|  |  |  |
| --- | --- | --- |
| Балтийский Государственный технический университет им. Д.Ф.Устинова (“Военмех”) | | |
| Кафедра И4 | | |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  к курсовому проекту по дисциплине  Проектирование радиоэлектронных устройств    “Устройство цифровой записи речи  (цифровой диктофон)” | | |
|  |  |  |
| Группа | И-4 |
| Студент | С |
| Проверил | П |
|  | | |
| Санкт-Петербург  200 | | |

## СОДЕРЖАНИЕ

Задание 3

Перечень условных обозначений 4

Введение 5

Анализ исходных данных 6

Выбор структуры устройства 8

Выбор элементов 10

Принцип действия 14

Описание принципиальной схемы устройства 18

Заключение 41

Список литературы 42

# ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

"Устройство цифровой записи речи"

**Принцип действия**

Происходит запись речевого сигнала в микросхемы памяти после соответствующего аналого-цифрового преобразования.

Дополнительный сервис по усмотрению разработчика.

**Исходные данные**

1. Спектр сигнала 20 Гц – 4 кГц
2. Uпит = 5 ± 10% В
3. Время записи – 60 минут
4. Использование чувствительного микрофона
5. Устройство портативное
6. Воспроизведение через встроенный динамик

**Перечень конструкторских документов.**

Пояснительная записка, структурная схема, функциональная схе­ма, схема электрическая принципиальная, спецификация, перечень элементов.

**Цель курсовой работы.**

спроектировать устройство для записи человеческого голоса в течение 60 минут и последующего воспроизведения записанной информации.

Характеристики устройства: автономное устройство с батарейным питанием, временем записи 60 минут. Обеспечивает долговременное хранение записанной информации при выключении питания устройства. Имеется индика­ция режимов работы цифрового диктофона, а также органы управления режимами устройства.

Отличительной особенностью цифрового диктофона является повышенная надежность и удобство в эксплуатации.

**ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ**

АЦП –аналого-цифровой преобразователь;

ЦАП –цифро-аналоговый преобразователь;

АТС – автоматическая телефонная станция;

МГц- мегагерцы;

CPU - Central Processor Unit (процессор);

TTL - транзисторно-транзисторная логика;

мкм - микроны;

мм - миллиметры;

мс — микросекунды;

ОЗУ- оперативное запоминающее устройство;

ПЗУ- постоянное запоминающее устройство;

ПП - печатная плата;

### ВВЕДЕНИЕ

В настоящее время огромное значение придается различной информации, в том числе и голосовой. Возникает проблема в удобных способах хранения этой информации. Для этого применяют различные звукозаписывающие устройства. Принцип построения этих устройств различен - от записи на восковой носитель первого фонографа, до преобразования звуковой информации в цифровой код, с последующим её хранением в виде цифровых кодов. Самым удобным средством хранения звуковой информации является диктофон. Профессий, для которых диктофон является необходимым устройством - множество. Это и журналисты, телевизионщики, корреспонденты, органы охраны правопорядка.

Одной из проблем, которая возникает перед отечественным пользователем - это выбор лучшей по соотношению качество/цена модели. Неизбалованные отечественные пользо­ватели часто не придают особого значения устройству и техническим характеристикам диктофонов. Мало кто сознает, насколько проще и лег­че может стать работа с этим устройством, если серьезно отнестись к его выбору. Диктофоны выпускаются многими ком­паниями. К числу наиболее известных фирм-производителей относятся: Panasonic, I-River, Sony, LG и многие другие.

Отличительной особенностью большинства моделей диктофонов является то, что носителем информации является магнитная лента. А она, как известно, подвержена старению, в результате чего после нескольких десятков использований резко ухудшаются её качества. К тому же, велика вероятность «потерять» записанную информацию вследствие влияния сильных магнитных полей. Также отличительной особенностью диктофонов является наличие сложного механизма со множеством движущихся частей, которые от интенсивного использования быстро приходят в негодность.

sitednl.narod.ru/1.zip - база сотовых по Петербургу

Современные диктофоны в качестве носителей информации используют цифровые модули памяти. Ресурс модулей памяти даже при самом интенсивном использовании составит не менее нескольких лет. В таких моделях отсутствует сложный лентопротяжный механизм.

Исходя из вышесказанного проектирование новых типов цифровых диктофонов является важной задачей ввиду все более нарастающей по­требности удобства хранения звуковой информации.

**АНАЛИЗ ИСХОДНЫХ ДАННЫХ**

Необходимо разработать автономное устройство с батарейным питанием, которое бы записывало голосовую информацию в течении 60 минут. Устройство должно обеспечивать долговременное хранение записанной информации при выключении питания устройства. Необходимо предусмотреть индика­цию режимов работы цифрового диктофона, а также органы управления режимами устройства.

Так как информация должна вводиться и преобразовываться для хранения в цифровой вид, то необходимо применить АЦП и оцифровывать сигнал по методу импульсно-кодовой модуляции, суть которой заключается в следующем. Если на рис.1 представлен исходный сигнал:

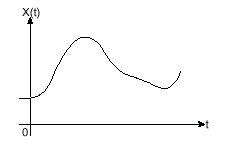


Рис.1. Исходный сигнал.

То для преобразования мы должны его разделить на временные отрезки

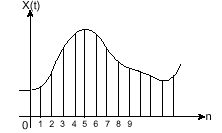


Рис.2. Дискретный сигнал.

Затем мы его должны квантировать по уровням (рис.3).

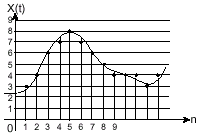


Рис.3. Квантированный сигнал.

Наконец мы можем получить кодовые отсчеты сигнала (рис.4).

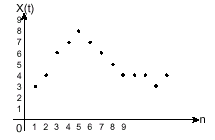


Рис.4. Оцифрованный сигнал.

Для определения длительности временных отрезков необходимо воспользоваться теоремой Котельникова, по которой для точного восстановления периодического сигнала необходимо взять минимум два отсчета за период. Таким образом, при верхней частоте спектра человеческого голоса - 4 кГц, частота дискретизации - 8 кГц.

Для квантования по уровню можно принять 256 уровней для обеспечения хорошего качества – в современной аппаратуре для обработки и передачи голосовой информации используются чаще всего именно 256 уровней.

Из этого следует, что для построения устройства необходимо использовать восьмибитовый АЦП, работающий на частоте 8 кГц.

Так как хранение информации должно производиться при выключенном питании, то в качестве устройства хранения нужно применить энергонезависимую память.

Объем памяти рассчитывается следующим образом: Так как каждую секунду записывается 8000 отсчетов по 8 бит, что составляет 8 кб, то в течение часа нам необходимо будет записать 3600 Х 8 кб, что составит примерно 29 Мб. Таким образом, применив память емкостью 32Мб, мы обеспечим нужное время записи. При использовании алгоритмов архивации, объем записываемой информации при необходимости можно увеличить.

**ВЫБОР СТРУКТУРЫ УСТРОЙСТВА**

Устройство работает с аналоговым сигналом, и, соответственно, чтобы обеспечить сопряжение с цифровой частью устройства применяется блок АЦП и блок входных цепей, которые обеспечивают усиление аналогового сигнала до необходимого уровня.

Устройство производит вывод записанной информации посредством блока ЦАП и блока выходных цепей, которые производят усиление выходного сигнала до необходимого уровня.

