроконтроллер ATtiny2313 имеет только один шестнадцатиразрядный таймер, который уже занят формированием звука, будем формировать задержку программным путем. Но в данном случае цикл формирования задержки построен немного по-другому.

Вообще-то, способов построения подобных подпрограмм может быть бесконечное множество. Все зависит от изобретательности. Использованный в данном примере способ более удобен для формирования задержки переменной длительности, пропорциональной заданному коэффициенту. Главной особенностью нового способа является шестнадцатиразрядный параметр цикла.

Для хранения этого параметра используется регистровая пара *Z*. Перед началом цикла задержки в нее записывается ноль. Затем начинается цикл, на каждом проходе которого содержимое регистровой пары *Z* увеличивается на единицу. После каждого такого увеличения производится сравнение нового значения *Z* с содержимым регистровой пары *Y*.

Заканчивается цикл тогда, когда содержимое *Z* и содержимое *Y* окажутся равны. В результате число, записанное в регистровой паре *Y*, будет определять количество проходов цикла. Поэтому и время задержки, формируемое этим циклом, будет пропорционально константе задержки. Однако это время будет слишком мало для получения приемлемого темпа воспроизведения мелодий. Для того, чтобы увеличить время до нужной нам величины, внутрь главного цикла задержки помещен еще один цикл, имеющий фиксированное количество проходов.

Описанная выше процедура задержки занимает строки121—135. В строках 121, 122 производится запись нулевого значения в регистровую пару *Z*. Большой цикл задержки занимает строки123—130. Малый внутренний цикл занимает строки124—125. Для хранения параметра малого цикла используется регистр loop.В строке 123 в него записывается начальное значение. Строки124,125 выполняются до тех пор, пока содержимое loopне окажется равным нулю.

В строке 126 содержимое регистровой пары *Z* увеличивается на единицу. В строках 127—130 производится сравнение содержимого двух регистровых пар *Y* и *Z*. Сравнение производится побайтно. Сначала сравниваются младшие байты (строка 127). Если они не равны, оператор условного перехода в строке 128 передает управление на начало цикла.

Если младшие байты равны, сравниваются старшие байты (строка 129). Если старшие байты неодинаковы, оператор *brne* в строке 130 опять заставляет цикл начинаться с начала. И только когда оба оператора сравнения дадут положительный результат (не вызовут перехода), цикл заканчивается, и подпрограмма формирования задержки переходит к завершающей фазе (к строкам 131—135).

2.6.6 Программа на языке СИ

Возможный вариант программы на языке СИ приведен в листинге2. В данном случае использована модификация языка поддерживаемая программной средой CodeVision. Описание программы рассчитано на программистов, знакомых с языком СИ.

Теперь рассмотрим подробнее программу с самого начала (Листинг 2, Приложение Б).

2.6.7 Описание программы (листинг 2)

Для формирования задержки мы будем использовать функцию из библиотеки delay.h. Поэтому в строках 1,2 программы, кроме файла описаний, мы присоединяем и эту библиотеку. Затем наминаются описания всех массивов. В строке 3 описывается массив, содержащий величины всех музыкальных длительностей.

Так как для формирования длительности мы будем использовать функцию delay\_ms, величина длительностей задана в миллисекундах. Как видно из текста программы, в данном случае мы используем массив типа unsigned int. Переменные этого типа имеют длину два байта, все 16 битов которых используются для хранения информации.

Именно такой тип наиболее подходит для хранения наших коэффициентов. Управляющее слово fleash перед описанием массива гарантирует, что эти данные будут размещены в программной памяти микроконтроллера.

В строках 4, 5, 6 описывается массив коэффициентов деления для всех нот. В этом месте программы мы впервые используем перенос строки. Перенос строки применяется в том случае, когда текст команды не помещается в одной строке. Язык СИ разрешает свободно переносить текст на следующую строку. При этом не требуется никаких специальных директив и указателей.

Перенос допускается в том месте команды, где между двумя соседними элементами выражения можно поставить пробел. Тип массива, как и в предыдущем случае,— usingnerd int. Содержимое массива tabkd полностью соответствует содержимому таблицы с тем же названием из ассемблерного варианта программы.

В строках 7—38 описываются семь массивов для хранения семи мелодий. Массивы имеют тип unsigned char. Переменные этого типа занимают в памяти один байт, и все восемь битов этого байта используются для хранения информации. Содержимое каждого из этих массивов полностью соответствует содержимому соответствующих таблиц в ассемблерной версии программы.

В строке 39 описывается массив, содержащий адрес начала каждой из семи мелодий. Это не просто массив, а массив ссылок, на что указывает символ звездочки в тексте его описания. Так же, как и ссылочная переменная, каждый элемент массива ссылок предназначен для хранения ссылки. Данный массив тоже хранится в памяти программ, на что указывает управляющее слово flesh в его описании. Элементы этого массива хранят указатели на начало каждого из массивов мелодий, что указано при его инициализации (в фигурных скобках).

Строки 40—72 занимает функция main. Начинается функция с описания переменных (строки 41—45). Две рабочих переменных count и temp, а также переменная для хранения кода тона (tnota) и переменная для хранения кода длительности (dnota) нам уже знакомы. Мы использовали их в предыдущей программе.

Интерес представляет описание переменной notа. Это ссылочная переменная, которая предназначена для хранения указателей на объекты в программной памяти, имеющие тип unsigned char. Она будет использоваться нами для обращения к элементам массивов, хранящим коды нот. Эти массивы, как уже говорилось, расположены в программной памяти. Поэтому в описании переменной имеется слово flash, а перед именем переменной в ее описании стоит символ звездочки. То есть это ссылка на массивы типа unsigned char, расположенные во flesh.

В строках 46—52 расположен блок инициализации. Эта часть программы полностью повторяет аналогичную часть программы из предыдущего примера (см. листинг 2).

Строки 53—72 занимает основной цикл программы. Цикл состоит всего из двух процедур. В начале цикла (строки 54—59) расположена процедура сканирования кнопок. Эта процедура один к одному скопирована из предыдущего примера (см. листинг 2 строки 14—21).

При обнаружении нажатой кнопки управление передается по метке *m3* (в новой программе это строка 60). Как вы помните, номер нажатой кнопки при выходе из процедуры сканирования содержится в переменной count.

Строки 60—72 занимает процедура проигрывания мелодии.

Проигрывание начинается с того, что в переменную nota помещается указатель на массив, содержащий нужную нам мелодию (строка 60). А указатель — это элемент массива tabm, с номером, равным коду нажатой кнопки. В строках 61—72 находится цикл, который последовательно считывает мелодию нота за нотой и проигрывает прочитанные ноты. Цикл организован при помощи оператора безусловного перехода (строка 72).

Для перемещения вдоль массива содержимое переменной nota каждый раз увеличивается на единицу (строка 71). В этом же цикле производится проверка состояния кнопки (нажата ли еще хоть одна кнопка) и проверка признака конца мелодии. Рассмотрим подробнее, как все это делается.

Проверка состояния кнопок происходит в строке 61. Если содержимое регистра *PIND* равно *0*х*7F*, то воспроизведение мелодии прекращается. Управление передается по метке *m2*. Там происходит выключение звука, а затем переход по метке *m1*, то есть к началу основного цикла программы.

Если хоть одна кнопка еще нажата, перехода не происходит и воспроизведение мелодии продолжается. В строке 62 производится проверка на конец мелодии. Содержимое элемента массива, на который указывает ссылочная переменная nota (код ноты), проверяется на равенство числу *0*x*FF*. Если код ноты равен *0*x*FF*, то управление передается по метке *m3*, где указатель снова устанавливается на начало мелодии.

В строке 63 вычисляется значение кода тона. Для этого на код ноты, на который указывает переменная notа, накладывается маска. Наложение маски производится при помощи оператора «&». Полученный код тона записывается в переменную fnota.

В строке 64 производится вычисление кода длительности. Для этого применяется составное математическое выражение. Операция (\*nota) *>>5* сдвигает биты кода ноты на пять шагов вправо. При этом три старших разряда кода становятся тремя младшими. Мы применяем сдвиг вправо потому, что циклический сдвиг влево, использованный нами в Ассемблере, язык СИ не поддерживает. Язык СИ может выполнять только логический сдвиг, но не циклический. На полученное в результате сдвига число налагается маска *0*x*07*. Полученный таким образом код длительности записывается в переменную dnota.

В строке 65 происходит проверка кода тона на равенство нулю.

Если код окажется равным нулю, то управление передается по метке *m5*, то есть к строке, где формируется пауза, обходя строки, где формируется звук.

Звук формируется в строках 66, 67. Сначала в регистр совпадения *OCR1A* помещается коэффициент деления из массива tabkd. Причем указатель массива равен коду тона. Затем в регистр управления TCCR1A записывается код, который подключает таймер к выводу *ОСІА* и, тем самым, включает звук.

В строке 68 происходит вызов функции задержки. В качестве параметра в эту функцию передается коэффициент, извлекаемый из массива tabz. Указатель массива при этом равен коду длительности. После выхода из функции задержки звук выключается.

Для этого в регистр *TCCR1А* записывается нулевое значение (строка 69). В строке 70 формируется пауза между нотами. В качестве параметра для функции delay\_ms в этом случае используется нулевой элемент массива tabz, то есть вырабатывается пауза минимальной длительности.

В строке 71, как уже говорилось, происходит приращение содержимого указателя nota. Оператор безусловного перехода в строке 72 замыкает цикл воспроизведения мелодии.

2.7 Выбор, описание и расчеты элементной базы

В электромузыкальном звонке с двумя режимами работы постараемся использовать широко, надежные и дешевые элементы.

Устройство проигрывания мелодий содержит микроконтроллер ATtiny 2313, кварцевый резонатор на 4 МГц, широкоиспользуемый кнопочный переключатель П2К на семь кнопок, дверная звонковая кнопка, любой маломощный транзистор обратной проводимости, любой громкоговоритель 0,1-0,5 Вт с сопротивление катушки 8 Ом.

Источником питания служит бестрансформаторный источник питания (для уменьшения размеров), при работе от сети переменного напряжения 220В и четыре пальчиковых элемента питания, при работе устройства в автономном режиме (Приложение В).

В точки А и В подключается сеть переменного напряжения 220 В. Однополупериодный выпрямитель с ограничением по току питает схему мощного стабилитрона VD2 и стабилизатора напряжения D2. Цепь понижения напряжения сети до уровня 9В (R2,C3) рассчитана на ток потребления 40 мА. Ток стабилизации стабилитрона составляет 20 мА – столько же потребляет стабилизатор напряжения D2, питающий микроконтроллер и электронный ключ во время работы устройства.

В момент включения основной ток потребляет конденсатор С5, но этот ток ограничен реактивным сопротивлением С5, также выполняет функцию источника тока в момент отключения питания сети. При включении сети зажигается светодиод LED 1, питание которого осуществляется через гасящий резистор R5. Конденсатор C4 сглаживает высокочастотные импульсные помехи, проходящие из сети в схему питания. Резисторы R3 и R4 составляют делитель напряжения на стабилитроне VD2 (необходим для измерительного канала микроконтроллера). Измерительный канал не содержит схему выборки и хранения, а также интегратора и фильтра верхних частот, поскольку эти функции выполняют фильтрующие элементы питания С4, С5. Функцию интегратора выполняет ограничительный стабилитрон VD2. При переходе устройства в автономный режим, питание осуществляется от пальчиковых элементов постоянного тока G1-G2. Потребление от элементов питания составляет порядка 15 мА. Стабилизатор напряжения 78LO5Z стабилизирует напряжение до 5 В.

Бестрансформаторная схема питания была предложена в связи с ограничением места в корпусе устройства. Единственное условие, которое следует выполнять – клемму «В» следует подключить к «нулевому» проводу сети, а «фазу» к клемме «А». Резисторы R2, R6 выбираем мощностью 1 Вт, чтобы предохранить устройство от перегрева.

При выключении сетевого напряжения, светодиод VED1 гаснет, т.к. попадание напряжения питания элементов питания G1-G2 ограничивает диод VD3. При включении сети происходит подзаряд элементов питания. Поэтому рекомендуется использовать пальчиковые аккумуляторы.

Конденсатор С3 на напряжение не ниже 400 В. Светодиод можно использовать любой. Электролитические конденсаторы на напряжение: С5 – не ниже 30В, С6 – не ниже 10В. Стабилитрон любой на напряжение стабилизации 9 В.

Нам удалось за счет использования нового способа формирования задержки, использовать специализированную программу (Раздел 2.6.7) при этом не увеличивая число элементов схемы, чем смогли уменьшить ток потребления схемы.

