

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт машиностроения
(наименования института полностью)

Кафедра «Промышленная электроника»
(наименование)

11.03.04 Электроника и наноэлектроника
(код и наименование направления подготовки, специальности)

Промышленная электроника
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Электронное устройство для тестовой проверки знаний

Студент	<u>П.А. Каргин</u> (И.О. Фамилия)	<u>_____</u> (личная подпись)
Руководитель	<u>к.т.н., доцент, М.В. Позднов</u> (ученая степень, звание, И.О. Фамилия)	
Консультант	<u>к.т.н., доцент, А.В. Кириллова</u> (ученая степень, звание, И.О. Фамилия)	

Тольятти 2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт машиностроения
(наименование института полностью)

Кафедра Промышленная электроника
(наименование)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

Студент Каргин Павел Алексеевич, Элб-1601а

1. Тема Электронное устройство для тестовой проверки знаний
2. Срок сдачи студентом законченной выпускной квалификационной работы _____

3. Исходные данные к выпускной квалификационной работе

Первичный источник питания: аккумулятор встроенный в корпус

Тип предоставляемых измерений: угол наклона джойстика

Источник питания, используемый для зарядки: внешний блок питания с выходом 5В DC

Наличие дисплея и графического интерфейса: да

4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов)

Аннотация

Введение

1. Состояние вопроса

2. Формулирование цели и задач проекта

3. Анализ исходных данных и известных решений

4. Формулировка требований к разрабатываемому устройству

5. Подбор электронной компонентной базы для текущего решения

6. Разработка и расчет электрической схемы устройства

7. Разработка ПО для устройства

8. Выбор остальных элементов устройства

9. Разработка корпуса для устройства

10. Изготовление прототипа устройства

Заключение

Список литературы

5. Ориентировочный перечень графического и иллюстративного материала

1. Блок-схема программы

2. Схема устройства структурная

3. Схема устройства электрическая принципиальная

4. Временные диаграммы работы

5. Изометрия корпуса устройства

6. Интерфейс и файловая структура устройства

6. Консультанты по разделам Позднов М.В.

7. Дата выдачи задания «26» января 2020 г.

Руководитель выпускной
квалификационной работы

(подпись)

М.В. Позднов

(И.О. Фамилия)

Задание принял к исполнению

(подпись)

П.А. Каргин

(И.О. Фамилия)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт машиностроения
(наименование института полностью)

Кафедра Промышленная электроника
(наименование)

КАЛЕНДАРНЫЙ ПЛАН
выполнения бакалаврской работы

Студент П.А. Каргин
по теме Электронное устройство для тестовой проверки знаний

Наименование раздела (главы) ¹	Плановый срок выполнения раздела (главы) ¹	Фактический срок выполнения раздела (главы) ¹	Отметка о выполнении раздела (главы) ¹
Состояние вопроса	10.02.2020	10.02.2020	Выполнено
Формулирование цели и задач проекта	15.02.2020	15.02.2020	Выполнено
Анализ исходных данных и известных решений	28.02.2020	28.02.2020	Выполнено
Формулировка требований к разрабатываемому устройству	10.03.2020	10.03.2020	Выполнено
Подбор электронной компонентной базы для текущего решения	15.03.2020	15.03.2020	Выполнено
Разработка и расчет электрической схемы устройства	01.04.2020	01.04.2020	Выполнено
Разработка ПО для устройства	20.04.2020	20.04.2020	Выполнено
Выбор остальных элементов устройства	25.04.2020	25.04.2020	Выполнено
Разработка корпуса для устройства	30.04.2020	30.04.2020	Выполнено
Изготовление прототипа устройства	15.05.2020	15.05.2020	Выполнено
Заключение	30.05.2020	30.05.2020	Выполнено
Список литературы	05.06.2020	05.06.2020	Выполнено

Руководитель бакалаврской работы _____

(подпись)

М.В. Позднов
(И.О. Фамилия)

Аннотация

Объектом исследования является системы проведения индивидуальных тестов.

Данная работа содержит введение, семь глав, таблиц, изображений, заключения, списка литературы, в том числе с зарубежными источниками, и графической части на шести листах формата А1.