Контроль за работой блока ЦАП и блока АЦП производится модулем контроллера записи/чтения.

Индикацию режимов работы и управление ими диктофона выполняет блок индика­ции и управления.

Хранение записанной информации происходит в энергонезависимой памяти.

Таким образом в устройстве можно выделить следующие блоки:

- входных цепей;

* выходных цепей;
* контроллер записи/чтения;
* блок питания;
* тактового генератора;
* энергонезависимой памяти;

- блок индикации и управления.

Блок входных цепей соединяется информационной связью с блоком АЦП, который в свою очередь соединяется информационной связью с блоком энергонезависимой памяти и управляющими связями с тактовым генератором и блоком контроллера записи/чтения.

Блок выходных цепей соединяется информационной связью с блоком ЦАП, который в свою очередь соединяется информационной связью с блоком энергонезависимой памяти и управляющими связями с тактовым генератором и блоком контроллера записи/чтения.

Контроллер записи/чтения соединяется управляющей связью с энергонезависимой памятью.

Блок питания соединяется со всеми блоками.

Блок входных цепей обеспечивает усиление входного сигнала от микрофона и ограничение верхней частоты входного сигнала до 4 кГц. Усиленный сигнал поступает на АЦП где преобразуется по сигналам от тактового генератора в кодовые отсчеты по уровню, представленные в двоичном коде. Кодовые отсчеты поступают в энергонезависимую память, где посредством контроллера записи/чтения происходит их запись. Контроллер записи/чтения формирует необходимые сигналы для записи и чтения из энергонезависимой памяти. В свою очередь, он получает управляющие сигналы от блока индикации и управления.

При воспроизведении голоса происходит выборка кодовых отсчетов из энергонезависимой памяти и подача их на ЦАП, где и происходит их преобразование в аналоговый сигнал.

Блок выходных цепей обеспечивает усиление выходного сигнала и ограничение верхней частоты выходного сигнала до 4 кГц, для того чтобы избавиться от высокочастотных гармоник в выходном сигнале, появляющихся при квантовании.

Блок питания необходим для обеспечения питания всех блоков диктофона.

ВЫБОР ЭЛЕМЕНТОВ

В качестве устройства управления выбран микроконтроллер RISC-архитектуры серии AVR фирмы Atmel AT90S8535. Он обладает встроенной памятью программ объемом 4096 слов и памятью данных 512 байт. Любая его команда выполняется за 1 такт процес­сора. Тактовая частота 8 МГц .

На выполнение процессором программного кода для обработки и записи отсчетов, полученных от АЦП, потребуется до 20 мс, так что выбранный процессор вполне удовлетворяет требованию скорости работы и успевает обработать всю необходимую информацию.

Обеспечение протокола работы с памятью организуется тем же процессором программно - аппаратными методами, так как в микроконтроллере имеется аппаратная поддержка протокола SPI.

Для хранения записываемой информации выбрана FLASH ПЗУ AT45DB32 фирмы Atmel, объемом 32Мб.

**AT90S8535** **Description**

The AT90S8535 is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S8535 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

**Features**

• AVR® – High-performance and Low-power RISC Architecture

– 118 Powerful Instructions – Most Single Clock Cycle Execution

– 32 x 8 General-purpose Working Registers

– Up to 8 MIPS Throughput at 8 MHz

• Data and Nonvolatile Program Memories

– 8K Bytes of In-System Programmable Flash SPI Serial Interface for In-System Programming Endurance: 1,000 Write/Erase Cycles

– 512 Bytes EEPROM Endurance: 100,000 Write/Erase Cycles

– 512 Bytes Internal SRAM

– Programming Lock for Software Security

• Peripheral Features

– 8-channel, 10-bit ADC

– Programmable UART

– Master/Slave SPI Serial Interface

– Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode

– One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Modes and Dual 8-, 9-, or 10-bit PWM

– Programmable Watchdog Timer with On-chip Oscillator

– On-chip Analog Comparator

• Special Microcontroller Features

– Power-on Reset Circuit

– Real-time Clock (RTC) with Separate Oscillator and Counter Mode

– External and Internal Interrupt Sources

– Three Sleep Modes: Idle, Power Save and Power-down

• Power Consumption at 4 MHz, 3V, 20°C

– Active: 6.4 mA

– Idle Mode: 1.9 mA

– Power-down Mode: <1 µA

• I/O and Packages

– 32 Programmable I/O Lines

– 40-lead PDIP, 44-lead PLCC, 44-lead TQFP, and 44-pad MLF

• Operating Voltages

– VCC: 4.0 - 6.0V AT90S8535

– VCC: 2.7 - 6.0V AT90LS8535

• Speed Grades:

– 0 - 8 MHz for the AT90S8535

– 0 - 4 MHz for the AT90LS8535

**Pin Descriptions**

**VCC** Digital supply voltage.

**GND** Digital ground.

**Port A (PA7..PA0)** Port A is an 8-bit bi-directional I/O port. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers can sink 20 mA and can drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. Port A also serves as the analog inputs to the A/D Converter. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is

not running.

**Port B (PB7..PB0)** Port B is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port B output buffers can sink 20 mA. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. Port B also serves the functions of various special features of the AT90S8535 as listed on page 74. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

**Port C (PC7..PC0)** Port C is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port C output buffers can sink 20 mA. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. Two Port C pins can alternatively be used as oscillator for Timer/Counter2. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

**Port D (PD7..PD0)** Port D is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. Port D also serves the functions of various special features of the AT90S8535 as listed on page 83. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

**RESET** Reset input. An external reset is generated by a low level on the RESET pin. Reset pulses longer than 50 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

**XTAL1** Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

**XTAL2** Output from the inverting oscillator amplifier.

**AVCC** AVCC is the supply voltage pin for Port A and the A/D Converter. If the ADC is not used, this pin must be connected to VCC. If the ADC is used, this pin must be connected to VCC via a low-pass filter. See page 65 for details on operation of the ADC.

**AREF** AREF is the analog reference input for the A/D Converter. For ADC operations, a voltage in the range 2V to AVCC must be applied to this pin.

**AGND** Analog ground. If the board has a separate analog ground plane, this pin should be connected to this ground plane. Otherwise, connect to GND.

**AT45DB32 Description**

The AT45DB321 is a serial-interface Flash memory suitable for in-system reprogramming. Its 34,603,008 bits of memory are organized as 8192 pages of 528 bytes each. In addition to the main memory, the AT45DB321 also contains two SRAM data buffers of 528 bytes each. The buffers allow receiving of data while a page in the main memory is being reprogrammed

**Features**

• Serial-interface Architecture

• Page Program Operation

– Single Cycle Reprogram (Erase and Program)

– 8192 Pages (528 Bytes/Page) Main Memory

• Optional Page and Block Erase Operations

• Two 528-byte SRAM Data Buffers – Allows Receiving of Data while Reprogramming of Nonvolatile Memory