2.8 Разработка схемы принципиальной

Принципиальная схема электромузыкального звонка выполнена в САПР *Accel Eda* (Рисунок 2.7).

Проект основывается на микроконтроллере ATtiny 2313. Устpойство состоит из 8 кнопок, частотозадающих элементов. Отсчет времени звучания ведется с помощью таймера TMR0. Во время работы постоянно сканируется состояние порта В, и если оно изменилось (какая-либо кнопка нажата или отпущена), то в соответствии с новой комбинации клавиш меняется и частота звучания. При нажатии кнопки S8, устройство переходит в режим воспроизведения и проигрывает записанную мелодию. В качестве динамика можно использовать 0.25ГД-19 8 Ом.

Микроконтроллер имеет встроенный детектор включения питания. Таймер запуска начинает счет выдержки времени после того, как напряжение питания пересекло уровень около 1,2...1,8 Вольт. По истечении выдержки около 72мс считается, что напряжение достигло номинала и запускается другой таймер-выдержка на стабилизацию кварцевого генератора. Программируемый бит конфигурации позволяет разрешать или запрещать выдержку от встроенного таймера запуска. Выдержка запуска меняется от экземпляров кристалла, от питания и температуры.

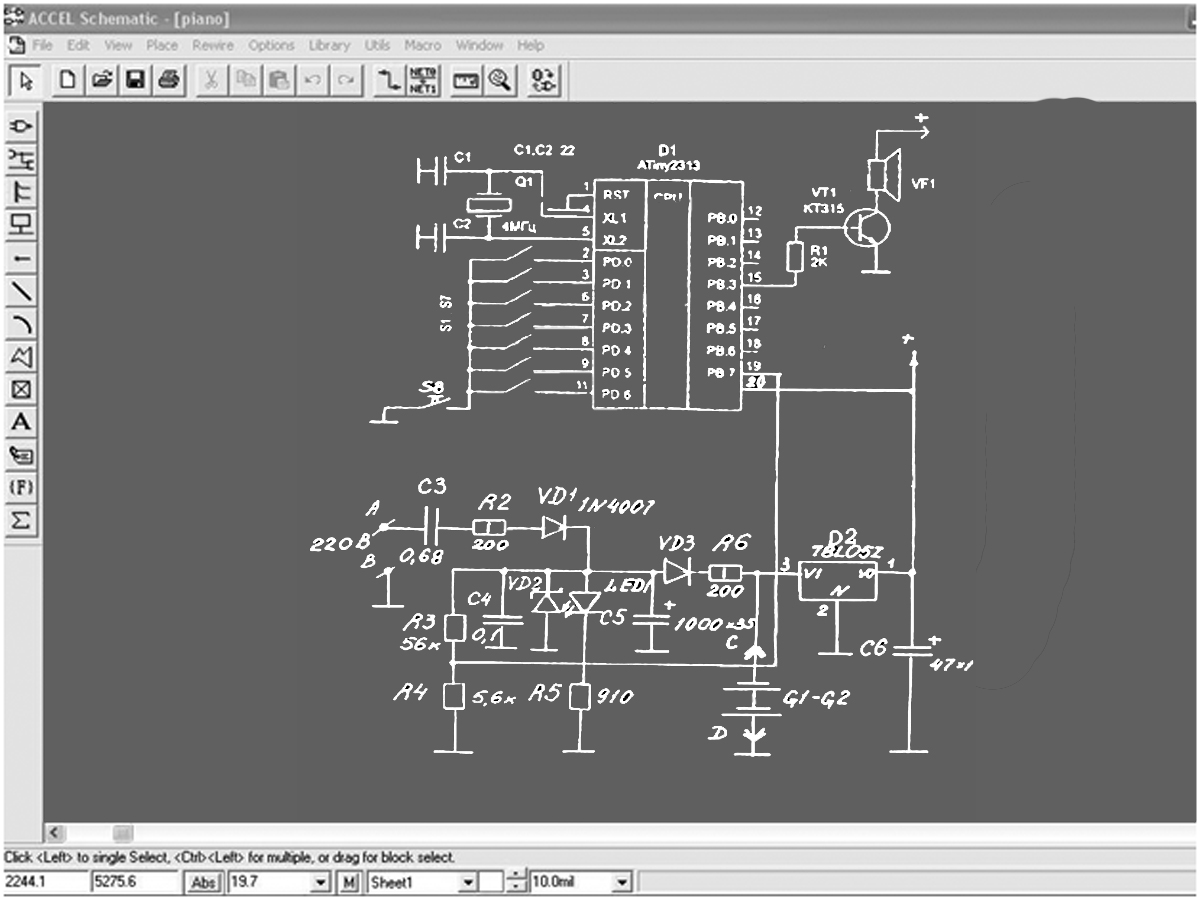


Рисунок 2.7 - Принципиальная схема электромузыкального звонка в *Accel EDA*

Таймер на стабилизацию генератора отсчитывает 1024 импульса от начавшего работу генератора. Считается, что кварцевый генератор за это время вышел на режим. При использовании RC генераторов выдержка на стабилизацию не производится.

Затем включается таймер ожидания внешнего сброса /MCLR. Это необходимо для тех случаев, когда требуется синхронно запустить в работу несколько PIC контроллеров через общий для всех сигнал /MCLR.   
Если такого сигнала не поступает, то через время Tost вырабатывется внутренний сигнал сброса и контроллер начинает ход по программе.

Принципиальная электрическая схема устройства, удовлетворяющая сформулированным выше требованиям приведена в Приложении В.Кнопки S1...S7 предназначены для выбора мелодий. Кнопка S8 производит запуск электромузыкального звонка. Для воспроизведения мелодии используется звуковой излучатель VF1, сигнал на который поступает с выхода РВ.3 микроконтроллера. В качестве усилителя сигнала используется электронный ключ R1, VT1.

Спецификация элементов приведена в Приложении Д.

3 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ОБЪЕКТА

РАЗРАБОТКИ

В данном разделе проводится технико-экономический расчет стоимости музыкального звонка с двумя режимами работы.

Стоимость устройства будет состоять из стоимости разработки ПО для микроконтроллера, стоимости разработки конструкторской документации и стоимости сборки и испытания устройства.

3.1 Расчет расходов на ПО для микроконтроллера, которое разрабатывается

Исходные данные для расчета стоимости разработки ПО, которое разрабатывается приведнны в таблице 3.1.

Таблица 3.1 – Исходные данные по предприятию

| №п/п | Статьи затрат | Усл. обоз. | Ед. изм. | Значения |
| --- | --- | --- | --- | --- |
|  |
| Проектирование и разработка ПО | | | | |
| 1 | Часовая тарифная ставка программиста | *Зпр* | грн. | 8,00 |
| 2 | Коэффициент сложности программы | *с* | коэф. | 1,40 |
| 3 | Коэффициент коррекции программы | *Р* | коэф. | 0,05 |
| 4 | Коэффициент увеличения расходов труда | *Z* | коэф. | 1,3 |
| 5 | Коэффициент квалификации программиста | *k* | коэф. | 1,0 |
| 6 | Амортизационные отчисления | *Амт* | % | 10,0 |
| 7 | Мощность компьютера, принтера | *WМ* | Квт/ч | 0,40 |
| 8 | Стоимость ПЕОМ IBM  Sempron LE1150(AM2)/1GB/TFT | *Втз* | грн. | 3200,00 | |
| 9 | Тариф на электроэнергию | *Це/е* | грн. | 0,56 | |
| 10 | Норма дополнительной зарплаты | *Нд* | % | 10,0 | |
| 11 | Отчисление на социальные расходы | *Нсоц* | % | 37,2 | |
| 12 | Транспортно-заготовительные расходы | *Нтр* | % | 4,0 | |
| Эксплуатация П0 | | | | |
| 13 | Численность обслуживающего персонала | *Чо* | чел | 1 |
| 14 | Часовая тарифная ставка обслуживающего персонала | *Зпер* | грн. | 6,00 | |
| 15 | Время обслуживания систем | *То* | час/г | 150 | |
| 16 | Стоимость ПЕОМ | *Втз* | грн. | 3200,00 | |
| 17 | Норма амортизационных отчислений на ПЕОМ | *На* | % | 10,0 | |
| 18 | Норма амортизационных отчислений на ПЗ | *НаПО* | % | 10,0 | |
| 19 | Накладные расходы | *Рнак* | % | 25,0 | |
| 20 | Отчисление на содержание и ремонт ПЕОМ | *Нр* | % | 10,0 | |
| 21 | Стоимость работы одного часа ПЕОМ | *Вг* | грн. | 6,5 | |

Первичными исходными данными для определения себестоимости ПО является количество исходных команд (операторов) конечного программного продукта. Условное количество операторов *Q* в программе задания может быть оценено по формуле:

|  |  |
| --- | --- |
| , | (3.1) |

где  *у* – расчетное количество операторов в программе, что разрабатывается (единиц);

*с* – коэффициент сложности программы;

*р* – коэффициент коррекции программы в ходе ее разработки.

Рассчитанное количество операторов в разработанной программе – 500.

Коэффициент *с* – относительная сложность задания относительно отношения к типичной задаче, сложность которой принята более 1, лежит в границах от 1,25 до 2,0 и выбирается равным 1,30.

Коэффициент коррекции программы *р* – увеличение объема работ за счет внесения изменений в программу лежит в границах от 0,05 до 0,1 и выбирается равным 0,05.

Подставим выбранные значения в формулу (3.1) и определим величину *Q*:

*Q* = 200∙1,3 (1 + 0,05) = 273.

3.2 Расчет расходов на создание ПО

Расчет расходов на ПО проводится методом калькуляции расходов, в основу которого положена трудоемкость и заработная плата разработчиков. Трудоемкость разработки ПО рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (3.2) |

где *То* – расходы труда на описание задания;

*Ти* – расходы труда на изучение описания задания;

*Та* – расходы труда на разработку алгоритма решения задания;

*Тп* – расходы труда на составление программы по готовой блок-схеме;

*Тотл* – расходы труда на отладку программы на ЭВМ;

*Тд* – расходы труда на подготовку документации.

Составные расходы труда, в свою очередь, можно определить по числу операторов *Q* для ПО, что разрабатывается. При оценке составных расходов труда используются:

- коэффициенты квалификации разработчика алгоритмов и программ – *k*;

– увеличение расходов труда в результате недостаточного описания задания – *Z*.

Коэффициент квалификации разработчика характеризует меру подготовленности исполнителя к порученной ему работе (он задается в зависимости от стажа работы), *k* = 1,0.

Коэффициент увеличения расходов труда в результате недостаточного описания задания характеризует качество постановки задания, выданной для разработки программы, в связи с тем, что задание требовало уточнения и некоторой доработки. Этот коэффициент принимается равным 1,3.

Все исходные данные приведенные в таблице 3.1.

а) Трудоемкость разработки П0 составляет:

Расходы труда на подготовку описания задания *То* принимаются равными 5 чел/час, исходя из опыта работы.

Расходы труда на изучение описания задания *Те*с учетом уточнения описания и квалификации программиста могут быть определены по формуле:

|  |  |
| --- | --- |
| ; | (3.3) |
| *Ти* = 273∙1,3/80∙1 = 5(чел/час) |  |

Расходы труда на разработку алгоритма решения задачи рассчитываются по формуле:

|  |  |
| --- | --- |
| ; | (3.4) |
| *Та=*273/25∙1 = 11(чел/час) |  |

Расходы труда на составление программы по готовой блок-схеме *Тп* рассчитываются по формуле:

|  |  |
| --- | --- |
| ; | (3.5) |
| (чел/час) |  |

Расходы труда на отладку программы на ПЕОМ *Тотл* рассчитываются по формуле:

* при автономной отладке одного задания:

|  |  |
| --- | --- |
| ; | (3.6) |
| (чел/час) |  |

* при комплексной отладке задания:

|  |  |
| --- | --- |
| ; | (3.7) |
| (чел/час) |  |

Расходы труда на подготовку документации по заданию *Тд* определяются по формуле:

|  |  |
| --- | --- |
| , | (3.8) |

где *Тдр* – расходы труда на подготовку материалов в рукописи:

|  |  |
| --- | --- |
| ; | (3.9) |
| (чел/час) |  |

*Тдо* – расходы труда на редактирование, печать и оформление документация:

|  |  |
| --- | --- |
| . | (3.10) |
| (чел/час) |  |

Подставляя приобретенных значений в формулу (3.8), получим:

|  |  |
| --- | --- |
| (чел/час) |  |

Определим трудоемкость разработки ПО, подставив полученные значения составляющих в формулу (3.2):

|  |  |
| --- | --- |
|  |  |

Расчет трудоемкости и зарплаты приведен в таблице 3.2.