Для моделирования схем была использована программа Microsoft Visio. Для моделирования корпуса была использована программа КОМПАС 3D. Код был синтезирован в программе Arduino IDE.

Основной задачей выпускной работы является проектирование специального устройства для проведения индивидуального тестирования. Проектируемое устройство должно иметь ряд обязательных функций и технических особенностей для расширения возможностей проводимых на данном устройстве тестов, а также иметь максимально эргономичные и относительно недорогие компоненты.

В выпускной квалификационной работе затрагивается проблема необходимости приобретения программных или технических средств, способных осуществить проведение тестирования среди обучающихся по различным образовательным программам.

В данном случае идет речь о минимальной стоимости, и как следствие, доступности, при максимальном необходимом функционале устройства для проведения качественного тестирования, что делает его востребованным среди образовательных учреждений регулярно сталкивающихся с нехваткой денежных средств для приобретения подходящего оборудования.

В работе рассмотрены различные решения, способные обеспечить требуемый функционал устройства с учетом минимальных денежных затрат. Выигрыш по стоимости не может быть конкретно установлен, ввиду рознящейся стоимости аналогов. Также описан процесс проектирования технической части и описан процесс синтеза программного кода.

В итоге мы представляем готовое устройство с необходимым требованиями, использование которого в образовательных целях поможет избавиться от необходимости приобретать дорогостоящее оборудование и разнообразит обучающий процесс.

Abstract

The title of the graduation work is Individual Testing Device.

The research focuses on the systems for conducting individual tests.

This work consists of introduction, seven chapters, tables, images, conclusion, references, including international sources, and a graphic part on 6 A1 sheets.

Microsoft Visio program was used for modeling of circuits. COMPAS 3D program was used for modeling the body of the device. The code was synthesized with the Arduino IDE program.

The main objective of the graduation work is to design a specialized device for individual testing. The device must have a number of mandatory functions and technical features in order to expand the capabilities of tests carried out on this device. It also must contain ergonomic and moderately priced components.

We touch upon the problem of the relevance of software and hardware purchase that can be used to carry out testing among students of different educational programs.

A special part of the work provides details about the minimum costs which results in affordability along with the maximum required functionality of the device. In this case, we are talking about the minimum cost and as a result, affordability along with the maximum functionality required for high-quality testing. It will make the device very attractive for educational establishments which often face a lack of funding for the purchase of suitable equipment.

The graduation work considers various solutions that can provide the required functionality of the device along with the minimum costs. The value gain cannot be specifically determined due to the varying prices of

analogues. The process of designing the technical part is also described as well as the process of synthesizing the program code.

As a result, we present the finished device which meets the established requirements. Its use for educational purposes will help to eliminate the need to purchase expensive equipment and diversify the learning process.

Содержание

Введение.....	3
1 Актуальность устройства	4
2 Формулирование цели и задач проекта	5
3 Состояние вопроса	6
4 Аппаратная часть.....	31
5 Программная часть.....	51
6 Выбор остальных элементов устройства	62
7 Корпус устройства	64
Заключение	65
Список используемой литературы	66
Приложение А Блок-схема программы	69
Приложение Б Структурная схема устройства	74
Приложение В Электрическая принципиальная схема устройства.....	75
Приложение Г Временные диаграммы	77
Приложение Д Корпус устройства	78
Приложение Е Программный код	79

Введение

С развитием прогресса, вычислительная электроника охватила более широкую аудиторию потребителей и плотно осела в бытовом применении. В том числе и в образовательной сфере в различных ее проявлениях.

В наши дни процессы обучения стремительно переходят на цифровой уровень (компьютерные тесты, развивающие электронные интерактивные игрушки, дистанционное обучение). Соответственно, данная тенденция формирует иной иное мышление у нового поколения людей, а старое вынуждено своевременно подстраиваться под нее.

Отдельно стоит обратить внимание на потребность обучающихся тех или иных образовательных заведений в потребности различных технических средств для образовательного процесса. Актуальность приобретения высока в связи с высокой рыночной стоимостью необходимого оборудования.