• Internal Program and Control Timer

• Fast Page Program Time – 7 ms Typical

• 120 µs Typical Page to Buffer Transfer Time

• Low-power Dissipation

– 4 mA Active Read Current Typical

– 3 µA CMOS Standby Current Typical

• 13 MHz Max Clock Frequency

• Hardware Data Protection Feature

• Serial Peripheral Interface (SPI) Compatible – Modes 0 and 3

• CMOS and TTL Compatible Inputs and Outputs

• Commercial and Industrial Temperature Ranges

**Pin Name Function**

**CS** Chip Select

**SCK** Serial Clock

**SI** Serial Input

**SO** Serial Output

**WP** Hardware Page Write Protect Pin

**RESET** Chip Reset

**RDY/BUSY** Ready/Busy

**ПРИНЦИП ДЕЙСТВИЯ**

Устройство состоит из следующих элементов: контроллер на базе микропроцессора, выполняющий функции управле­ния устройством. В состав MC также входят АЦП и ЦАП, поэтому микроконтроллер также выполняет функции оцифровывания аналогового сигнала и преобразования цифровых кодов в аналоговый сигнал. Усилители U1 и U2 предназначены для усиления аналоговых сигналов и ограничения верхних граничных частот этих сигналов до 4 кГц. Микрофон и динамик предназначены соответственно для ввода и воспроизведения голосовой информации. Генератор опорного напряжения U3 формирует опорное напряжение для встроенного АЦП микроконтроллера MC. Тактовый генератор G предназначен для тактирования всех внутренних схем микроконтроллера. Энергонезависимая память RAM предназначена для хранения всей записываемой голосовой информации. Блок клавиатуры KBD предназначен для управления режимами устройства. Блок индикации LED предназначен для индикации режимов работы устройства.

Входной сигнал с микрофона поступает на усилитель-фильтр U1, где он усиливается до размаха в несколько вольт и ограничивается верхней частотой до 4 кГц. С выхода усилителя-фильтра U1 сигнал поступает на вход встроенного АЦП AD0. Обработанный сигнал программно записывается в RAM по линиям, предусмотренным протоколом SPI (рис.5).

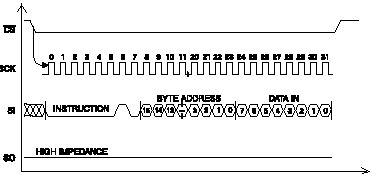


Рис.5. Запись в память.

После окончания записи сигнал может быть прочитан из памяти микроконтроллером (рис.6).

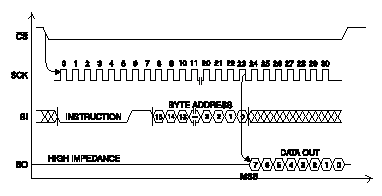


Рис.6. Чтение из памяти.

Микропроцессор MС выдает через встроенный ЦАП сигналы на усилитель-фильтр U2, где они усиливаются и ограничиваются по верхней частоте до 4 кГц. В качестве ЦАП в микроконтроллере работает встроенный широтно-импульсный модулятор, который с помощью дифференцирующих сигналов и интегрирующей цепи может восстановить форму исходного сигнала.



Рис.7. Восстановление сигнала при помощи широтно-импульсной модуляции.

Отображение режимов работы устройства производят светоизлучающие диоды LED при высоком уровне сигнала на соответствующем выходе микроконтроллера.

В этом примере окончательный сигнал состоит из 8 бит. Теперь он может быть сохранён в DataFlash. DataFlash не требует отдельного цикла стирания перед программированием. При использовании команд «Буфер в основную страницу памяти программы с встроенным стиранием» и «Основная страница памяти программы сквозь буфер», DataFlash будет автоматически стирать определённую страницу в массиве памяти перед программированием действительных данных. Если система требует большую программную пропускную способность (больше 200K bps), то области массива основной памяти могут быть предварительно очищены, для уменьшения суммарного программного времени. При предварительной очистке части главного массива памяти, для уменьшения общего времени, может использоваться команда «Буфер в основную страницу памяти программы без встроенного стирания».

После очистки памяти данные могут записываться до тех пор, пока не заполнятся все страницы.

Для записи в DataFlash используется буфер 1. Когда этот буфер заполнится (528 выборками), он записывается в память во время 529 преобразования. Данные записываются до тех пор, пока нажата кнопка «Запись» или память не заполнилась. Если вся память заполнена, то новые данные не могут быть записаны, пока не очищена DataFlash. Если память заполнена лишь частично, то при повторном нажатии кнопки «Запись» новые данные будут добавлены сразу за уже записанными данными.

Воспроизведение звука всегда начинается с начала DataFlash. Оно прекращается, если все записанные данные воспроизведены или когда кнопка «Воспроизведение» отпущена.

DataFlash позволяет проигрывать данные либо напрямую из основной страницы памяти, либо путём копирования страницы в один из двух буферов и последующим чтением из буфера. Метод прямого доступа не подходит для этого примера, так как это метод двухадресный (один адрес для страницы, другой для позиции байта), и, следовательно, в DataFlash должна быть отправлена длинная загрузочная последовательность для каждого отдельного байта. Это занимает больше одного цикла ШИМ, который длится 510 тактовых импульсов для 8-битного ШИМ сигнала.

Поэтому, одна страница памяти копируется в один из двух буферов. Пока данные читаются из этого буфера, следующая страница памяти копируется в другой буфер. Когда все данные считаны из первого буфера, чтение продолжается из другого буфера, в это время первый буфер перезагружается. Чтение данных из буфера DataFlash синхронизируется частотой ШИМ.

Выходной фильтр сглаживает выходной сигнал и удаляет высокочастотную несущую ШИМ сигнала. Результирующий выходной сигнал для сигнала из примера похож на тот, что изображён на рисунке 8. Если исключить ошибку квантования и отсутствующее усиление, то сигнал полностью похож на входной аналоговый сигнал (Рисунок 1).

Рис 8. Выходной ШИМ сигнал



Пользователь может управлять устройством через три кнопки: «Очистка», «Запись» и «Воспроизведение». Если кнопки не нажаты, то внутренний нагрузочный резистор обеспечивает VCC на PС4…PС6. Нажатие кнопки переключает входную линию на GND.

В качестве обратной связи для пользователя выступает светодиод, отображающий состояние системы.

Кварцевый резонатор с двумя развязывающими конденсаторами (22 пФ) генерирует системные тактовые импульсы.

Микрофонный усилитель является простым инвертирующим усилителем. Коэффициент усиления устанавливается через R1 и R9 (коэффициент усиления = R1/R9). R4 предназначен для питания микрофона, а C1 блокирует любые DC составляющие на входе усилителя. R2 и R3 устанавливают смещение. R5 и С8 формируют простой фильтр нижних частот первого порядка. Также R5 защищает усилитель от любых повреждений, если выходная цепь закорочена.

C3 блокирует любую DC составляющую на входе динамика.

**Настройка**

Когда программа запущена, порты должны быть настроены. Это делается в подпрограмме «setup» (установка).

Протокол SPI определяет одно устройство как «ведущее», а другие устройства, подключенные к «ведущему», как «ведомые». В данном примере, микроконтроллер AVR выступает в роли «ведущего», а DataFlash в роли «ведомого».

SPI интерфейс AT90S8535 определён как альтернативная функция PortB (PB0…PB4). В данном примере, управляющие сигналы для DataFlash являются также настроечными на PortB (PB0…PB2 и PB4). Для установок «ведущего», сигналы Serial Clock(SCK), Master Out/Slave In(MOSI), Chip Select(#CS), Write Protect(#WP) и Reset(#RST) являются выходами, тогда как Master In/Slave Out(MISO) и Ready/Busy(RDY/#BSY) являются входами. Состоянием по умолчанию PortB является: всех выходы в высоком состоянии, а на всех входах - внутренние нагрузочные резисторы.