б) Расчет материальных расходов на разработку ПЗ

Материальные расходы *Мз*, которые необходимы для создания ПО приведенные в таблице 3.3.

Таблица 3.2 – Трудоемкость и зарплата разработчиков ПО

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов  разработки | Трудоемкость  чел/часов | Почасовая тарифная ставка программиста, грн. | Сумма зарплаты, грн. |
| Описание задания | 5 | 8,00 | 40,00 |
| Изучение задания | 5 | 8,00 | 40,00 |
| Составление алгоритма решения задачи | 11 | 8,00 | 88,00 |
| Программирование | 13 | 8,00 | 104,00 |
| Отладка программы | 55 | 8,00 | 440,00 |
| Оформление документации | 25 | 8,00 | 200,00 |
| ВСЕГО: | 114 | 8,00 | 912,00 |

Таблица 3.3 – Расчет материальных расходов на разработку ПО

|  |  |  |  |
| --- | --- | --- | --- |
| Материал | Фактическое количество | Цена за единицу, грн. | Сумма, грн. |
| 1. DVD | 2 | 3,00 | 6,00 |
| 2. Бумага | 500 | 0,10 | 50,00 |
| ВСЕГО: |  |  | 56,00 |
| ТЗР (4%) |  |  | 2,24 |
| ИТОГО: |  |  | 57,24 |

в) Расходы на использование ЭВМ при разработке ПО

Расходы на использование ЭВМ при разработке ПО рассчитываются, исходя расходов одного часа, по формуле:

|  |  |
| --- | --- |
| , | (3.12) |

где *Вг* – стоимость работы одного часа ЭВМ, грн.;

*Тотл* – расходы труда на наладку программы на ЭВМ, чел./час.;

*Тд*– расходы труда на подготовку документации, чел./час.;  
 *Тп* – расходы труда на составление программы по готовой блок-схеме, чел./час.

|  |  |
| --- | --- |
| (грн.) |  |

г) Расчет технологической себестоимости создания программы

Расчет технологической себестоимости создания программы проводится методом калькуляции расходов (таблица 3.4).

Таблица 3.4 – Калькуляция технологических расходов на создание ПО

|  |  |  |
| --- | --- | --- |
|  | Наименование | Расходы, грн. |
| 1 | Материальные расходы | 57,24 |
| 2 | Основная зарплата | 912,00 |
| 3 | Дополнительная зарплата (15,0 %) | 136,80 |
| 4 | Отчисление на социальные мероприятия (37,2 %) | 390,15 |
| 5 | Накладные расходы (25,0 %) | 228,00 |
| 6 | Расходы на использование ЭВМ  составлении программного обеспечения ПО | 604,50 |
| 7 | Себестоимость ПО микроконтроллера | 2328,69 |

В таблице 3.4 величина материальных расходов *Мз* рассчитана в таблице 3.3, основная зарплата *Со* берется из таблицы 3.2, дополнительная зарплата составляет 15% от основной зарплаты, отчисление на социальные потребности – 37,2% от основной и дополнительной зарплат (вместе), накладные расходы – 25% от основной зарплаты. Себестоимость разработанной программы *СПО* рассчитывается как сумма пунктов 1 – 6.

Стоимость ПО для микроконтроллера составляет 2328,69 грн. на единицу продукции. Если организовать массовый выпуск продукции эта стоимость разделится на количество выпущенных изделий.

3.3 Расчет стоимости разработки конструкторской документации и

сборки устройства

а) Трудоемкость разработки КД изделия (*Т*) рассчитывается по формуле:

|  |  |
| --- | --- |
| , | (3.13) |

где *Татз* – расходы труда на анализ технического задания (ТЗ), чел./час;

*Трес* – расходы труда на разработку электрических схем, чел./час;

*Трк* – расходы труда на разработку конструкции, чел./час;

*Трт* – расходы труда на разработку технологии, чел./час;

*Токд* – расходы труда на оформление КД, чел./час;

*Твидз* – расходы труда на изготовление и испытание опытного образца, чел./час.

Данные расчета заносятся в таблицу 3.5.

Таблица 3.5 - Расчет заработной платы на разработку КД изделия

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Виды работ | Условные обозначения | Почасовая тарифная ставка -  *Сст,* грн. | Факт.  расходы времени  чел./час; | Зарплата, грн. |
| 1. Анализ ТЗ | *Татз* | 6,00 | 2 | 12,00 |
| 2. Разработка электрических схем | *Трес* | 6,00 | 4 | 24,00 |
| 3. Разработка конструкции | *Трк* | 6,00 | 4 | 24,00 |
| 4. Разработка технологии | *Трт* | 6,00 | 2 | 12,00 |
| 5. Оформление КД | *Токд* | 6,00 | 2 | 12,00 |
| 6. Изготовление и испытание опытного образца | *Твидз* | 6,00 | 8 | 48,00 |
| Всего: | *∑* | 6,00 | 22 | 132,00 |

Заработная плата на разработку КД изделия определяется по формуле:

|  |  |
| --- | --- |
| , | (3.14) |

где - почасовая тарифная ставка разработчика, грн.;



- трудоемкость разработки КД изделия.



б) Расчет материальных расходов на разработку КД

Материальные расходы *Мв*, которые необходимы для разработки (создании) КД, приведены в таблице 3.6.

Таблица 3.6 - Расчет материальных расходов на разработку КД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Материал | Обозначение пометь. | Факт. кол.  чество | Цена за ед. грн.  цу, грн. | Сумма,  грн. |
| 1. CD DVD |  | 2 | 3,00 | 6,00 |
| 2. Бумага |  | 500 | 0,07 | 35,00 |
| ВСЕГО: | *∑* |  |  | 41,00 |
| ТЗР (4%) |  |  |  | 1,64 |
| Итого: | *Мв* |  |  | 42,64 |

в) Расходы на использование ЭВМ при разработке КД

Расходы, на использование ЭВМ при разработке КД, рассчитываются исходя из расходов работы одного часа ЭВМ по формуле, грн.:

|  |  |
| --- | --- |
| , | (3.15) |
|  |  |

где  *Вг* – стоимость работы одного часа ЭВМ, грн.

*Трес* – расходы труда на разработку электрических схем, чел./час;

*Трк* – расходы труда на разработку конструкции, чел./час;

*Трт* – расходы труда на разработку технологии, чел./час;

*Токд* – расходы труда на оформление КД, чел./час;

При этом, стоимость работы одного часа ЭВМ (других технических средств - ТС) *Вг* определяется по формуле, грн.:

|  |  |
| --- | --- |
| , | (3.16) |
|  |  |

где *Те/е* *–* расходы на электроэнергию, грн.;

*Ваморт* – величина 1-ого часа амортизации ЭВМ (ТС), грн.;

*Зперс*– почасовая зарплата обслуживающего персонала, грн.;

*Трем* *–* расходы на ремонт, покупку деталей, грн.;

Стоимость одного часа амортизации *Ваморт* определяется по формуле, грн.: (при 40 часовой рабочей неделе)

|  |  |
| --- | --- |
| , | (3.17) |

где *Втз* - стоимость технических средств, грн.

*На* - норма годовой амортизации (%).

*Кт* - количество недель на год (52 недели/год).

*Гт* - количество рабочих часов в неделю (40 час/неделя)

Почасовая оплата обслуживающего персонала *Зперс* рассчитывается по формуле, грн.:

|  |  |
| --- | --- |
| , | (3.18) |

где *Окл* - месячный оклад обслуживающего персонала, грн.

*Крг* - количество рабочих часов в месяц (160 часов/месяц);

*Нрем*- расходы на оплату труда ремонта ЭВМ (6 % *Окл*).

Расходы на ремонт, покупку деталей для ЭВМ *Трем* определяются по формуле, грн.:

|  |  |
| --- | --- |
| , | (3.19) |
|  |  |

где  *Втз* - стоимость технических средств, грн.

*Нрем* - процент расходов на ремонт, покупку деталей (%);

*Кт* - количество недель на год (52 недели/год).

*Гт* - количество рабочих часов в неделю (36 ÷ 168 час./неделя)

Расходы на использование электроэнергии ЭВМ и техническими средствами *Те/е* определяются по формуле, грн.:

|  |  |
| --- | --- |
| , | (3.20) |

где *Ве/е* – стоимость одного кВт/час электроэнергии, грн.;

*Wпот* – мощность компьютера, принтера и сканера (за 1 час), (кВт/час.).

Таким образом, стоимость одного часа работы ЭВМ при разработке КД будет составлять (см. формулу 3.16), грн.:

.



Расходы на использование ЭВМ при разработке, грн. (см. формулу 3.15):



г) Расчет технологической себестоимости создания КД

Расчет технологической себестоимости создания КД изделия проводится методом калькуляции расходов (таблица 3.7).

Таблица 3.7 - Калькуляция технологических расходов на создание КД

изделия

| №  п/п | Наименование статей | Условные  обозначения | Расходы (грн.) |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 1. | Материальные расходы | *Мв* | 42,64 |
| 2. | Основная зарплата | *Зо* | 132,00 |
| 3. | Дополнительная зарплата | *Зд* | 19,80 |
| 4. | Отчисление на социальные мероприятия | *37,2%(Зо+Зд)* | 56,47 |
| 5. | Общепроизводственные (накладные)  расходы предприятия | *Ннакл* | 33,00 |
| 6. | Расходы на использование ЭВМ при составлении программного обеспечения КД | *ВЕОМ* | 26,60 |
| 7. | Себестоимость КД изделия | *Скд= ∑ (1÷6)* | 310,51 |

В таблице 3.7 величина материальных расходов *Мв* рассчитана в таблице 3.6, основная зарплата *Со* берется из таблицы 3.5, дополнительная зарплата 15 % от основной зарплаты, отчисление на социальные мероприятия –37,2% - от основной и дополнительной зарплаты (вместе). Накладные расходы 25% от основной зарплаты. Себестоимость разработанной конструкторской документации *Скд* рассчитывается как сумма пунктов 1–6.

## 3.4 Расчет расходов на стадии производства изделия

Себестоимость изделия которое разрабатывается рассчитывается на основе норм материальных и трудовых расходов. Среди исходных данных, которые используются для расчета себестоимости изделия, выделяют нормы расходов сырья и основных материалов на одно изделие.

Таблица 3.8 -Расчет расходов на сырье и основные материалы на одно изделие

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Материалы | Норма расходов  (единиц) | Оптовая цена грн./ед. | Фактические расходы  (единиц) | Сумма  грн. |
| 1 | 2 | 3 | 4 | 5 |
| Стеклотекстолит СФ-2-35  (лист 1,0 ГОСТ 10316 - 78), кг | 0,5 | 24,00 | 0,4 | 9,60 |
| Припой ПОС - 61 (ГОСТ 21930 - 76), кг | 0,05 | 18.00 | 0,05 | 0,90 |
| Всего: |  |  |  | 10,50 |
| Транспортно-заготовительные расходы (4%) |  |  |  | 0,42 |
| Итого: |  |  |  | 10,92 |

В ходе расчета себестоимости изделия, как исходные данные, используют спецификации материалов, покупных комплектующих изделии и полуфабрикатов, которые используются при сборке одного изделия (Приложение Ж).

Расчет зарплаты основных производственных рабочих проводим на основе норм трудоемкости по видам работ и по часовым ставкам рабочих (таблица 3.9).

Таблица 3.9 - Расчет основной зарплаты

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование операции | Почасовая тарифная ставка, грн. | Норма временичел./час. | Сдельная зарплата, грн. |
| 1 | 2 | 3 | 4 |
| Заготовительная | 5,67 | 1 | 5,67 |
| Фрезерная | 5,67 | 1 | 5,67 |
| Слесарная | 5,67 | 1 | 5,67 |
| Гравировка | 5,67 | 1 | 5,67 |
| Фотохимпечать | 5,67 | 2 | 11,34 |
| Гальваническая | 5,67 | 2 | 11,34 |
| Маркировочная | 5,67 | 1 | 5,67 |
| Сборка | 5,67 | 2 | 11,34 |
| Монтаж | 5,67 | 1 | 5,67 |
| Настройка | 5,67 | 2 | 11,34 |
| Другие | - | - | - |
| Всего: |  | 14 | 62,37 |

Калькуляция себестоимости и определения цены выполняется в таблице 3.10.