На рынке представлен весьма узкий ассортимент решений, большинство из которых основаны на программной составляющей для персональных компьютеров в оффлайн или онлайн виде.

1 Актуальность устройства

Возможности предоставления обучаемому образовательной среды часто ограничены финансово, и как следствие, технически. Обеспечение компьютерного класса с современным оборудованием для школы, или создание сайта с системой обучения и тестирования требует весьма значительных затрат, варьирующихся в широком ценовом спектре. Поэтому разработка доступных в плане стоимости устройств или сред обучения может быть весьма актуальной на сегодняшний день.

1.1 Вывод

В данном разделе было определено, что на данный момент проектируемое устройство имеет определенную актуальность для широкой аудитории обучающихся.

2 Формулирование цели и задач проекта

Основная задача – разработать компактное устройство, отвечающее за проведение индивидуального теста с четырьмя вариантами ответа.

Для достижения поставленной цели необходимо выполнить следующие этапы:

- Рассмотреть возможные аналоги проектируемого устройства и обозначить их плюсы и минусы;
- Определить функции устройства. Составить структурную схему устройства;
- Осуществить подбор компонентов в соответствие с требуемыми функциями и ранее обозначенными плюсами и минусами аналогов;
- Протестировать работоспособность отдельных сложных модулей;
- Составить электрическую принципиальную схему устройства;
- Составить программный код устройства;
- Разработать корпус устройства.

2.1 Вывод

В данном разделе была определена основная цель текущей работы. Также были обозначены этапы, выполнение которых необходимо для достижения поставленной цели.

3 Состояние вопроса

Согласно требованиям к изделию, была создана первичная структурная схема (рисунок 1).