АЦП AT90S8535 подключено к PortA. Поэтому PortA определён как вход в высокоимпедансном состоянии.

PortС служит в качестве входа для кнопок.

.

**ОПИСАНИЕ ПРИНЦИПИАЛЬНОЙ СХЕМЫ**

###### УСТРОЙСТВА

Микропроцессор MCU (AT90S8535) управляет через порт В работой FLASH - памяти DD1 Так как в процессоре имеется аппаратная поддержка SPI протокола, то микросхема памяти, работающая по этому протоколу непосредственно подключается к выводам микроконтроллера DD2 miso, mosi, sck.

В режиме хранения информации микроконтроллер и микросхема памяти переключаются в спящий режим - микроконтроллер путем выполнения специальной команды, а микросхема памяти путем удержания микроконтроллером сигнала #cs в высоком уровне.

Микросхема памяти и микропроцессор выполнены по КМОП - технологии, что позволяет их непосредственно питать от аккумуляторных батарей небольшой емкости, которые подключаются через разъем Х1.

Для обеспечения работы процессора на частоте 8 МГц примене­ны элементы BQ1 - кварцевый резонатор и конденсаторы С8 и С9.

Опорное напряжение для АЦП берется напрямую с шины питания. Изменение его значения не приведет к структурному искажению сигнала, а только к его масштабированию.

Сигнал от микрофона усиливается микросхемой DA1 и подается на вход АЦП. Усилитель DA1 также охвачен частотно- зависимой отрицательной обратной связью, для того чтобы ограничить верхнюю частотную границу входного сигнала до 4 кГц.

К порту С микроконтроллера подключаются кнопки управления и светодиоды для индикации режимов работы устройства.

К разъему Х1 подключается аккумуляторная батарея для питания устройства. Конденсаторы С1 и С2 служат для сглаживания пульсаций по цепи питания, которые могут возникнуть при протекании динамических процессов внутри микросхем.

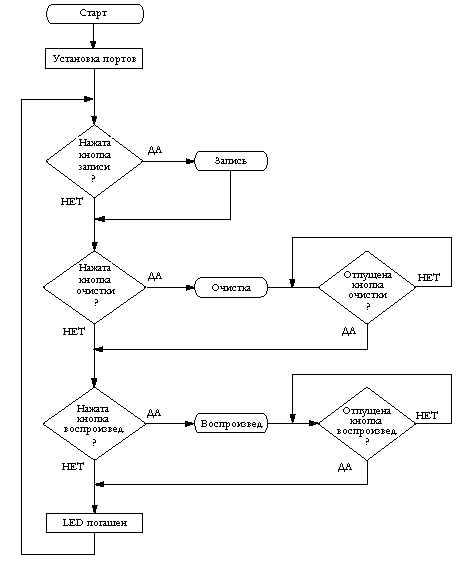
Главный цикл

В главном цикле, отслеживается состояние всех трёх кнопок. Если одна из них нажата, то LED загорается и показывает, что система занята, и вызывается соответствующая подпрограмма.

Дополнительный цикл выполняется до тех пор, пока кнопка нажата, в качестве программной противодребезговой защиты для функций «Очистка» и «Воспроизведение».

Во время главного цикла, LED погашен, это значит, что система работает в холостом режиме.

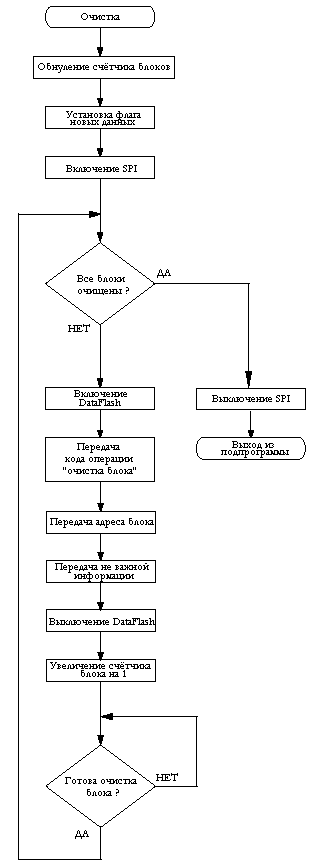
Рисунок 11. Главный цикл



**Очистка**

Data Flash может быть предварительно очищена.

Рисунок 12. Очистка



При вызове подпрограммы «erase» (очистка), устанавливается флаг, который показывает, что в следующем цикле записи новые данные могут быть сохранены в начале DataFlash.

SPI должен быть установлен для доступа к DataFlash. Здесь не используются прерывания. Порядок данных для DataFlash следующий: MIB является первым, а AT90S8535 - «ведущим».

DataFlash принимает либо сигнал SCK, который находится в низком состоянии, когда #CS переключается из высокого в низкое состояние (SPI режим 0), либо сигнал SCK, который находится в высоком состоянии, когда #CS переключается из низкого в высокое состояние (SPI режим 3), во время положительной фазы тактовых импульсов. В данном примере SPI установлен в режим 3. Для того чтобы получить наибольшую скорость передачи данных, выбирается наименьшее деление тактовой частоты, шина SPI запускается на частоте 2 МГц, если используется кварцевый генератор с частотой 8 МГц.

Для выполнения очистки блока, линия #CS переводится в низкое состояние и в DataFlash, следом за двумя зарезервированными битами (нулями), загружается код операции 0x50, затем 9-разрядный адрес блока и 13 не имеющих значения бит. Эта последовательность передаётся побайтно «ведомому». После каждого байта, регистр состояния SPI (SPSR) проверяется до тех пор, пока флаг прерываний SPI не покажет, что передача завершена. После записи всей последовательности, сразу после перевода линии #CS в высокое состояние, начинается очистка блока. Вывод Ready/Busy переводится памятью DataFlash в низкое состояние, до тех пор, пока блок не очиститься. Затем следующий блок будет очищен тем же самым способом, что и текущий. Очистка будет продолжаться, пока все 512 блоков не очистятся. Очищенные зоны читаются как 0xFF.

**Запись**

Подпрограмма записи состоит из установки АЦП и пустого цикла, который продолжается пока нажата кнопка «Запись». В данном примере используется вывод ADC0, для которого требуется, чтобы регистр выбора мультиплексора АЦП (ADMUX) был установлен в нуль. В регистре управления и состояния АЦП (ADCSR) разрешается работа с коэффициентом деления тактовой частоты 32, устанавливается режим одиночного преобразования, разрешаются прерывания, а также сбрасываются флаги прерываний. Аналого-цифровое преобразование начинается сразу. Первое преобразование занимает больше времени, чем последующие преобразования (832 тактовых импульса вместо 448). После этого времени, возникает прерывание АЦП, показывающее, что преобразование закончено, и результат может быть прочитан из регистра данных АЦП.