Таблица 3.10 **-** Калькуляция себестоимости и определения цены изделия

|  |  |
| --- | --- |
| Наименование статей расходов | Расходы  грн. |
| 1 | 2 |
| Сырье и материалы | 10,92 |
| Покупные комплектующие изделия | 31,50 |
| Транспортно-заготовительные расходы (4%) от покуп изд. | 1,26 |
| Основная зарплата рабочих | 62,37 |
| Дополнительная зарплата (15%) | 9,36 |
| Отчисление на социальные мероприятия (37,2%) | 26,68 |
| Накладные расходы (25% по данным предприятия) | 15,59 |
| Стоимость КД | 310,51 |
| Общая стоимость электрозвонка | 468,19 |

Общая стоимость изделия будет составлять:

Собщ. = С прог. + С баз. Бл. (3.21)

где С прог. – себестоимость составления программы для микроконтроллера;

С баз. Бл – себестоимость подготовки КД и сборки устройства.

При единичном изготовлении

Собщ. = 2328,69+ 468,19 = 2796,88 (грн.).

3.5 Анализ устройств-аналогов

Количество выпускаемых электрозвонков разнообразно и они отличаются, как выполняемыми функциями, так и источниками питания. Рассмотрим несколько аналогичных устройств, которые предлагаются в розничной торговле (Таблица 3.11).

Таблица 3.11 – Анализ устройств-аналогов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип,  название | Страна  изготовитель | Электропитание | Функции, кол-во | Дополнит.  функции | Цена, грн. |
| «ESSO» | Китай | 3,0В (две батарейки по 1,5 В) | Одна  мелодия | Нет | 60,00 |
| «Трембита-2» | Украина | 220 В | 4  мелодии | мелодия меняется авт. | 90,00 |
| «Соловей» | Россия | 220 В | 3  мелодии | Переключение мелодий по выбору | 120,00 |
| Нет названия | Корея | 3,0 В | Одна  мелодия | Нет | 65,00 |
| Наше устройство | Украина | 3,5-5,0В,  220 В | Семь мелодий, выбор вручную мелодии | Воспроизв. При нажатой кнопке | 160,32 |

Сравнивая наше устройство с электромузыкальными звонками-аналогами приходим к выводу, что наше устройство обладает рядом преимуществ по сравнению с предлагаемыми:

- универсальность питания;

- выбор мелодии по желанию (популярные отечественные мелодии);

- надежность устройства и малое энергопотребление;

- воспроизведение мелодий происходит только при нажатой дверной кнопке.

Проанализировав данные преимущества приходим к выводу, что данное устройство целесообразно выпускать, уменьшив стоимость за счет массового производства и систематической доработки схемы (внедрения смены дополнительных мелодий и т.д.).

Стоимость звонка сократиться при массовом производстве.

При выпуске 1000 штук электрозвонков стоимость звонка станет:

Собщ.1000 = 2328,69/1000 +310,51/1000 +157,68 = 160,32 (грн.)

Вывод:

Чем больше устройств будем выпускать, тем меньше себестоимость данного устройства. Предлагаемый электрозвонок при массовом производстве будет конкурентоспособным на рынке ( если учесть, что стоимость комплектующих также упадет за счет приобретения последних оптом).

РАЗДЕЛ 4 ОХРАНА ТРУДА

В настоящее время компьютерная техника широко применяется во всех областях деятельности человека. При работе с компьютером человек подвергается воздействию ряда опасных и вредных производственных факторов: электромагнитных полей (диапазон радиочастот: ВЧ, УВЧ и СВЧ), инфракрасного и ионизирующего излучений, шума и вибрации, статического электричества и др..

Работа с компьютером характеризуется значительным умственным напряжением и нервно-эмоциональной нагрузкой операторов, высокой напряженностью зрительной работы и достаточно большой нагрузкой на мышцы рук при работе с клавиатурой ЭВМ. Большое значение имеет рациональная конструкция и расположение элементов рабочего места, что важно для поддержания оптимальной рабочей позы человека-оператора.

В процессе работы с компьютером необходимо соблюдать правильный режим труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

4.1 Требования к производственным помещениям

4.1.1 Окраска и коэффициенты отражения

Источники света, такие как светильники и окна, которые дают отражение от поверхности экрана, значительно ухудшают точность знаков и влекут за собой помехи физиологического характера, которые могут выразиться в значительном напряжении, особенно при продолжительной работе. Отражение, включая отражения от вторичных источников света, должно быть сведено к минимуму.

Для защиты от избыточной яркости окон могут быть применены шторы и экраны.

Окраска помещений и мебели должна способствовать созданию благоприятных условий для зрительного восприятия, хорошего настроения.

В зависимости от ориентации окон рекомендуется следующая окраска стен и пола:

окна ориентированы на юг: - стены зеленовато-голубого или светло-голубого цвета; пол - зеленый;

окна ориентированы на север: - стены светло-оранжевого или оранжево-желтого цвета; пол - красновато-оранжевый;

окна ориентированы на восток: - стены желто-зеленого цвета; пол зеленый или красновато-оранжевый;

окна ориентированы на запад: - стены желто-зеленого или голубовато-зеленого цвета; пол зеленый или красновато-оранжевый.

В помещениях, где находится компьютер, необходимо обеспечить следующие величины коэффициента отражения: для потолка: 60-70%, для стен: 40-50%, для пола: около 30%. Для других поверхностей и рабочей мебели: 30-40%.

4.1.2 Освещение

Правильно спроектированное и выполненное производственное освещение улучшает условия зрительной работы, снижает утомляемость, способствует повышению производительности труда, благотворно влияет на производственную среду, оказывая положительное психологическое воздействие на работника, повышает безопасность труда и снижает травматизм.

Недостаточность освещения приводит к напряжению зрения, ослабляет внимание, приводит к наступлению преждевременной утомленности. Чрезмерно яркое освещение вызывает ослепление, раздражение и резь в глазах.

Неправильное направление света на рабочем месте может создавать резкие тени, блики, дезориентировать работающего. Все эти причины могут привести к несчастному случаю или профзаболеваниям, поэтому столь важен правильный расчет освещенности.

Существует три вида освещения - естественное, искусственное и совмещенное (естественное и искусственное вместе).

Естественное освещение - освещение помещений дневным светом, проникающим через световые проемы в наружных ограждающих конструкциях помещений.

Естественное освещение характеризуется тем, что меняется в широких пределах в зависимости от времени дня, времени года, характера области и ряда других факторов.

Искусственное освещение применяется при работе в темное время суток и днем, когда не удается обеспечить нормированные значения коэффициента естественного освещения (пасмурная погода, короткий световой день).

Освещение, при котором недостаточное по нормам естественное освещение дополняется искусственным, называется совмещенным освещением.

Искусственное освещение подразделяется на рабочее, аварийное, эвакуационное, охранное. Рабочее освещение, в свою очередь, может быть общим или комбинированным. Общее - освещение, при котором светильники размещаются в верхней зоне помещения равномерно или применительно к расположению оборудования. Комбинированное - освещение, при котором к общему добавляется местное освещение.

Согласно СНиП II-4-79 в помещений вычислительных центров необходимо применить систему комбинированного освещения.

При выполнении работ категории высокой зрительной точности (наименьший размер объекта различения 0,3…0,5мм) величина коэффициента естественного освещения (КЕО) должна быть не ниже 1,5%, а при зрительной работе средней точности (наименьший размер объекта различения 0,5…1,0 мм) КЕО должен быть не ниже 1,0%. В качестве источников искусственного освещения обычно используются люминесцентные лампы типа ЛБ или ДРЛ, которые попарно объединяются в светильники, которые должны располагаться над рабочими поверхностями равномерно.

Требования к освещенности в помещениях, где установлены компьютеры, следующие: при выполнении зрительных работ высокой точности общая освещенность должна составлять 300лк, а комбинированная - 750лк; аналогичные требования при выполнении работ средней точности - 200 и 300лк соответственно.

Кроме того все поле зрения должно быть освещено достаточно равномерно – это основное гигиеническое требование. Иными словами, степень освещения помещения и яркость экрана компьютера должны быть примерно одинаковыми, т.к. яркий свет в районе периферийного зрения значительно увеличивает напряженность глаз и, как следствие, приводит к их быстрой утомляемости.

4.1.3 Параметры микроклимата

Параметры микроклимата могут меняться в широких пределах, в то время как необходимым условием жизнедеятельности человека является поддержание постоянства температуры тела благодаря терморегуляции, т.е. способности организма регулировать отдачу тепла в окружающую среду. Принцип нормирования микроклимата – создание оптимальных условий для теплообмена тела человека с окружающей средой.

Вычислительная техника является источником существенных тепловыделений, что может привести к повышению температуры и снижению относительной влажности в помещении. В помещениях, где установлены компьютеры, должны соблюдаться определенные параметры микроклимата. В санитарных нормах СН-245-71 установлены величины параметров микроклимата, создающие комфортные условия. Эти нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера производственного помещения (см. табл. 4.1)

Объем помещений, в которых размещены работники вычислительных центров, не должен быть меньше 19,5м3/человека с учетом максимального числа одновременно работающих в смену. Нормы подачи свежего воздуха в помещения, где расположены компьютеры, приведены в табл. 4.2.

Для обеспечения комфортных условий используются как организационные методы (рациональная организация проведения работ в зависимости от времени года и суток, чередование труда и отдыха), так и технические средства (вентиляция, кондиционирование воздуха, отопительная система).

Таблица 4.1- Параметры микроклимата для помещений, где установлены компьютеры

|  |  |  |
| --- | --- | --- |
| Период года | Параметр микроклимата | Величина |
| Холодный | Температура воздуха в помещении | 22…24°С |
| Относительная влажность | 40…60% |
| Скорость движения воздуха | до 0,1м/с |
| Теплый | Температура воздуха в помещении | 23…25°С |
| Относительная влажность | 40…60% |
| Скорость движения воздуха | 0,1…0,2м/с |

Таблица 4.2 - Нормы подачи свежего воздуха в помещения, где расположены компьютеры

|  |  |
| --- | --- |
| Характеристика помещения | Объемный расход подаваемого в помещение свежего воздуха, м3 /на одного человека в час |
| Объем до 20м3 на человека | Не менее 30 |
| 20…40м3 на человека | Не менее 20 |
| Более 40м3 на человека | Естественная вентиляция |

4.1.4 Шум и вибрация

Шум ухудшает условия труда оказывая вредное действие на организм человека. Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах и т. д. Такие нарушения в работе ряда органов и систем организма человека могут вызвать негативные изменения в эмоциональном состоянии человека вплоть до стрессовых. Под воздействием шума снижается концентрация внимания, нарушаются физиологические функции, появляется усталость в связи с повышенными энергетическими затратами и нервно-психическим напряжением, ухудшается речевая коммутация. Все это снижает работоспособность человека и его производительность, качество и безопасность труда. Длительное воздействие интенсивного шума [выше 80 дБ(А)] на слух человека приводит к его частичной или полной потере.

В табл. 4.3 указаны предельные уровни звука в зависимости от категории тяжести и напряженности труда, являющиеся безопасными в отношении сохранения здоровья и работоспособности.

Таблица 4.3 - Предельные уровни звука, дБ, на рабочих местах

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Категория напряженности труда | Категория тяжести труда | | | |
| Легкая | Средняя | Тяжелая | Очень тяжелая |
| I. Мало напряженный | 80 | 80 | 75 | 75 |
| II. Умеренно напряженный | 70 | 70 | 65 | 65 |
| III. Напряженный | 60 | 60 | - | - |
| IV. Очень напряженный | 50 | 50 | - | - |

Уровень шума на рабочем месте математиков-программистов и операторов видеоматериалов не должен превышать 50дБА, а в залах обработки информации на вычислительных машинах - 65дБА. Для снижения уровня шума стены и потолок помещений, где установлены компьютеры, могут быть облицованы звукопоглощающими материалами. Уровень вибрации в помещениях вычислительных центров может быть снижен путем установки оборудования на специальные виброизоляторы.

4.1.5 Электромагнитное и ионизирующее излучения

Большинство ученых считают, что как кратковременное, так и длительное воздействие всех видов излучения от экрана монитора не опасно для здоровья персонала, обслуживающего компьютеры. Однако исчерпывающих данных относительно опасности воздействия излучения от мониторов на работающих с компьютерами не существует и исследования в этом направлении продолжаются.

Допустимые значения параметров неионизирующих электромагнитных излучений от монитора компьютера представлены в табл. 4.4.

Максимальный уровень рентгеновского излучения на рабочем месте оператора компьютера обычно не превышает 10мкбэр/ч, а интенсивность ультрафиолетового и инфракрасного излучений от экрана монитора лежит в пределах 10-100мВт/м2.