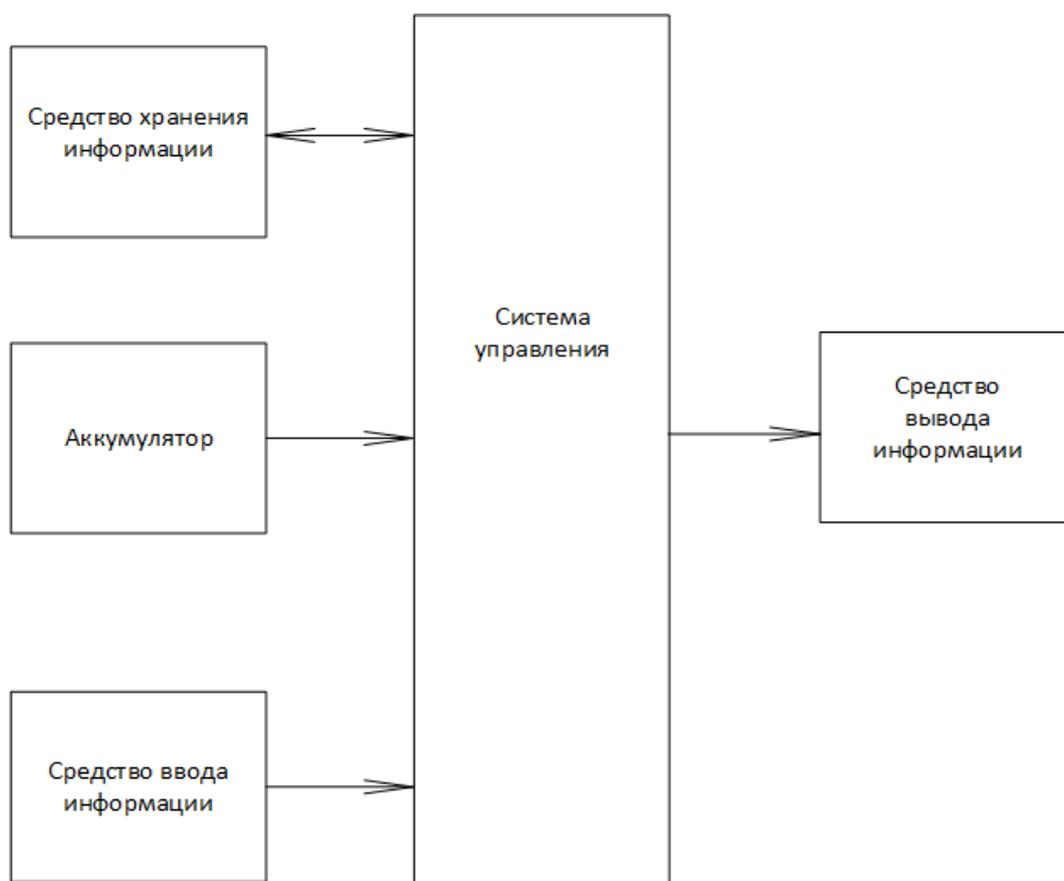


Рисунок 1 - Первичная структурная схема

Из схемы можно понять, что система управления будет считывать вводимые данные, обращаться к средству хранения информации, и выводить эту информацию с помощью одного или нескольких средств.

Перед подбором компонентной базы для реализации данных функций, следует провести анализ известных решений, с целью обнаружения возможных недостатков.

3.1 Анализ исходных данных и известных решений

Был произведен анализ аналогов разрабатываемого устройства на рынке. В ходе поиска, были обнаружены следующие решения.

3.1.1 Интерактивные детские коврики с воспроизведением аудио

Данный аналог похож на устройство тем, что в нем присутствует интерактивная составляющая и наличие функции аудиовоспроизведения вопросов и ответов. Пример аналога показан на рисунке 2.1.



Рисунок 2.1 - Интерактивный детский коврик

Аналог представляет собой многослойный пластиковый “коврик”, с контактными площадками под каждой из ключевых изображений. В данном случае это цифры и изображения транспорта. Сверху располагается пластиковый корпус с электронной логической схемой с динамиком, в которую записаны вопросы в аудио формате. При нажатии на ту или иную контактную площадку, происходит ответ на поставленный вопрос.

Характеристики аналога варьируются. Постоянным остается питание от двух батарей типа АА. Также, как правило, эти коврики занимают большую площадь.

Минусы аналога:

- Количество вариантов ответа ограничено изображениями на поверхности;
- Отсутствие возможности перезаписи информации;
- Большие габариты;
- Узкая целевая аудитория (в основном дети дошкольного возраста).

Плюсы аналога:

- Относительно низкая стоимость;
- Автономное питание за счет аккумуляторов;
- Наличие озвученных ответов и вопросов, что дополнительно повышает интерес пользователя и упрощает конструкцию.

3.1.2 Детский ноутбук

Данный аналог нацелен на аудиторию школьников начальных классов, а также детей дошкольного возраста. Пример аналога показан на рисунке 2.2.



Рисунок 2.2 - Детский ноутбук

Аналог представляет собой корпус, имитирующий ноутбук, с нехарактерно маленьким монохромным экраном, множеством кнопок, а также имитацией компьютерной мыши.

В программной базе данного устройства присутствуют множество интерактивных игр, предполагающих использование экрана, динамиков и клавиатуры устройства.

Питание осуществляется за счет батарей типа АА в количестве от двух штук, в зависимости от модели устройства.

Минусы аналога:

- Монохромность экрана;

- Отсутствие возможности перезаписи программ и вопросов;
- Узкая целевая аудитория, как и у предыдущего аналога;
- Невозможность проведения теста в привычном понимании, что для целей данной работы также является минусом.

Плюсы аналога:

- Невысокая стоимость;
- Наличие дисплея;
- Возможность ввода текста и выбора вариантов ответа;
- Озвучка вопросов и ответов, с выводом соответствующих изображений;
- Автономное питание за счет аккумуляторов.

3.1.2 Персональный компьютер

Данное устройство присутствует почти в каждой семье и способно обеспечить выполнение множества задач, в том числе и проведение тестов. Состоит из системного блока, монитора и различных устройств ввода и вывода; таких как клавиатура, мышь и наушники. В память компьютера устанавливается программное обеспечение, обеспечивающее эргономичную работу пользователя с компьютером. Функционал программы, обеспечивающей процесс тестирования, может быть расширен ее автором в соответствии с нуждами заказчика.