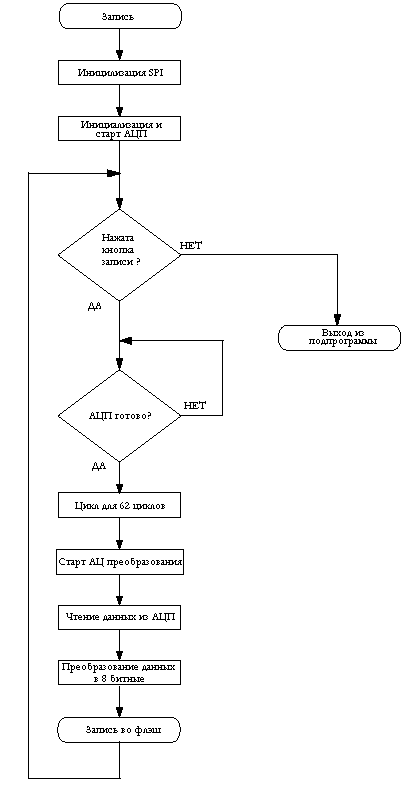
Аналоговый сигнал из цепи микрофона выбирается на частоте 15.686 Гц. Это та же самая частота, что и выходная (ШИМ) частота.

Для достижения частоты выборки 15.686 Гц, выборка должна происходить каждые 510 циклов (15.686 Гц x 510 = 8 МГц). Для получения одного результата АЦП, нужно каждые 510 циклов запускать АЦП в режиме одиночного преобразования с коэффициентом деления частоты 32. Одиночное преобразование занимает 14 циклов АЦП. Поэтому преобразование будет готово после 14 x 32 = 448 циклов.

Когда преобразование закончено, возникает прерывание. Процедура прерывания выполняет цикл для заполнения пустых 62 циклов (510–448), перед началом нового преобразования.

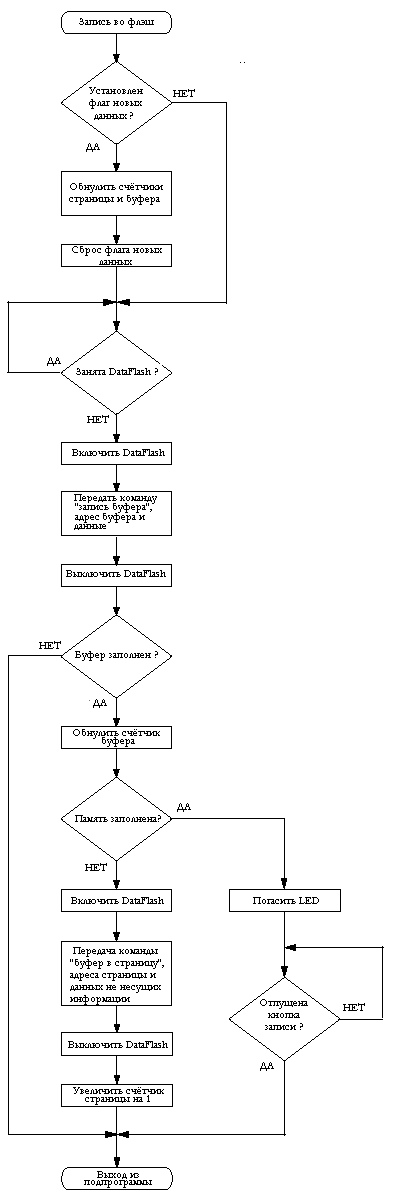
Результатом 10-разрядного преобразования является величина на входе АЦП, которая появляется через 2 цикла после начала преобразования. Эти 10 бит перекрывают диапазон от AGND до AREF (в данном примере от 0 до 5В). Выходной сигнал цепи микрофона ограничен диапазоном 2.3В…3.5В. Поэтому из результата 10-разрядного преобразования вычитается минимальное входное напряжение. Это 0x1D5 для 2.3В. Часть данных, представляющих сигнал величиной выше 3.5В, убирается путём удаления двух MSB. Это делается автоматически, когда результат преобразования передаётся в подпрограмму «запись во флэш», так как эти переменные «flash\_data» определяются типом «char» (8-бит). Последние 8-бит данных должны быть записаны в DataFlash перед следующим прерыванием преобразования.

Рисунок 13. Запись



**Запись данных в DataFlash**

Рисунок 14. Запись в DataFlash



Запись данных в DataFlash производится путём записи сначала в буфер, а, когда этот буфер будет заполнен, его содержимое запишется в одну страницу главной памяти.

В подпрограмме «write\_to\_flash» переменная «j» соответствует номеру байта в буфере, а переменная «k» номеру страницы, в которую будет записываться содержимое буфера. Если флаг новых данных показывает, что DataFlash пуста, то оба счётчика устанавливаются в нуль.

Если память уже содержит некоторые данные, то переменные показывают следующее свободное место в памяти, и гарантируют, что новые данные добавятся к содержимому памяти.

Для того чтобы защитить содержимое этих переменных при двух вызовах функций, они объявляются статическими переменными.

Для записи данных в буфер, линия #CS переводится в низкое состояние и в DataFlash загружается операционный код 0x84. Это следует за 14 не имеющими смысла битами и 10-битовым адресом положения внутри буфера. Затем вводятся 8-бит данных.

Эта последовательность передаётся «ведомому» побайтно. После каждого байта проверяется регистр состояния SPI (SPSR), пока флаг прерывания SPI не покажет, что последовательная передача завершена. После записи всей последовательности линия #CS переводится в высокое состояние.

Если буфер заполнен и остались пустые страницы, то буфер копируется на следующую страницу DataFlash. Так как память была очищена раньше, то данные могут быть записаны без дополнительного стирания.

Если память заполнена, то цикл выполняется, пока нажата кнопка «запись». Любые данные, записанные в то время, когда память уже заполнена, будут потеряны.

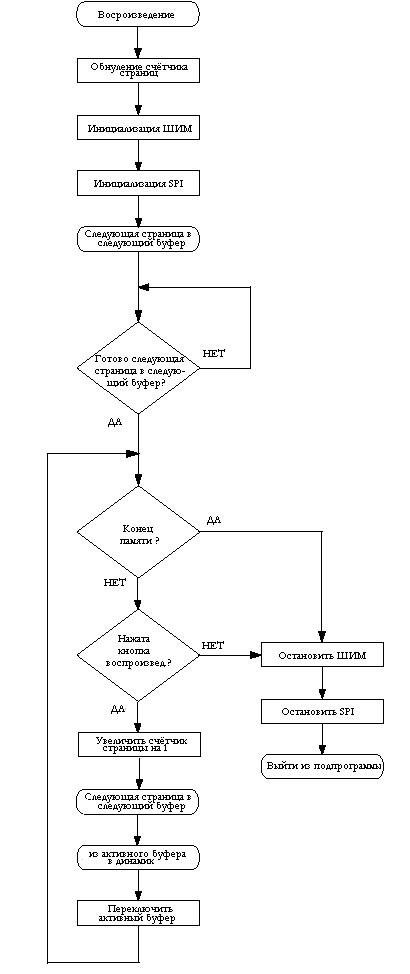
**Воспроизведение**

В процедуре «воспроизведения», содержимое DataFlash считывается и модулируется как 8-разрядная ШИМ на частоте 15.686 Гц. Для достижения большей скорости, данные не читаются напрямую из основной памяти, а передаются в один из двух буферов и затем читаются из буфера. В это время копируется следующая страница памяти в другой буфер. Для ШИМ, 16-разрядный Таймер/Счётчик 1 используется с выходом ШИМ на OC1B. Это описывается в регистре управления Таймера/Счётчика A и B (TCCRA/TCCRB). Для запуска ШИМ с возможной наибольшей частотой, делитель тактовой частоты ШИМ устанавливается в 1.