Таблица 4.4 - Допустимые значения параметров неионизирующих

электромагнитных излучений (в соответствии с СанПиН 2.2.2.542-96)

|  |  |
| --- | --- |
| Наименование параметра | Допустимые значения |
| Напряженность электрической составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора | 10В/м |
| Напряженность магнитной составляющей электромагнитного  поля на расстоянии 50см от поверхности видеомонитора | 0,3А/м |
| Напряженность электростатического поля не должна превышать:  для взрослых пользователей  для детей дошкольных учреждений и учащихся средних специальных и высших учебных заведений | 20кВ/м  15кВ/м |

Для снижения воздействия этих видов излучения рекомендуется применять мониторы с пониженным уровнем излучения (MPR-II, TCO-92, TCO-99), устанавливать защитные экраны, а также соблюдать регламентированные режимы труда и отдыха.

4.2 Эргономические требования к рабочему месту

Проектирование рабочих мест, снабженных видеотерминалами, относится к числу важных проблем эргономического проектирования в области вычислительной техники.

Рабочее место и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологическим требованиям. Большое значение имеет также характер работы. В частности, при организации рабочего места программиста должны быть соблюдены следующие основные условия: оптимальное размещение оборудования, входящего в состав рабочего места и достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения.

Эргономическими аспектами проектирования видеотерминальных рабочих мест, в частности, являются: высота рабочей поверхности, размеры пространства для ног, требования к расположению документов на рабочем месте (наличие и размеры подставки для документов, возможность различного размещения документов, расстояние от глаз пользователя до экрана, документа, клавиатуры и т.д.), характеристики рабочего кресла, требования к поверхности рабочего стола, регулируемость элементов рабочего места.

Главными элементами рабочего места программиста являются стол и кресло.

Основным рабочим положением является положение сидя.

Рабочая поза сидя вызывает минимальное утомление программиста.

Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, средств труда и документации. То, что требуется для выполнения работ чаще, расположено в зоне легкой досягаемости рабочего пространства.

Моторное поле - пространство рабочего места, в котором могут осуществляться двигательные действия человека.

Максимальная зона досягаемости рук - это часть моторного поля рабочего места, ограниченного дугами, описываемыми максимально вытянутыми руками при движении их в плечевом суставе.

Оптимальная зона - часть моторного поля рабочего места, ограниченного дугами, описываемыми предплечьями при движении в локтевых суставах с опорой в точке локтя и с относительно неподвижным плечом.

На рис. 4.1 показан пример размещения основных и периферийных составляющих ПК на рабочем столе программиста.

Для комфортной работы стол должен удовлетворять следующим условиям:

- высота стола должна быть выбрана с учетом возможности сидеть свободно, в удобной позе, при необходимости опираясь на подлокотники;

- нижняя часть стола должна быть сконструирована так, чтобы программист мог удобно сидеть, не был вынужден поджимать ноги;

- поверхность стола должна обладать свойствами, исключающими появление бликов в поле зрения программиста;

- конструкция стола должна предусматривать наличие выдвижных ящиков (не менее 3 для хранения документации, листингов, канцелярских принадлежностей);

- высота рабочей поверхности рекомендуется в пределах 680-760мм;

- высота поверхности, на которую устанавливается клавиатура, должна быть около 650мм.

Большое значение придается характеристикам рабочего кресла. Так, рекомендуемая высота сиденья над уровнем пола находится в пределах 420-

550мм. Поверхность сиденья мягкая, передний край закругленный, а угол наклона спинки - регулируемый.

**1**

**4**

**3**

**2**

**5**

Рисунок 4.1- Размещения основных и периферийных составляющих ПК на рабочем столе программиста:

1 – сканер, 2 – монитор, 3 – принтер, 4 – поверхность рабочего стола,

5 – клавиатура, 6 – манипулятор типа «мышь».

Необходимо предусматривать при проектировании возможность различного размещения документов: сбоку от видеотерминала, между монитором и клавиатурой и т.п. Кроме того, в случаях, когда видеотерминал имеет низкое качество изображения, например заметны мелькания, расстояние от глаз до экрана делают больше (около 700мм), чем расстояние от глаза до документа (300-450мм). Вообще при высоком качестве изображения на видеотерминале расстояние от глаз пользователя до экрана, документа и клавиатуры может быть равным.

Положение экрана определяется:

- расстоянием считывания (0,6 - 0,7м);

- углом считывания, направлением взгляда на 20˚ ниже горизонтали к центру экрана, причем экран перпендикулярен этому направлению.

Должна также предусматриваться возможность регулирования экрана:

- по высоте +3 см;

- по наклону от -10˚ до +20˚ относительно вертикали;

- в левом и правом направлениях.

Большое значение также придается правильной рабочей позе пользователя.

При неудобной рабочей позе могут появиться боли в мышцах, суставах и сухожилиях. Требования к рабочей позе пользователя видеотерминала следующие:

- голова не должна быть наклонена более чем на 20˚,

- плечи должны быть расслаблены,

- локти - под углом 80˚-100˚,

- предплечья и кисти рук - в горизонтальном положении.

Причина неправильной позы пользователей обусловлена следующими факторами: нет хорошей подставки для документов, клавиатура находится слишком высоко, а документы - низко, некуда положить руки и кисти, недостаточно пространство для ног.

В целях преодоления указанных недостатков даются общие рекомендации: лучше передвижная клавиатура; должны быть предусмотрены специальные приспособления для регулирования высоты стола, клавиатуры и экрана, а также подставка для рук.

Существенное значение для производительной и качественной работы на компьютере имеют размеры знаков, плотность их размещения, контраст и соотношение яркостей символов и фона экрана. Если расстояние от глаз оператора до экрана дисплея составляет 60-80 см, то высота знака должна быть не менее 3мм, оптимальное соотношение ширины и высоты знака составляет 3:4, а расстояние между знаками – 15-20% их высоты. Соотношение яркости фона экрана и символов - от 1:2 до 1:15.

Во время пользования компьютером медики советуют устанавливать монитор на расстоянии 50-60 см от глаз. Специалисты также считают, что верхняя часть видеодисплея должна быть на уровне глаз или чуть ниже. Когда человек смотрит прямо перед собой, его глаза открываются шире, чем когда он смотрит вниз. За счет этого площадь обзора значительно увеличивается, вызывая обезвоживание глаз. К тому же если экран установлен высоко, а глаза широко открыты, нарушается функция моргания. Это значит, что глаза не закрываются полностью, не омываются слезной жидкостью, не получают достаточного увлажнения, что приводит к их быстрой утомляемости.

Создание благоприятных условий труда и правильное эстетическое оформление рабочих мест на производстве имеет большое значение, как для облегчения труда, так и для повышения его привлекательности, положительно влияющей на производительность труда.

4.3 Режим труда

Как уже было неоднократно отмечено, при работе с персональным компьютером очень важную роль играет соблюдение правильного режима труда и отдыха. В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

В табл. 4.5 представлены сведения о регламентированных перерывах, которые необходимо делать при работе на компьютере, в зависимости от продолжительности рабочей смены, видов и категорий трудовой деятельности с ВДТ (видеодисплейный терминал) и ПЭВМ (в соответствии с САнНиП 2.2.2 542-96 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работ»).

Таблица 4.5 - Время регламентированных перерывов при работе на компьютере

|  |  |  |  |
| --- | --- | --- | --- |
| Категория работы с ВДТ или ПЭВМ | Уровень нагрузки за  рабочую смену при  видах работы с ВДТ, количество знаков | Суммарное время  регламентированных перерывов, мин | |
| При 8-часовой  смене | При 12-часовой  смене |
| Группа А | до 20000 | 30 | 70 |
| Группа Б | до 40000 | 50 | 90 |
| Группа В | до 60000 | 70 | 120 |

Примечание. Время перерывов дано при соблюдении указанных Санитарных правил и норм. При несоответствии фактических условий труда требованиям Санитарных правил и норм время регламентированных перерывов следует увеличить на 30%.

В соответствии со САнНиП 2.2.2 546-96 все виды трудовой деятельности, связанные с использованием компьютера, разделяются на три группы: группа А: работа по считыванию информации с экрана ВДТ или ПЭВМ с предварительным запросом; группа Б: работа по вводу информации; группа В: творческая работа в режиме диалога с ЭВМ.

Эффективность перерывов повышается при сочетании с производственной гимнастикой или организации специального помещения для отдыха персонала с удобной мягкой мебелью, аквариумом, зеленой зоной и т.п.

4.4 Расчет освещенности

Расчет освещенности рабочего места сводится к выбору системы освещения, определению необходимого числа светильников, их типа и размещения. Исходя из этого, рассчитаем параметры искусственного освещения.

Обычно искусственное освещение выполняется посредством электрических источников света двух видов: ламп накаливания и люминесцентных ламп. Будем использовать люминесцентные лампы, которые по сравнению с лампами накаливания имеют ряд существенных преимуществ:

- по спектральному составу света они близки к дневному, естественному свету;

- обладают более высоким КПД (в 1,5-2 раза выше, чем КПД ламп накаливания);

- обладают повышенной светоотдачей (в 3-4 раза выше, чем у ламп накаливания);

- более длительный срок службы.

Расчет освещения производится для комнаты площадью 15м2 , ширина которой - 5м, высота - 3 м. Воспользуемся методом светового потока.

Для определения количества светильников определим световой поток, падающий на поверхность по формуле:

F = E∙S∙Z∙К / n , (4.1)

Где F - рассчитываемый световой поток, Лм;

Е - нормированная минимальная освещенность, Лк (определяется по таблице). Работу программиста, в соответствии с этой таблицей, можно отнести к разряду точных работ, следовательно, минимальная освещенность будет Е = 300Лк;

S - площадь освещаемого помещения (в нашем случае S = 15м2);

Z - отношение средней освещенности к минимальной (обычно принимается равным 1,1-1,15 , пусть Z = 1,1);

К - коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение зависит от типа помещения и характера проводимых в нем работ и в нашем случае К = 1,5);

n - коэффициент использования, (выражается отношением светового потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризуемых коэффициентами отражения от стен (РС) и потолка (РП)), значение коэффициентов РС и РП были указаны выше: РС=40%, РП=60%. Значение n определим по таблице коэффициентов использования различных светильников.

Для этого вычислим индекс помещения по формуле:

I = A∙B / h (A+B), (4.2)

где h - расчетная высота подвеса, h = 2,92 м;

A - ширина помещения, А = 3 м;

В - длина помещения, В = 5 м.

Подставив значения получим:

I= 0,642.

Зная индекс помещения I, по таблице 7 [23] находим n = 0,22.

Подставим все значения в формулу (4.1) для определения светового потока F, получаем F = 33750 Лм.

Для освещения выбираем люминесцентные лампы типа ЛБ40-1, световой поток которых Fл = 4320 Лк.

Рассчитаем необходимое количество ламп по формуле:

N = F / Fл, (4.3)

где N - определяемое число ламп;

F - световой поток, F = 33750 Лм;

Fл- световой поток лампы, Fл = 4320 Лм.

N = 8 ламп.

При выборе осветительных приборов используем светильники типа ОД. Каждый светильник комплектуется двумя лампами.

Значит требуется для помещения площадью S = 15 м2 четыре светильника типа ОД.

Расчет естественного освещения помещений

Организация правильного освещения рабочих мест, зон обработки и производственных помещений имеет большое санитарно-гигиеническое значение, способствует повышению продуктивности работы, снижения травматизма, улучшения качества продукции. И наоборот, недостаточное освещение усложняет исполнения технологического процесса и может быть причиной несчастного случая и заболевания органов зрения.

Освещение должно удовлетворять такие основные требования:

- быть равномерным и довольно сильным;

- не создавать различных теней на местах работы, контрастов между освещенным рабочем местом и окружающей обстановкой;

- не создавать ненужной яркости и блеска в поле взора работников;

- давать правильное направление светового потока;

Все производственные помещения необходимо иметь светлопрорезы, которые дают достаточное природное освещение. Без природного освещения могут быть конференц-залы заседаний, выставочные залы, раздевалки, санитарно-бытовые помещения, помещения ожидания медицинских учреждений, помещений личной гигиены, коридоры и проходы.

Коэфициент естественного освещения в соответствии с ДНБ В 25.28.2006, для нашого III пояса светового климата составляет 1,5.

Исходя из этого произведем расчет необходимой площади оконных проемов.