В зависимости от комплектации, цена компьютера варьируется. Цена с необходимыми вычислительными возможностями для проведения теста и базовой комплектацией ориентировочно составляет 25 тысяч рублей. Пример аналога показан на рисунке 2.3.



Рисунок 2.3 - Персональный компьютер

Плюсы аналога:

- Высокая производительность;
- Возможность редактирования и добавления новых программ, а в данном случае, тестов;
- Наличие цветного экрана, что позволяет демонстрировать не только текст, но и изображения;
- Возможность воспроизводить аудиофайлы, что требуется при тестах с аудированием;
- Наличие клавиатуры, что позволяет отвечать на вопросы более развернуто.

Минусы аналога:

- Отсутствие мобильности (компьютер работает от сети);
- Громоздкость конструкции;
- Высокая стоимость.

Исходя из вышеперечисленного, проектируемое устройство должно иметь помимо прочих следующие преимущества:

- Мобильность;
- Наличие цветного экрана;
- Возможность редактирования и перезаписи информации;

- Низкая стоимость;
- Возможность воспроизведения аудио.

3.2 Подбор электронной компонентой базы для текущего решения

Исходя из установленных недостатков рассмотренных аналогов, была составлена более подробная структурная блок-схема, показанная в Приложении Б. Исходя из ее составляющих, можно рассматривать способы осуществления определенных средств.

3.2.1 Система управления

Система управления была выбрана с учетом простоты освоения и доступности на рынке. Выбор был сделан в сторону микроконтроллеров Arduino, которые к тому же имеют встроенные гнезда для коммутации.

Существуют несколько популярных часто используемых моделей микроконтроллеров Arduino.

3.2.1.1 Arduino Nano

Микроконтроллер отличается малыми размерами и небольшим объемом памяти, однако для рядовых проектов, как правило, его вполне достаточно. Цена варьируется в зависимости от оригинальности платы. Стоимость реплики составляет 150 рублей. Пример платы можно увидеть на рисунке 3.1.

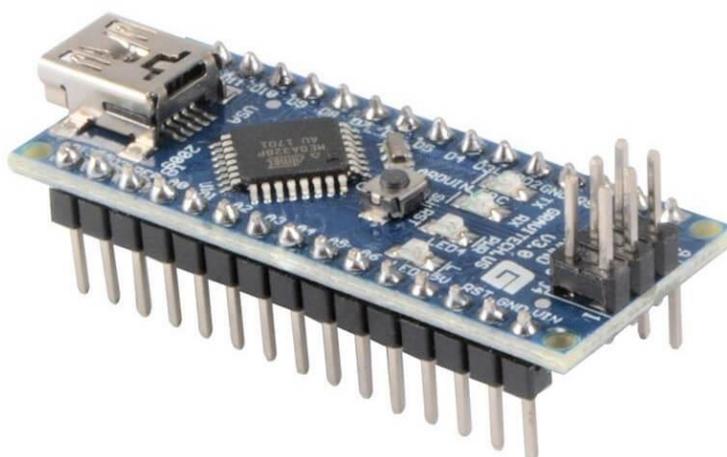


Рисунок 3.1 - Микроконтроллер Arduino Nano

Данный микроконтроллер имеет следующие характеристики [9]:

Таблица 1.1 - Характеристики микроконтроллера Arduino Nano

Наименование	Характеристика
Микроконтроллер	Atmel ATmega168 или ATmega328
Рабочее напряжение (логическая уровень)	5 В
Входное напряжение (рекомендуемое)	7-12 В
Входное напряжение (предельное)	6-20 В
Цифровые Входы/Выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	8
Постоянный ток через вход/выход	40 мА
Флеш-память	16 Кб (ATmega168) или 32 Кб (ATmega328) при этом 2 Кб используются для загрузчика
ОЗУ	1 Кб (ATmega168) или 2 Кб (ATmega328)
EEPROM	512 байт (ATmega168) или 1 Кб (ATmega328)
Тактовая частота	16 МГц
Размеры	1.85 см x 4.2 см

Преимущества:

- Небольшой размер;
- Питание и передача данных через порт microUSB.

Недостатки:

- Небольшая встроенная память;
- Небольшое количество входов/выходов.