Когда установка завершена, первая страница копируется в буфер 1, посредством перевода линии #CS в низкое состояние и передачей соответствующих команд в DataFlash. Передача страницы в буфер начинается, когда линия #CS переводится снова в высокое состояние. Когда состояние на выводе Ready/Busy меняется памятью DataFlash на высокое, то это означает, что буфер 1 содержит действительные данные. Затем начинается передача следующей страницы в буфер 2. Так как оба буфера независимы друг от друга, то данные могут всегда читаться из буфера 1, пока DataFlash остаётся занятой копированием данных из второй страницы в буфер 2.

Для чтения байта из буфера, в DataFlash должна быть записана фиктивная величина. Операция записи «ведущего» в SPI «ведомого» приводит к тому, что содержимое его регистра данных SPI (SPDR) будет изменено. После записи фиктивного байта в DataFlash, регистр SPDR микроконтроллера AVR содержит выходные данные из DataFlash.

Рисунок 15. Воспроизведение



Когда значения ШИМ счётчика равно «0», Таймер 1 вызывает прерывание переполнения. Это прерывание используется для синхронизации выходных данных из DataFlash частотой ШИМ. Когда значение из буфера сдвигается в микроконтроллер AVR, цикл выполняется до тех пор, пока Таймер 1 не вызовет прерывание переполнения. Затем данные записываются в выходной регистр сравнения Таймера/Счётчика 1 B (OCR1B), автоматически защёлкивая выход ШИМ, когда счётчик ШИМ достигнет максимального значения (255 для 8-разрядной ШИМ).

После того как считается последнее значение из буфера, активный буфер переключится.

Если воспроизведена вся память, то все прерывания отключены и Таймер/Счётчик 1 остановлен.

Рисунок 16. Следующая страница в следующий буфер

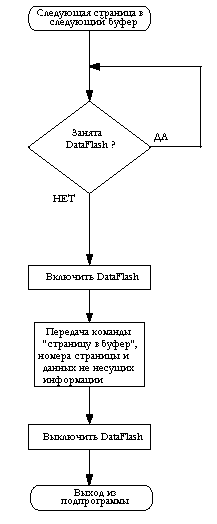
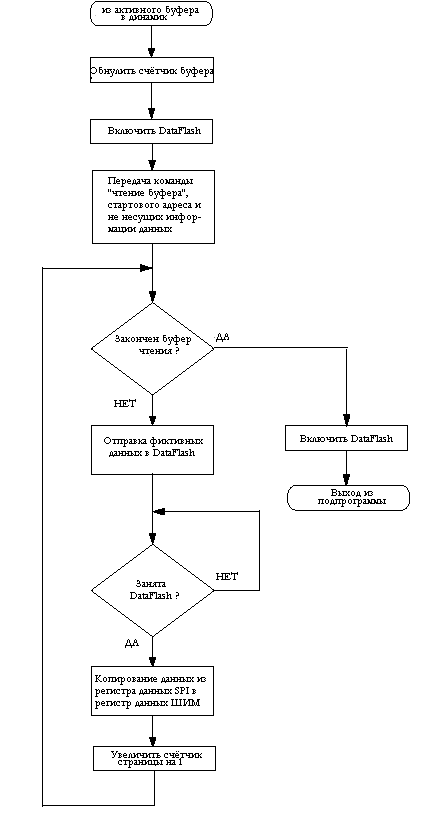


Рисунок 17. Активный буфер в динамик



**Изменение и оптимизация**

Сигнал с выхода микрофона может изменяться в зависимости от типа используемого микрофона. Для достижения лучших результатов важно выбрать такой коэффициент усиления микрофонного усилителя, который обеспечит максимальный сигнал, наиболее близкий к AREF.

Данные, записанные в DataFlash, полностью соответствуют данным, считанным с АЦП. В случае записи в течение большого промежутка времени или записи стерео сигнала может потребоваться упаковка этих данных.

В этом примере приведены два способа применения флага состояния.

Первый способ – использование глобальной переменной (т.е. переменная «wait» используется в подпрограмме «playback»). Второй способ – использование незадействованного бита в регистре. В подпрограмме «стирания», используется бит ACIS1 (регистра управления и состояния аналогового компаратора (ACSR))для отображения того, что следующими этапом должно быть сохранение новых данных.

Частота выборки равная 15.686 Гц (приблизительно 510 циклов), генерируется с помощью прерывания АЦП и цикла задержки. Она может быть заменена независимым таймером (Таймер/Счётчик 0 или Таймер/Счётчик 2), если он не используется для других целей.

# Пример программы на языке C

/\* Очистка всех страниц в случае необходимости. Запись данных в буфер 1. Если буфер заполнен,

то его содержимое записывается в страницу памяти. Чтение DataFlash через буфер 1 и буфер 2 в

регистр данных.

\*/

#include “io8535.h”

#include

#include “stdlib.h”

#include “dataflash.h”

// прототипы

void setup (void);

void erasing (void);

void recording (void);

void interrupt[ADC\_vect] sample\_ready (void);

void write\_to\_flash (unsigned char ad\_data);

void playback (void);

void next\_page\_to\_next\_buffer (unsigned char active\_buffer, unsigned int page\_counter);

void interrupt[TIMER1\_OVF1\_vect] out\_now(void);

void active\_buffer\_to\_speaker (unsigned char active\_buffer);

// глобальные переменные

volatile unsigned char wait = 0;

void setup(void)

{

DDRB = 0xBD; // Инициализация порта SPI

// SCK, MISO, MOSI, CS, LED, WP , RDYBSY, RST

// PB7, PB6, PB5, PB4, PB3, PB2 , PB1, PB0

// O I O O O O I O

// 1 0 1 1 1 1 0 1

PORTB = 0xFF; // все выходы в высоком состоянии, на входах

нагрузочные резисторы (LED погашен)

DDRA = 0x00; // Port A определяется как вход

PORTA = 0x00;

DDRD = 0x10; // Port D определяется как вход (D4: выход)

\_SEI(); // прерывания разрешены

}

void erasing(void)

{

unsigned int block\_counter = 0;

unsigned char temp = 0x80;

ACSR |= 0x02; // установка флага, показывающего, что следующим

этапом должна быть запись данных

// прерывания запрещены, порт SPI включён, «ведущий» режим, первый MSB, 3 режим SPI, Fcl/4

SPCR = 0x5C;

while (block\_counter < 512)

{

PORTB &= ~DF\_CHIP\_SELECT; // включение DataFlash

SPDR = BLOCK\_ERASE;

while (!(SPSR & temp)); // ожидание завершения передачи

SPDR = (char)(block\_counter>>3);

while (!(SPSR & temp)); // ожидание завершения передачи

SPDR = (char)(block\_counter<<5);

while (!(SPSR & temp)); // ожидание завершения передачи

SPDR = 0x00; // не важно

while (!(SPSR & temp)); // ожидание завершения передачи

PORTB |= DF\_CHIP\_SELECT; // выключение DataFlash

block\_counter++;

while(!(PINB & 0x02)); // ожидание очистки блока

}

SPCR = 0x00; //отключение SPI

}

void recording(void)

{

// прерывания запрещены, порт SPI включён, «ведущий» режим, первый MSB, 3 режим SPI, Fcl/4

SPCR = 0x5C;

ADMUX = 0x00; // номер входного вывода АЦП = 0

ADCSR = 0xDD; // одиночное АЦ преобразование, fCK/32, старт

преобразования

while (!(PIND & 8)); // цикл продолжается пока нажата кнопка записи

(кнопка 3)