Расчет площади окон при боковом освещении определяется, по формуле:

Sо = (Ln\*Кз.\*N0\*Sn\*Кзд.)/(100 \*T0\*r1) (4.4)

Где, Ln – нормированное значение КЕО;

Кз – коэффициент запаса (равен 1,2);

N0 – световая характеристика окон;

Sn – площадь достаточного естественного освещения;

Кзд. – коэффициент, учитывающий затенение окон противостоящими зданиями;

r1 – коэффициент, учитывающий повышение КЕО при боковом освещении

T0 – общий коэффициент светопропускания, который рассчитывается по формуле:

T0 = T1 \* T2 \* T3 \* T4 \* T5, (4.5)

где T1 – коэффициент светопропускания материала;

T2 – коэффициент, учитывающий потери света в переплетах светопроема;

T3 – коэффициент, учитывающий потери света в несущих конструкциях;

T4 – коэффициент, учитывающий потери света в солнцезащитный устройствах;

T5 – коэффициент, учитывающий потери света в защитной сетке, устанавливаемой под фонарями, принимается равным 1;

Теперь следует рассчитать боковое освещение для зоны, примыкающей к наружной стене. По разряду зрительной работы нужно определить значение КЕО. КЕО = 1,5 нормированное значение КЕО с учетом светового климата необходимо вычислить по формуле:

Ln=*l*\*m\*c, (4.6)

где *l* – значение КЕО (l=1.5);

m – коэффициент светового климата (m=1);

c – коэффициент солнечности климата (c=1)

Ln=1,5

Теперь следует определить отношение длины помещения Ln к глубине помещения B:

Ln/B=3/5 =0,6;

Отношение глубины помещения В к высоте от уровня условной рабочей поверхности до верха окна h1 (в данном случае h1=1,8) :

B/h1=5/1,8 = 2,77.

Световая характеристика световых проемов N0=9.

Кзд=1

Значение T0=0,8\*0,7\*1\*1\*1=0,56.

Ln для 4 разряда зрительных работ равен 1,5 при мытье окон два раза в год.

Определяем r1, r1=1,5.

Кз.=1,2.

Теперь следует определить значение Sп:

Sп=Ln\*В=3\*10=30 м2.

Кзд.=1.

На данном этапе следует рассчитать необходимую площадь оконных проемов: (Ln\* Кз.\*N0\*Sn\*Кзд.) / (100\*T0\*r1)

Sо = (1,5\*1,2\*9\*30\*1)/(100\*0,56\*1,5)=486/84= 5,78 м2;

Принимаем количество окон 1 штука:

S1=5,78 м2 площадь одного окна

Высота одного окна составляет – 2,5 м, ширина 2,3 м.

4.5. Расчет вентиляции

В зависимости от способа перемещения воздуха вентиляция бывает естественная и принудительная.

Параметры воздуха, поступающего в приемные отверстия и проемы местных отсосов технологических и других устройств, которые расположены в рабочей зоне, следует принимать в соответствии с ГОСТ 12.1.005-76. При размерах помещения 3 на 5 метров и высоте 3 метра, его объем 45 куб.м. Следовательно, вентиляция должна обеспечивать расход воздуха в 90 куб.м/час. В летнее время следует предусмотреть установку кондиционера с целью избежания превышения температуры в помещении для устойчивой работы оборудования. Необходимо уделить должное внимание количеству пыли в воздухе, так как это непосредственно влияет на надежность и ресурс эксплуатации ЭВМ.

Мощность (точнее мощность охлаждения) кондиционера является главной его характеристикой, от неё зависит на какой объем помещения он рассчитан. Для ориентировочных расчетов берется 1 кВт на 10 м2 при высоте потолков 2,8 – 3 м (в соответствии со СНиП 2.04.05-86 "Отопление, вентиляция и кондиционирование").

Для расчета теплопритоков данного помещения использована упрощенная методика:

Q=S·h·q (4.8)

где: Q – Теплопритоки

S – Площадь помещения

h – Высота помещения

q – Коэффициент равный 30-40 вт/м3 (в данном случае 35 вт/м3)

Для помещения 15 м2 и высотой 3 м теплопритоки будут составлять:

Q=15·3·35=1575 вт

Кроме этого следует учитывать тепловыделение от оргтехники и людей, считается (в соответствии со СНиП 2.04.05-86 "Отопление, вентиляция и кондиционирование") что в спокойном состоянии человек выделяет 0,1 кВт тепла, компьютер или копировальный аппарат 0,3 кВт, прибавив эти значения к общим теплопритокам можно получить необходимую мощность охлаждения.

Qдоп=(H·Sопер)+(С·Sкомп)+(P·Sпринт) (4.9)

где: Qдоп – Сумма дополнительных теплопритоков

C – Тепловыделение компьютера

H – Тепловыделение оператора

D – Тепловыделение принтера

Sкомп – Количество рабочих станций

Sпринт – Количество принтеров

Sопер – Количество операторов

Дополнительные теплопритоки помещения составят:

Qдоп1=(0,1·2)+(0,3·2)+(0,3·1)=1,1(кВт)

Итого сумма теплопритоков равна:

Qобщ1=1575+1100=2675 (Вт)

В соответствии с данными расчетами необходимо выбрать целесообразную мощность и количество кондиционеров.

Для помещения, для которого ведется расчет, следует использовать кондиционеры с номинальной мощностью 3,0 кВт.

4.6 Расчет уровня шума

Одним из неблагоприятных факторов производственной среды в ИВЦ является высокий уровень шума, создаваемый печатными устройствами, оборудованием для кондиционирования воздуха, вентиляторами систем охлаждения в самих ЭВМ.

Для решения вопросов о необходимости и целесообразности снижения шума необходимо знать уровни шума на рабочем месте оператора.

Уровень шума, возникающий от нескольких некогерентных источников, работающих одновременно, подсчитывается на основании принципа энергетического суммирования излучений отдельных источников:

∑L = 10·lg (Li∙n), (4.10)

где Li – уровень звукового давления i-го источника шума;

n – количество источников шума.

Полученные результаты расчета сравнивается с допустимым значением уровня шума для данного рабочего места. Если результаты расчета выше допустимого значения уровня шума, то необходимы специальные меры по снижению шума. К ним относятся: облицовка стен и потолка зала звукопоглощающими материалами, снижение шума в источнике, правильная планировка оборудования и рациональная организация рабочего места оператора.

Уровни звукового давления источников шума, действующих на оператора на его рабочем месте представлены в табл. 4.6.

Таблица 4.6 - Уровни звукового давления различных источников

|  |  |
| --- | --- |
| Источник шума | Уровень шума, дБ |
| Жесткий диск | 40 |
| Вентилятор | 45 |
| Монитор | 17 |
| Клавиатура | 10 |
| Принтер | 45 |
| Сканер | 42 |

Обычно рабочее место оператора оснащено следующим оборудованием: винчестер в системном блоке, вентилятор(ы) систем охлаждения ПК, монитор, клавиатура, принтер и сканер.

Подставив значения уровня звукового давления для каждого вида оборудования в формулу (4.4) , получим:

∑L=10·lg(104+104,5+101,7+101+104,5+104,2)=49,5 дБ

Полученное значение не превышает допустимый уровень шума для рабочего места оператора, равный 65 дБ (ГОСТ 12.1.003-83). И если учесть, что вряд ли такие периферийные устройства как сканер и принтер будут использоваться одновременно, то эта цифра будет еще ниже. Кроме того при работе принтера непосредственное присутствие оператора необязательно, т.к. принтер снабжен механизмом автоподачи листов.

В данном разделе дипломной работы были изложены требования к рабочему месту инженера - программиста. Созданные условия должны обеспечивать комфортную работу. На основании изученной литературы по данной проблеме, были указаны оптимальные размеры рабочего стола и кресла, рабочей поверхности, а также проведен выбор системы и расчет оптимального освещения производственного помещения, произведен расчет рационального кондиционирования помещения, а также расчет уровня шума на рабочем месте. Соблюдение условий, определяющих оптимальную организацию рабочего места инженера - программиста, позволит сохранить хорошую работоспособность в течение всего рабочего дня, повысит как в количественном, так и в качественном отношениях производительность труда программиста, что в свою очередь будет способствовать быстрейшей разработке и отладке программного продукта.

ВЫВОДЫ

В данном проекте был разработан музыкальный звонок с двумя режимами работы: автономный ( на батарейках) и от сети.

В процессе разработки были рассмотрены общие вопросы разработки устройств на микроконтроллерах, рассмотрены этапы разработки, разработка программного обеспечения, внедрения в производство. На основании изученной литератур и сети Интернет получили теоретические знания в области проектирования устройств на микроконтроллерах.

В практической части были разработаны структурная, функциональная и принципиальная схемы устройства, составлен алгоритм работы микроконтроллера, выбран микроконтроллер удовлетворяющий требованиям ТЗ, произведен расчет элементной базы устройства, произведено подробное описание программы.

В экономической части был произведен расчет себестоимости проектируемого изделия, сравнение с устройствами- аналогами, сделан вывод о целесообразности изготовления данного устройства.

В последнем разделе дипломной работы были изложены требования к охране труда. Созданные условия должны обеспечивать комфортную работу. На основании изученной литературы по данной проблеме, были указаны оптимальные размеры рабочего стола и кресла, рабочей поверхности, а также проведен выбор системы и расчет оптимального освещения производственного помещения, произведен расчет рационального кондиционирования помещения, а также расчет уровня шума на рабочем месте. Соблюдение условий, определяющих оптимальную организацию рабочего места, позволит сохранить хорошую работоспособность в течение всего рабочего дня, повысит как в количественном, так и в качественном отношениях производительность труда, что в свою очередь будет способствовать быстрейшей разработке и отладке программного продукта.

ПЕРЕЧЕНЬ ССЫЛОК

1. "Dimmable Fluorescent Ballast" – User Guide, 10/07, Atmel Corporation, http://www.atmel.com/dyn/resources/prod\_documents/doc7597.pdf

2. ГОСТ13109-97. Нормы качества электрической энергии в системах электроснабжения общего назначения.

3. G. Howell "Five questions about resistors" // EDN, 9/28/2006, http://www.edn.com/contents/images/6372835.pdf

4. П. Хоровиц, У. Хилл "Искусство схемотехники" – Изд. 6-е, М.: Мир, 2003.

5. C. Hillman "Common mistakes in electronic design" // EDN, 12/14/2007 http://www.edn.com/contents/images/6512156.pdf

6."Frequently asked questions about dimmers" // http://www .lutron.com/product\_technical/faq.asp

7. Л. Н. Кечиев, Е. Д. Пожидаев "Защита электронных средств от воздействия статического электричества" – М.: ИД "Технологии", 2005.

8. Жидецкий В.Ц., Джигирей В.С., Мельников А.В. Основы охраны труда: Учебник – Львов, Афиша, 2008 – 351с.

9. Денисенко Г.Ф. Охрана труда: Учебн.пособие – М., Высшая школа, 1989 – 319с.

10. Самгин Э.Б. Освещение рабочих мест. – М.: МИРЭА, 1989. – 186с.

11. Справочная книга для проектирования электрического освещения. / Под ред. Г.Б. Кнорринга. – Л.: Энергия, 1976.

12. Борьба с шумом на производстве: Справочник / Е.Я. Юдин, Л.А. Борисов;

Под общ. ред. Е.Я. Юдина – М.: Машиностроение, 1985. – 400с., ил.

13. Зинченко В.П. Основы эргономики. – М.: МГУ, 1979. – 179с.

14.Методичні вказівки до виконання дипломної роботи для учнів спеціальності «Оператор комп’ютерного набору; оператор комп’ютерної верстки»/ Упоряд.: Д.О. Дяченко, К.О. Ізмалкова, О.Г. Меркулова. – Сєверодонецьк: СВПУ, 2007. – 40 с.

15.Н. Заец. Многофункциональные часы. — Схемотехника, 2006, № 2,с. 41,42.

16.Н. Заец. Термометр - часы с датчиками фирмы Dallas Semiconductor. — Схемотехника, 2005, № 5, с. 52 - 55.

17.Н. Заец. Радиолюбительские конструкции на Р1С-микроконтроллерах. Книга 3. — М.: СОЛОН-Пресс, 2005, с. 248.

18.Н. Заец. Отечественные жидкокристаллические индикаторы TIC9162 с драйверами по технологии COG. - Схемотехника, 2005, №9, с. 9-11.

19. Н. Заец. Таймеры десятичного счета. — Электрик, 2006, № 7-8, с. 36 -39.

20.Мощные полевые переключательные транзисторы фирмы International Rectifier. — Радио, 2001, № 5, с. 45.

21.А. Долгий. Разработка и отладка устройств на микроконтроллерах. — Радио, 2001 ,№ 5-12, 2002, № 1.

22.А. Долгий. Программаторы и программирование микроконтроллеров. — Радио, 2004, № 1-12.

23.Н. Заец. Универсальный таймер. — Схемотехника, 2003, № 1, с. 53.