Учитывая требования к количествам входов/выходов и объему памяти, данный вариант не подходит для реализации проектируемого устройства.

3.2.1.2 Arduino Uno

Данный микроконтроллер чаще всего используется в проектах на базе микроконтроллеров Arduino, так как имеет весь функционал, требуемый рядовым пользователем. Цена варьируется в зависимости от оригинальности платы. Стоимость реплики составляет 280 рублей. Пример платы можно увидеть на рисунке 3.2.



Рисунок 3.2 - Микроконтроллер Arduino Uno

Краткие технические характеристики следующие [10]:

Таблица 1.2 - Характеристики микроконтроллера Arduino Uno

Наименование	Характеристика
Микроконтроллер	ATmega328
Рабочее напряжение	5 В
Входное напряжение (рекомендуемое)	7-12 В

Продолжение таблицы 1.2

Входное напряжение (предельное)	6-20 В
Цифровые Входы/Выходы	14 (6 из которых могут использоваться как выходы ШИМ)
Аналоговые входы	6
Постоянный ток через вход/выход	40 мА
Постоянный ток для вывода 3.3 В	50 мА
Флеш-память	32 Кб (АТmega328) из которых 0.5 Кб используются для загрузчика
ОЗУ	2 Кб (АТmega328)
EEPROM	1 Кб (АТmega328)
Тактовая частота	16 МГц

Преимущества:

- Увеличенный объем памяти;
- Возможность отдельного использования источника питания.

Недостатки:

- Средний объем памяти;
- Все еще недостаточное количество входов/выходов;
- Учитывая требования к количествам входов/выходов и объему памяти, данный вариант также не подходит для реализации проектируемого устройства.

3.2.1.3 Arduino Mega

Данный микроконтроллер приобретается с целью использования в проекте большой компонентной базы, так как данная модель имеет преимущество перед большинством микроконтроллеров на базе Arduino. Цена варьируется в зависимости от оригинальности платы. Стоимость реплики составляет 650 рублей. Пример платы можно увидеть на рисунке 3.3.



Рисунок 3.3 - Микроконтроллер Arduino Mega

Краткие технические характеристики следующие [8]:

Таблица 1.3 - Характеристики микроконтроллера Arduino Mega

Наименование	Характеристика
Микроконтроллер	ATmega2560
Рабочее напряжение	5В
Входное напряжение (рекомендуемое)	7-12В
Входное напряжение (предельное)	6-20В
Цифровые Входы/Выходы	54 (14 из которых могут работат также как выходы ШИМ)
Аналоговые входы	16
Постоянный ток через вход/выход	40 mA
Постоянный ток для вывода 3.3 В	50 mA
Флеш-память	256 КВ (из которых 8 КВ используются для загрузчика)
ОЗУ	8 КВ
Энергонезависимая память	4 КВ
Тактовая частота	16 МHz

Из вышеперечисленных характеристик следует, что данная модель подходит для реализации проектируемого устройства, так как имеет большой объем памяти и большое число входов/выходов.

3.2.2 Вывод текста

Вывод текста с возможными вариантами ответа может быть осуществлен с помощью нескольких вариантов электронных модулей:

3.2.2.1 OLED дисплей

Органический светодиод (англ. *organic light-emitting diode*, сокр. OLED) — полупроводниковый прибор, изготовленный из органических соединений, эффективно излучающих свет при прохождении через них электрического тока. Пример экрана изображен на рисунке 3.4.



Рисунок 3.4 - OLED дисплей

Основное применение OLED-технология находит при создании устройств отображения информации (дисплеев).

Достоинства:

- Самый дешевый вариант для вывода текста;
- Высокая яркость;
- Низкое энергопотребление.

Недостатки:

- Маленький размер у большинства монохромных моделей и как следствие, большая ограниченность в отображении данных;
- Низкий срок службы;
- Хрупкость моделей, используемых в связке с микроконтроллером Arduino.

3.2.2.2 Текстовый LCD дисплей

Дисплей на жидких кристаллах (Liquid Crystal Display) используется для отображения графической или текстовой информации в компьютерных мониторах (также и в ноутбуках), телевизорах, телефонах, цифровых фотоаппаратах, электронных книгах, навигаторах, планшетах, электронных переводчиках, калькуляторах, часах и т. п., а также во многих других электронных устройствах.