ADCSR = 0x00; // выключение АЦП

SPCR = 0x00; // выключение SPI

}

void interrupt[ADC\_vect] sample\_ready(void)

{

unsigned char count = 0;

while (count < 6) count++; // ожидание в течение нескольких циклов

ADCSR |= 0x40; // старт нового АЦ преобразования

write\_to\_flash(ADC-0x1D5); // чтение данных, преобразование 8 бит и

сохранение во флэш

}

void write\_to\_flash(unsigned char flash\_data)

{

static unsigned int buffer\_counter;

static unsigned int page\_counter;

unsigned char temp = 0x80;

if((ACSR & 0x02)) // если флаг установлен, то новые данные должны быть

установлены

{

buffer\_counter = 0;

page\_counter = 0; // сброс счётчика если должны быть записаны

новые данные

ACSR &= 0xFD; // очистка флага сигнала

}

while(!(PINB & 0x02)); // проверка занятости флэша

PORTB &= ~DF\_CHIP\_SELECT; // включение DataFlash

SPDR = BUFFER\_1\_WRITE;

while (!(SPSR & temp)); // ожидание завершения передачи

SPDR = 0x00; // не важно

while (!(SPSR & temp)); // ожидание завершения передачи

SPDR = (char)(buffer\_counter>>8); // не важно + первые два бита буфера

адреса

while (!(SPSR & temp)); // ожидание завершения передачи

SPDR = (char)buffer\_counter; // буфер адреса (макс. 2^8 = 256

страниц)

while (!(SPSR & temp)); // ожидание завершения передачи

SPDR = flash\_data; // запись данных в регистр данных SPI

while (!(SPSR & temp)); // ожидание завершения передачи

PORTB |= DF\_CHIP\_SELECT; // выключение DataFlash

buffer\_counter++;

if (buffer\_counter > 528) // если буфер заполнен, то его

содержимое записывается в страницу памяти

{

buffer\_counter = 0;

if (page\_counter < 4096) // если память не заполнена

{

PORTB &= ~DF\_CHIP\_SELECT; // включить DataFlash

SPDR = B1\_TO\_MM\_PAGE\_PROG\_WITHOUT\_ERASE; // записать

данные из буфера 1 в страницу

while (!(SPSR & temp)); // ожидание завершения

передачи

SPDR = (char)(page\_counter>>6);

while (!(SPSR & temp)); // ожидание завершения

передачи

SPDR = (char)(page\_counter<<2);

while (!(SPSR & temp)); // ожидание завершения

передачи

SPDR = 0x00; // не важно

while (!(SPSR & temp)); // ожидание завершения

передачи

PORTB |= DF\_CHIP\_SELECT; // выключение DataFlash

page\_counter++;

}

else

{

PORTB |= 0x08; // погасить LED

while (!(PIND & 8)); // ждать пока кнопка записи

не отпущена (кнопка 3)

}

}

}

void playback(void)

{

unsigned int page\_counter = 0;

unsigned int buffer\_counter = 0;

unsigned char active\_buffer = 1; // активный буфер = буфер 1

unsigned char temp = 0x80;

TCCR1A = 0x21; // 8 бит ШИМ, используется COM1B

TCNT1 = 0x00; // обнуление счётчика 1

TIFR = 0x04; // сброс флага превышения счётчика 1

TIMSK = 0x04; // разрешение прерывания переполнения счётчика 1

TCCR1B = 0x01; // коэф. Пересчёта счётчика 1 = 1

OCR1B = 0x00; // обнуление выходного регистра сравнения B

// прерывания запрещены, порт SPI включён, «ведущий» режим, первый MSB, 3 режим SPI, Fcl/4

SPCR = 0x5C;

next\_page\_to\_next\_buffer (active\_buffer, page\_counter); // чтение страницы 0

в буфер 1

while (!(PINB & 0x02)); // ожидание завершения передачи данных из страницы 0

в буфер 1

while ((page\_counter < 4095)&(!(PIND & 2))) // пока кнопка воспроизведения

(кнопка 1) нажата

{

page\_counter++; // теперь берём следующую страницу

next\_page\_to\_next\_buffer (active\_buffer, page\_counter);

active\_buffer\_to\_speaker (active\_buffer);

if (active\_buffer == 1) // если буфер 1 является активным буфером

{

active\_buffer++; // то устанавливаем буфер 2 в качестве активного

}

else // иначе

{

active\_buffer--; // устанавливаем буфер 1 в качестве активного

}

}

TIMSK = 0x00; // запрещаем все прерывания

TCCR1B = 0x00; // останавливаем счётчик 1

SPCR = 0x00; // отключаем SPI

}

void next\_page\_to\_next\_buffer (unsigned char active\_buffer, unsigned int page\_counter)

{

unsigned char temp = 0x80;

while(!(PINB & 0x02)); // ждём, пока флэш не освободится

PORTB &= ~DF\_CHIP\_SELECT; // включаем DataFlash

if (active\_buffer == 1) // если буфер 1 активный

{

SPDR = MM\_PAGE\_TO\_B2\_XFER; // то передаём следующую страницу в

буфер 2

}

else // иначе

{

SPDR = MM\_PAGE\_TO\_B1\_XFER; // передаём следующую страницу в

буфер 1

}

while (!(SPSR & temp)); // ожидаем завершения передачи

SPDR = (char)(page\_counter >> 6);

while (!(SPSR & temp)); // ожидаем завершения передачи

SPDR = (char)(page\_counter << 2);

while (!(SPSR & temp)); // ожидаем завершения передачи

SPDR = 0x00; // записываем не имеющий значения байт

while (!(SPSR & temp)); // ожидаем завершения передачи

PORTB |= DF\_CHIP\_SELECT; // выключаем DataFlash и начинаем передачу

}

void interrupt[TIMER1\_OVF1\_vect] out\_now(void)

{

wait = 0; // возникновение прерывания

}

void active\_buffer\_to\_speaker (unsigned char active\_buffer)

{

// пока активный буфер не очистится воспроизводим его содержимое на динамике

unsigned int buffer\_counter = 0;

unsigned char temp = 0x80;

PORTB &= ~DF\_CHIP\_SELECT; // включение DataFlash

if (active\_buffer == 1) // если буфер 1 активный буфер

{

SPDR = BUFFER\_1\_READ; // то читаем из буфера 1

}

else // иначе

{

SPDR = BUFFER\_2\_READ; // читаем из буфера 2

}

while (!(SPSR & temp)); // ожидаем завершения передачи

SPDR = 0x00; // запись не имеющего значения байта

while (!(SPSR & temp)); // ожидаем завершения передачи

SPDR = 0x00; // запись не имеющего значения байта

while (!(SPSR & temp)); // ожидаем завершения передачи

SPDR = 0x00; // начать с адреса 0 буфера

while (!(SPSR & temp)); // ожидаем завершения передачи

SPDR = 0x00; // запись не имеющего значения байта

while (!(SPSR & temp)); // ожидаем завершения передачи

while (buffer\_counter < 528)

{

SPDR = 0xFF; // записываем фиктивное значение в начало

сдвигового регистра

while (!(SPSR & temp)); // ожидаем завершения передачи

while(wait); // ожидаем прерывание переполнения таймера 1

OCR1B = SPDR; // воспроизводим данные из сдвигового регистра

wait = 1; // сброс флага сигнала

buffer\_counter++;

}

PORTB |= DF\_CHIP\_SELECT; // выключение DataFlash

}

void main(void)