24.Н. Заец. Электронные самоделки для быта, отдыха **и** здоровья.— М.: СОЛОН-Пресс, 2009, 423 с.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Файл основной программы

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | |  | | |  | Псевдокоманды управления | | | |
| 1 | | include | | Tn23l3def | | | inc. | :присоединение файла описание | | | |
| 2 | | . list | |  | | |  | : Включение листинга | | | |
| 3 | | .def | | loop1 **=** | | | RO | . Три ячейки для процедуры задержки | | | |
| 4 | | .def | | loop2 = | | | R1 |  | | | |
| ***5*** | | .def | | loop З = | | | R21 |  | | | |
| 6 | | .def | | temp = | | | R16 | : Вспомогательный регистр | | | |
| **7** | | .def | | temp1 = | | | R17 | : Второй вспомогательный регистр | | | |
| 8 | | .def | | count = | | | R17 | :Определение регистра счетчика опросаклавиш | | | |
| 9 | | .def | | fnota= | | | R19 | :Частота текущей ноты | | | |
| 10 | | .def | | dnota | | | = R20 | :Длительность текущей ноты | | | |
|  | | -------- Начало программного кода | | | | | | | | | |
| 11 | |  | | cseg | |  | | :Выбор сегмента программного кода | | | |
| 12 | |  | | org | | 0 | | : Установка текущего адреса на ноль | | | |
| 13 | | start | | rtmp | | init : Переход на начало программы | | | | | |
| 14 | |  | | ret1 | |  | | : Внешнее прерывание 0 | | |
| 15 | |  | | ret1 | |  | | : Внешнее прерывание 1 | | |
| 16 | |  | | ret1 | |  | | : Таймер/счётчик 1 захват | | |
| 17 | |  | | ret1 | |  | | : Таймер/счетчик 1. совпадение канал А | | |
| 18 | |  | | ret1 | |  | | : Таймер/счетчик 1, прерывание попереполнению | | |
| 19 | |  | | ret l | |  | | : Таймер/счетчик 0, прерывание по переполнению | | |
| 20 | |  | | ret1 | |  | | ; Прерывание UART прием завершен | | |
| 21 | |  | | ret1 | |  | | ; Прерывание UART регистр данных пуст | | |
| 22 | |  | | ret1 | |  | | . Прерывание UART передана завершена | | |
| 25 | |  | | ret1 | |  | | ; Прерывание по компаратору | | |
| 24 | |  | | ret1 | |  | | : Прерывание ло изменений на любом контакте | | |
| 25 | |  | | ret1 | |  | | : Таимер/счетчик 1 Совпадение, канал В | | |
| 26 | |  | | ret1 | |  | | : Таймер/счетчик 0 Совпадение, канал В | | |
| 27 | |  | | ret1 | |  | | :Таймер/ счетчик 0 Совпадение, канал А | | |
| 28 | |  | | ret1 | |  | | USI готовность кстарту | | |
| 29 | |  | | ret1 | |  | | USI Переполнение | | |
| 30 | |  | | ret1 | |  | | EEPR0M Готовность | | |
| 31 | |  | | ret1 | |  | | Переполнение охранного таймера | | |
|  |  | |  | |  | | | |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | init |  |  | |  |
|  |  |  | Инициализация стека | | |
| 32 |  | ld1 | temp RAMED | Инициализация стека | |
| 33 |  | out | SPL temp |  | |
|  |  |  | Инициализация портов B/B | | |
| 34 |  | ld1 | temp 0x08 | Инициализация порта PB | |
| 35 |  | out | temp 0x7F | Инициализация порта PD | |
| 36 |  | out | PORT temp |  | |
| 37 |  | ld1 | temp 0x00 |  | |
| 38 |  | out | DDRD: temp |  | |
|  |  |  | Инициализация (выключение) компаратора | | |
| 39 |  | ld1 | temp 0x80 |  | |
| 40 |  | out | ACSR. temp |  | |
|  |  |  | Инициализация таймера T1 | | |
| 41 |  | ld1 | temp 0x09 | Включение режима CTC | |
| 42 |  | out | TCCR1B temp |  | |
| 43 | m1 | ld1 | temp 0x00 | Выключение звука | |
| 44 |  | out | TCCR1A temp |  | |
|  |  | Начало основной программы | |  | |
|  | main: |  |  |  | |
|  |  |  | Вычисление номера нажатой кнопки | | |
| 45 |  | clr | COUNT | Обнуление счетчика опроса клавиш | |
| 46 |  | ln | temp. PIND | Чтение порта D | |
| 47 | m2 | lsr | temp | Сдвигаем входной байт | |
| 48 |  | brcc | m3 | Если текущий разряд был равен 0 | |
| 49 |  | lnc | count | Увеличиваем показание счетчика | |
| 50 |  | cp1 | count 7 | Сравнение(7-конец сканирования) | |
| 51 |  | brne | m2 | Если не конец продолжить | |
| 52 |  | rlmp | m1 | Если не одна клавиша не нажата | |
|  |  |  | Выбор мелодии | | |
| 53 | m3 | mov | YL count | Вычисляем адрес где | |
| 54 |  | ld1 | ZL low(tamb\*2) | Храниться начало мелодии | |
| 55 |  | ld1 | ZH high(tamb\*2) |  | |
| 56 |  | lcall | addw | К программе 16-разрядного сложения | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | | |
| 57 |  | lpm | | XL Z+ | Извлекаем адрес из таблицы |
| 58 |  | lpm | | XH Z | И помещаем в x |
|  |  |  | | Воспроизведение мелодии | |
| 59 | m4 | mov | | ZH. XH | Записываем в Z начало мелодии |
| 60 |  | mov | | ZL .XL |  |
| 61 | m5 | ln | | temp PINO | Читаем содержимое порта D |
| 62 |  | cpl | | temp 0x7F | Проверяем на равенство 7FH |
| 63 |  | dreg | | m1 | Если равно(скобки опущены) в начало |
| 64 |  | lpm | | temp Z | Извлекаем код ноты |
| 65 |  | cpl | | temp 0xFF | Проверяем не конец ли мелодии |
| 66 |  | dreg | | m4 | Если конец начинаем мелодию сначала |
| 67 |  | and1 | | temp 0x1f | Выделяем код тона из кода ноты |
| 68 |  | mov | | tnota temp | Записываем в регистр кода тона |
| 69 |  | lpm | | temp Z+ | Еще раз берем код ноты |
| 70 |  | col | | temp | Производим сдвиг кода ноты |
| 71 |  | col | | temp |  |
| 72 |  | col | | temp |  |
| 73 |  | col | | temp |  |
| 74 |  | and1 | | temp. 0х07 | выделяем код длительности |
| 75 |  | mov | | dnota temp | помещаем ее в регистр длительности |
| 76 |  | rcall | | nota | к подпрограмме воспроизведения ноты |
| 77 |  | rjmp | | m S | В начало цикла (следующая нота) |
|  |  | Вспомогательные подпрограммы | | | |
| 78 | addw | duch | YH | | подпрограмма 16 разрядного сложения |
| 79 |  | lsl | YL | | Умножение первого слагаемого на 2 |
| 80 |  | ld1 | YH.O | | второй байт первого слагаемого |
| 81 |  | add | ZL. YL | | Складывается два слагаемых |
| 82 |  | adc | ZH. YH | |  |
| 83 |  | pop | YH | |  |
| 84 |  | ret |  | |  |
|  |  |  | Подпрограмма исполнения одной ноты | | |
| 85 | nota. | push | ZH | |  |
| 86 |  | push | ZL | |  |
| 87 |  | push | YL | |  |
| 88 |  | push | temp | |  |
| 89 |  | col | fnota. 0x00 | | Проверка не пауза ли |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 90 |  | breg | ntl | Если пауза переходим сразу к задержке |
| 91 |  | mov | YL. fnota | Вычисляем адрес где храниться |
| 92 |  | idi | ZL low (tabkd\*2) | коэффициент деления текущей ноты |
| 93 |  | idi | ZH high (tabkd\*2) |  |
| 94 |  | rcali | addw | К подпрограмме 16-разрядного сложения |
| 95 |  | lpm | temp Z+ | Извлекаем ид. Разряд КД для текущей ноты |
| 96 |  | lpm | templ Z | Извлекаем ст. разряд КД для текущей ноты |
| 97 |  | out | OCRIAH temp | Записать в старш. Часть регистра совпадения |
| 98 |  | out | OCRIAL temp | Записать в младш. Часть регистра совпадения |
| 99 |  | idi | temp 0x40 | Включить звук |
| 100 |  | out | TCCRIA temp |  |
| 101 | ntl: | rcall | wait | К программе задержки |
| 102 |  | idi | temp 0x00 | Выключить звук |
| 103 |  | out | TCCRIA temp |  |
| 104 |  | idi | dnota 0 | Отбрасываем задержку для паузы между нотами |
| 105 |  | rcall | wait | Пауза между нотами |
| 106 |  | pop | temp | Завершение подпрограммы |
| 107 |  | pop | YL |  |
| 108 |  | pop | ZL |  |
| 109 |  | pop | ZH |  |
| 110 |  | ret |  |  |
|  |  |  | Подпрограмма формирования задержки | |
| 111 | wait | push | ZH |  |
| 112 |  | push | ZL |  |
| 113 |  | push | YH |  |
| 114 |  | push | YL |  |
| 115 |  | mov | YL dnota | Вычисляем адрес где храниться |
| 116 |  | ld1 | ZL low(tabz\*2) | нужный коэффициент задержки |
| 117 |  | ld1 | ZH high(tabz\*2) | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| 118 |  | rcall | addw | К программе 16-разрядного сложения |
| 119 |  | lpm | YL Z+ | Читаем первый байт коэффициента задержки |
| 120 |  | lpm | YH Z | Читаем второй байт коэффициента задержки |
| 121 |  | ctr | ZL | обнуляем регистровую пару Z |
| 122 |  | ctr | ZH |  |
|  |  | Цикл задержки | |  |
| 123 | w1 | ldl | loop 255 | Пустой внутренний цикл |
| 124 | w2 | dec | loop |  |
| 125 |  | brne | w2 |  |
| 126 |  | adlw | R3O 1 | Увеличение регистровой пары Z на единицу |
| 127 |  | co | YL ZL | Проверка младшего раздела |
| 128 |  | brne | w1 |  |
| 129 |  | cp | YH ZH | Проверка старшего разряда |
| 130 |  | brne | w1 |  |
| 131 |  | pop | YL | Завершение подпрограммы |
| 132 |  | pop | YH |  |
| 133 |  | pop | ZL |  |
| 134 |  | pop | ZH |  |
| 135 |  | ret |  |  |
|  |  | Таблица длительности задержек | | |
| 136 | tabz | dw | 128.256/512/1024.2048.4096.8192 | |
|  |  | Таблица коэффициента деления | | |
| 137 | tabkd | dw | 0 |  |
| 138 |  | dw | 4748.4480.4228.3992.3768.3556.3358.3168.2990.2822.2664.2514 | |
| 139 |  | dw | 2374.2240.2114.1996.1884.1778.1678.1584.1495.1411.1332.1257 | |
| 140 |  | dw | 1187.1120/1057.998.942.889.839.792 | |
|  |  | Таблица начал всех мелодий | | |
| 141 | tabа | dw | mei1\*2, mei2\*2, mei3\*2, mei4\*2, | |
| 142 |  | dw | mei5\*2, mei6\*2, mei7\*2, | |
|  |  | Таблица мелодий | | |
|  |  | В траве сидел кузнечек | | |
| 143 | mel1 | db | 109.104.109.104.109.108.108.96.108.104 | |
| 144 |  | db | 108.104.108.109.109.96.109.104.109.104 | |
| 145 |  | db | 109.108.108.96.108.104. 109.104.109.104 | |
| 146 |  | db | 112.112.112.111. 108.109.109.96.109.104 | |
|  | | | | |
| 147 |  | db | 111.79.79.111.111.112.80.80.112.112.80 | |
| 148 |  | db | 112.112.112.111. 108.109. 104.109.104 | |
| 149 |  | db | 109.108.108.96.108.141.128.96.255 | |
|  |  | Песенка крокодила Гены | | |
| 150 | mel2 | db | 109.110.141.102.104.105.102.109.110.141 | |
| 151 |  | db | 111.79.79.111.111.112.80.80. 108.96.108 | |
| 152 |  | db | 108.96.108.141.104.105.102.109.110.141 | |
| 153 |  | db | 79.111.111.112.80. 96.108.104. 109 | |
| 154 |  | db | 146.109.105.136.107.105.134.128.128.102 | |
| 155 |  | db | 79.111.111.112.80. 96.110.176.112.108.109 | |
| 156 |  | db | 109.105.136.107.105.134.105.142.146.144. | |
| 157 |  | db | 105.136.107.105.134. 134.128.128.102.141 | |
| 158 |  | db | 111.111.112.80. 96.110.176.136.107.105.134 | |
| 159 |  | db | 105.134. 134.128.128.142.141.105.109.139 | |
| 160 |  | db | 126.173.146.128.96.255 | |
|  |  | В лесу родилась елочка | | |
| 161 | mel3 | db | 132.141.141.139.141.137.132.132.132.141 | |
| 162 |  | db | 141.142.139.176.128.144.146.146.154.154 | |
| 163 |  | db | 153.151.149.144.153.153.151.153.181.128 | |
| 164 |  | db | 96.255 | |
|  |  | Happy births to you | | |
| 165 | mel4 | db | 107.107.141.139.144.143.128.107.107.141 | |
| 166 |  | db | 139.146.144.128.107.107.151.148.146.112 | |
| 167 |  | db | 111.149.117.117.148.144.146.144.128.255 | |
|  |  | С чего начинается родина | | |
| 168 | mel5 | db | 99.175.109.107.106.102.99.144.111.175 | |
| 169 |  | db | 96.99.107.107.107.107.102.104.170.96 | |
| 170 |  | db | 99.109.109.109.109.107.106.143.109.141 | |
| 171 |  | db | 99.109.109.109.109.104.106.171.96.99 | |
| 172 |  | db | 111.109.107.106.102.99.144.111.143.104 | |
| 173 |  | db | 114.114.114.114.109.111.176.96.104.116 | |
| 174 |  | db | 112.109.107.106.64.73.143.107.131.99 | |
| 175 |  | db | 144.80.80.112.111.64.75.173.128.255 | |
|  |  | Песня из кинофильма «Веселые ребята» | | |
| 176 | mel6 | db | 105.109.112.149.116.64.80.148.114.64 | |
| 177 |  | db | 78.146.112.96.105.105.109.144.111.64 | |
| 178 |  | db | 80.145.112.64.81.178.96.117.117.117 | |
| 179 |  | db | 149.116.64.82.146.112.64.79.146.144 | |
| 180 |  | db | 96.105.105.107.141.108.109.112.110.102 | |
| 181 |  | db | 104.137.128.96.105.105.105.137.102.64 | |
| 182 |  | db | 73.142.105.107.109.64.75.137.96.105 | |
| 183 |  | db | 105.105.137.102.105.142.112.64.82.180 | |
| 184 |  | db | 96.116.116.116.148.114.112.142.109.64 | |
| 185 |  | db | 78.146.144.96.105.105.107.141.108.109 | |
| 186 |  | db | 112.110.102.104.169.96.96.255 | |
|  |  | Улыбка | | |
| 187 | mel7 | db | 107.104.141.139.102.105.104.102.164.128 | |
| 188 |  | db | 104.107.109.109.109.111.114.112.111.109 | |
| 189 |  | db | 144.139.128.109.111.144.96.111.109.104 | |
| 190 |  | db | 107.105.173.128.111.109.112.107.111.109 | |
| 191 |  | db | 109.107.102.104.134.132.128.100.103.107 | |
| 192 |  | db | 107.107.107.139.112.100.103.102.102.102 | |
| 193 |  | db | 134.102.103.107.105.107.108.108.108.108 | |
| 194 |  | db | 107.105.107.108.144.142.128.112.107.110 | |
| 195 |  | db | 140.112.105.108.107.107.107.105.140.139 | |
| 196 |  | db | 139.112.103.102.103.105.108.107.105.103 | |
| 197 |  | db | 128.112.107.110.108.108.108.108.140.112.105 | |
| 198 |  | db | 108.107.107.107.139.112.103.102.103.105 | |
| 199 |  | db | 108.107.105.103.105.1039.132.128.96.96 | |
| 200 |  | db | 96.255 | |