В данном случае рассматривается вариант дисплея с отдельными ячейками для отображения символов и разделением на строки. Такие дисплеи используются в кассовых аппаратах, станках ЧПУ и везде, где требуется отображение небольшого количества текстовой информации, удобной для прочтения. Пример дисплея изображен на рисунке 3.5.



Рисунок 3.5 - LCD дисплей

Достоинства:

- Отображение символов пригодным для чтения размером шрифта;
- Достаточный объем отображаемых данных;

- Небольшое число контактов (при использовании платы I2c).
- Недостатки:
- Относительно большой размер;
- Большое количество подключаемых контактов, при использовании без специальной шины.

3.2.2.3 TFT дисплей

Жидкокристаллический дисплей с активной матрицей (TFT LCD, англ. thin-film transistor — тонкоплёночный транзистор) — разновидность жидкокристаллического дисплея, в котором используется активная матрица, управляемая тонкоплёночными транзисторами.

В данном случае рассматриваются модели, применяемые в связке с микроконтроллером Arduino. Пример дисплея показан ниже на рисунке 3.6.

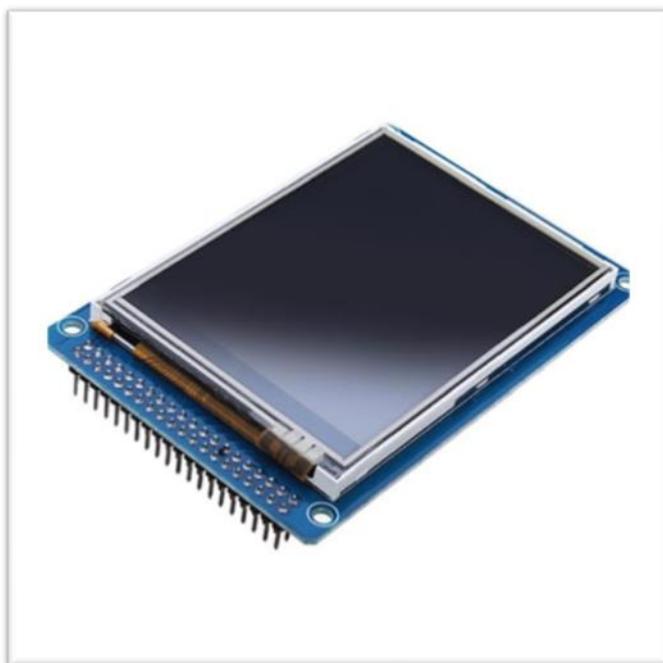


Рисунок 3.6 - TFT дисплей

Достоинства:

- Большое пространство для отображения данных;

- Возможность выбора цвета;
- Возможность выбрать подходящий размер.

Недостатки:

- Большое количество выводов;
- Высокая стоимость;
- Непригодность для вывода большого объема текста крупным шрифтом.

Рассмотрев возможные решения, выбор был сделан в пользу LCD экрана, так как данный вид дисплеев идеально подходит для отображения текстовой информации.

Таблица 2.1 – Варианты моделей текстовых LCD экранов

Наименование	Описание	Цена, руб.
LCM2004A-D02	Текстовый LCD экран 4 строки по 20 символов, интерфейс I2C, поддержка кириллицы.	~250
HD44780	Текстовый LCD экран 2 строки по 16 символов, интерфейс I2C.	~200
SUR2004A_I2C	Текстовый LCD экран 4 строки по 20 символов, интерфейс I2C, вывод кириллических символов возможен, посредством сторонней программной библиотеки, но ограничен.	~250

Была выбрана модель LCM2004A-D02. Ее технические характеристики указаны ниже [14]:

- Потребляемое напряжение - 5В;
- Количество выводов – 16;
- Размер с платой – 97x60 мм;

- Потребляемый ток – 60мАч.