{

setup();

for(;;)

{

if (!(PIND & 8)) // если кнопка записи нажата (кнопка 3)

{

PORTB &= 0xF7; // зажигаем LED

recording();

}

if (!(PIND & 4)) // если нажата кнопка очистки (кнопка 2)

{

PORTB &= 0xF7; // зажигаем LED

erasing();

while (!(PIND & 4)); // ждём пока кнопка очистки не

отпущена (кнопка 2)

}

if (!(PIND & 2)) //если нажата кнопка воспроизведения(кнопка 1)

{

PORTB &= 0xF7; // зажигаем LED

playback();

while (!(PIND & 2)); // ждём пока кнопка воспроизведения

не отпущена (кнопка 1)

}

PORTB |= 0x08; // гасим LED во время «холостой» работы

}

}

DataFlash.h

// изменён 19.04.1999

// для использования 8535

#include “ina90.h”

#pragma language=extended

// DataFlash вывод сброса порта (PB 0)

#define DF\_RESET 0x01

// DataFlash вывод состояния порта готов/занят (PB 1)

#define DF\_RDY\_BUSY 0x02

// DataFlash защита от записи загрузочного сектора (PB 2)

#define DF\_WRITE\_PROTECT 0x04

// DataFlash вывод порта выбора микросхемы (PB 4)

#define DF\_CHIP\_SELECT 0x10

// буфер 1

#define BUFFER\_1 0x00

// буфер 2

#define BUFFER\_2 0x01

// определение всех кодов операций

// запись буфера 1

#define BUFFER\_1\_WRITE 0x84

// запись буфера 2

#define BUFFER\_2\_WRITE 0x87

// чтение буфера 1

#define BUFFER\_1\_READ 0x54

// чтение буфера 2

#define BUFFER\_2\_READ 0x56

// Буфер 1 в основную страницу памяти программы с встроенным стиранием

#define B1\_TO\_MM\_PAGE\_PROG\_WITH\_ERASE 0x83

// Буфер 2 в основную страницу памяти программы с встроенным стиранием

#define B2\_TO\_MM\_PAGE\_PROG\_WITH\_ERASE 0x86

// Буфер 1 в основную страницу памяти программы без встроенного стирания

#define B1\_TO\_MM\_PAGE\_PROG\_WITHOUT\_ERASE 0x88

// Буфер 2 в основную страницу памяти программы без встроенного стирания

#define B2\_TO\_MM\_PAGE\_PROG\_WITHOUT\_ERASE 0x89

// Основная страница памяти программы сквозь буфер 1

#define MM\_PAGE\_PROG\_THROUGH\_B1 0x82

// Основная страница памяти программы сквозь буфер 2

#define MM\_PAGE\_PROG\_THROUGH\_B2 0x85

// автоматическая перезапись страницы через буфер 1

#define AUTO\_PAGE\_REWRITE\_THROUGH\_B1 0x58

// автоматическая перезапись страницы через буфер 2

#define AUTO\_PAGE\_REWRITE\_THROUGH\_B2 0x59

// сравнение основной страницы памяти с буфером 1

#define MM\_PAGE\_TO\_B1\_COMP 0x60

// сравнение основной страницы памяти с буфером 2

#define MM\_PAGE\_TO\_B2\_COMP 0x61

// передача основной страницы памяти в буфер 1

#define MM\_PAGE\_TO\_B1\_XFER 0x53

// передача основной страницы памяти в буфер 2

#define MM\_PAGE\_TO\_B2\_XFER 0x55

// регистр состояния DataFlash для чтения плотности, сравнения состояний,

// и состояния готов/занят

#define STATUS\_REGISTER 0x57

// чтение основной страницы памяти

#define MAIN\_MEMORY\_PAGE\_READ 0x52

// очистка 528 байт страницы

#define PAGE\_ERASE 0x81

// очистка 512 страниц

#define BLOCK\_ERASE 0x50

#define TRUE 0xff

#define FALSE 0x00

**РАСЧЕТ ВРЕМЕНИ НАРАБОТКИ НА ОТКАЗ**

Интенсивность отказов устройства, содержащего разнотипные элементы, определяется следующим соотношением:



Среднее время наработки на отказ устройства составляет:

Tср = 1/λ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Тип элемента | Число элементов Q | λ | Q\*λ | |
| 1 | К1401УД2  AT90S8535-8PI  AT45DB32 | 2  1  1 | 0.017\*10-6  0.017\*10-6  0.017\*10-6 | 0.034\*10-6  0.017\*10-6  0.017\*10-6 | |
| 2 | Паяные соединения | 210 | 10-9 | 0.21\*10-6 | |
| 3 | Конденсаторы | 9 | 0.004\*10-6 | 0.012\*10-6 | |
| 4 | Резисторы МЛТ | 15 | 0.02\*10-6 | 0.30\*10-6 | |
| 5 | Диод | 4 | 0.02\*10-6 | 0.08\*10-6 | |
| 6 | Транзистор | 1 | 0.05\*10-6 | 0.05\*10-6 | |
| 7 | Вилка разъема | 1 | 0.011\*10-6 | 0.011\*10-6 | |
| 8 | Резонатор | 1 | 0.09\*10-6 | 0.09\*10-6 | |
| 9 | Микрофон | 1 | 10-5 | 10-5 | |
| 10 | Громкоговоритель | 1 | 10-5 | 10-5 | |
| 11 | Переключатель | 3 | 0.01\*10-5 | 0.03\*10-5 | |
| Суммарная интенсивность отказов | | | | | 2.851\*10-6 |

Перечень комплектующих элементов устройства и значений интенсивности их отказов:

Время наработки на отказ разработанного устройства при этом составит:

Тотк=106/1.691=350754 час

**ОЦЕНКА КОЛИЧЕСТВА ОШИБОК В ПРОГРАММЕ**

N=ΣajZj

Z1- сложность условных операторов If

Z2 – общее число ветвей в программе

Z3 – число связей с прикладными программами

Z4 – число связей с системными программами

Z5 – число операций I/O

Z6 – число вычислительных операторов

Z7 – число операторов обработки данных

Z8 – число комментариев

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 |
| 0.4 | 0.02 | 0.4 | 0.07 | 0.05 | 0.03 | 0.02 | -0.002 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Z1 | Z2 | Z3 | Z4 | Z5 | Z6 | Z7 | Z8 |
| 8 | 5 | 4 | 7 | 28 | 25 | 30 | 39 |

N=8.062

Т.о. при данной сложности ПО возможно проявление 9 ошибок

#### ЗАКЛЮЧЕНИЕ

В результате была разработана структурная и принципиальная схема цифрового диктофона для записи голосовой информации в течение 1 часа. Спроектированное устройство обладает следующими достоинствами: высокая надежность хранения записанной информации, высокая надежность самого устройства вследствие отсутствия движущихся механических деталей; звуковое сопровождение «на­жатий» клавиш; удобство работы с диктофоном.

##### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гук М. Аппаратные средства IBM PC. - С.-Петербург: Издательство «Питер Ком» 999. -816с.

2. Аванесян Г. Р., Левшин В. П. Интегральные микросхемы ТТЛ, ТТЛШ: Справочник. - М.: Машиностроение, 1993

3. Воробьёв Е. П., Сенин К. В. Интегральные микросхемы производства СССР и их зарубежные аналоги - М.: 1990

4. www.atmel.ru