ПРИЛОЖЕНИЕ Б

Файл программы прерываний

Chip type Altiny2313

Clock fregency 4.000000 MHz

……………………………………………/

1. #incluid <tine2313 h>
2. #incluid <delay h>

//Объявление инициализация массивов

//Таблица задержек

3 flash unsidned int tabkd[]={16.32.64.128.256.512.1024}

//Массив коэффициента деления

1. flash unsidned int tabkd[]={0.4748.4480.4228.3992.3768.3556.3358.3168.2990.2822

5 2664.2514. 2374.2240.2114.1996.1884.1778.1678.1584.1495.1411.1332.1257

6 1187.1120/1057.998.942.889.839.792

//Таблица мелодий

//В траве сидел кузнечик

7 flash unsidned int tabkd[]={ 109.104.109.104.109.108.108.96.108.104. 108.104.108.

8 109.109.96.109.104.109.104.109.108.108.96.108.104. 109.104.109.104.109.

9 111.79.79.111.111.112.80.80.112.112.80.112.112.112.111.108.109.109.

10 96.109.104.109.108.108.96.108.141.128.96.255 }

//Песенка крокодила Гены

11 flash unsidned int tabkd[]={ 109.110.141.102.104.105.102.109.110.141.104.105.107

12 111.79.79.111.111.112.80.80.108.96.108.108.96.108.141.104.105.102.109.114

13 110.141.79.111.112.80.96.108.104.109.146.105.109.105.105.134.128.128.105

14 79.111.111.112.80.96.110.176.112.108.109.105.109.105.136.107.105.134.105.

15 105.136.107.142.134.134.128.128.102.141.105.134.134.128.128.142.141.105.

16 144. 105.134. 134.128.128.141.105.109.139.126.173.146.128.96.255 }

//В лесу родилась елочка

17 flash unsidned int tabkd[]={ 132.141.141.139.141.137.132.132.132.141.141.142.139.

18 176.128.144.146.146.154.154.153.151.149.153.153.151.153.181.128.96.255}

//Happy births day to you

19 flash unsidned int tabkd[]={107.107.141.139.144.143.128.107.107.141.139.146.144.

20 143.128.107.107.141.128.111.149.117.117.148.144.146.144.128.255}

//С чего начинается родина

21 flash unsidned int tabkd[]={99.175.109.107.106.102.99.144.111.175.96.99.107.107.

22 107.107.102.104.170.96.99.109.109.107.106.143.109.141.99.109.109.109.109.

23 104.106.171.96.99.111.109.107.106.102.99.144.111.143.104.114.114.114.114.

24 109.111.176.96.104.116.112.109.107.106.64.73.143.107.131.99.144.80.80.112.

25 111.64.75.173.128.255}

//Из кинофильма «Веселые ребята»

26 flash unsidned int tabkd[]={105.109.112.149.116.64.80.148.114.64.78.146.112.96.105.

27 105.109.144.111.64.80.145.112.64.81.178.96.117.117.117. 149.116.64.82.146.112.

28 64.79.146.144. 96.105.105.107.141.108.109.112.110.102. 104.137.128.96.105.105.

29 105.137.102.64.73.142.105.107.109.64.75.137.96.105.105.105.137.102.105.142.

30 64.82.180.96.116.116.116.148.114.112.142.109.64.78.146.144.96.105.105.107.141.

31 108.109.112.110.102.104.169.96.96.255}

//Улыбка

32 flash unsidned int tabkd[]={107.104.141.139.102.105.104.102.164.128. 104.107.109.109.

33 109.111.114.112.111.109.144.139.128.109.144.96.111.109.104.107.105.173.128.

34 111.109.112.107.111.109.109.107.102.104.134.132.128.100.103.107.107.107.139.

35 112.100.103.102.102.102.134.102.103.107.105.107.108.108.108.108.107.105.108.

36 144.142.128.112.107.110. 140.112.105.108.107.107.107.105.140.139. 139.112.103.

37 103.105.108.107.105.103. 128.112.107.110.108.108.108.108.140.112.105.107.107.

38 139.112.103.102.103.105. 103.105.139.132.128.96.96. 96.255}

//Воспроизведение мелодии

39 nota=tamb[count]. //Установить указатель на первую ноту

40 if(PIND==0x78)goto m2 //Если ни одна кнопка не нажата. закончить

41 if(dnota==0xFF)goto m3 //Проверка на конец мелодии

43 fnota=(dnota)$0x1F //Определение кода тона

44 dnota=((dnota)>>5)$0x07 //Определение код длительности

45 if(fnota==0)goto m5 //Если пауза не воспроизводим звук

46 OCR1A=0x40 //Программируем частоту звука

47 TCCR1A=0x40 //Включаем звук

48 delay ms (tabz[dnota]) //Формируем задержку

49 TCCR1A=0x00. //Выключаем звук

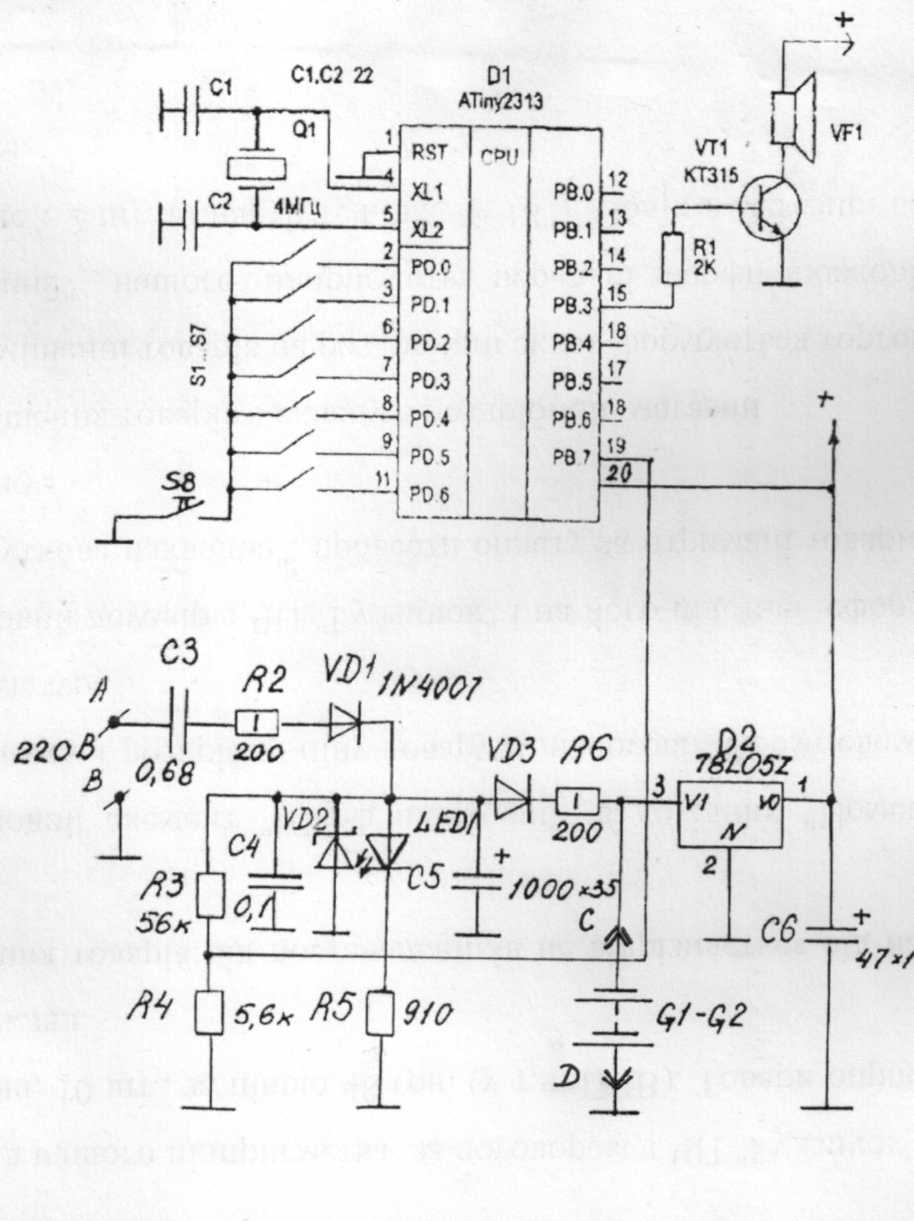
50 delay ms (tabz[0]) //Задержка между нотами

51 nota++ //Перемещаем указатель на следующую ноту

52 goto m4 //К началу цикла

ПРИЛОЖЕНИЕ В

Схема принципиальная электрическая музыкального звонка с двумя режимами работы: автономный и от сети



ПРИЛОЖЕНИЕ Ж

Калькуляция комплектующих электромузыкального звонка