Отличительной особенностью выбранной модели является наличие кириллической прошивки, что отличает данную модель от остальных представителей с ограничениями отображения кириллицы. Также, экран имеет четыре строки по двадцать символов в каждой, что вполне приемлемо для вывода вариантов ответа, которых будет как раз четыре, как в классическом тесте.

3.2.3 Отображение графических изображений

Это также должно быть предусмотрено устройством. В следствии чего, был выбран экран TFT из предыдущего сравнения, ввиду возможности отображения цветов, относительно низкой стоимости и распространенности на рынке.

Таблица 2.2 – Варианты моделей графических TFT экранов

Наименование	Описание	Цена, руб.
EYEWINK Nextion	Графический TFT экран 2.4 дюйма, разрешение 320x240, microSD разъем присутствует.	~1000
ST7735	Графический TFT экран 1.8 дюйма, разрешение 128x160, microSD разъем отсутствует.	~180
ST7735SD	Графический TFT экран 1.8 дюйма, разрешение 128x160, microSD разъем присутствует.	~250

Была выбрана модель на базе ST7735 [18] без разъема microSD, ввиду небольшой цены и размеров. Также отличительной особенностью данного модуля является отсутствие разъема под microSD, что в нашем случае выгодно, так как носитель microSD в данном проекте будет работать не только с экраном TFT.

3.2.4 Воспроизведение аудиофайлов

Осуществляется с целью экономии места для вывода текста а также проведения аудирования, может быть осуществлено с помощью:

3.2.4.1 Сторонний модуль воспроизведения mp3

Данный модуль может быть полноценным плеером, так как его выводы отвечают за выполнение всех основных функций. В электронной составляющей присутствует усилитель звука и специальный разъем для карты памяти microSD. Вдобавок, для работы потребуется динамик. Модуль изображен на рисунке 3.7.

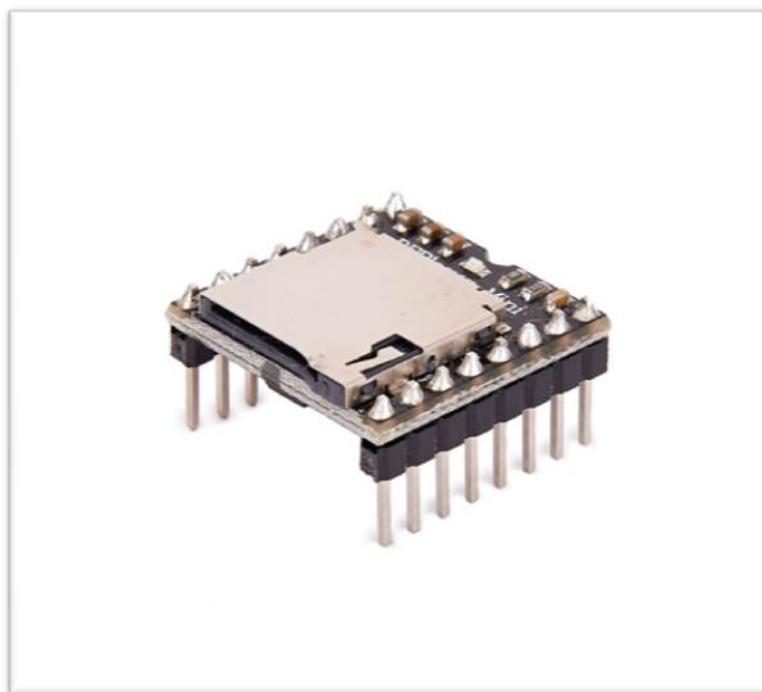


Рисунок 3.7 - Модуль воспроизведения mp3

Достоинства:

- Возможность воспроизведения файлов в самом распространенном на сегодняшний день формате mp3;
- Высокая громкость воспроизведения за счет встроенного усилителя;
- Небольшие габариты.

Недостатки:

- Высокая стоимость;
- Требуется отдельный microSD носитель, что создает неудобство для обновления файлов в данном проекте, также от этого повышается цена;
- Необходимость стирать и перезаписывать библиотеку аудиофайлов при каждом ее обновлении. Особенность заключается в устройстве модуля, который ведет отсчет файлов не по имени, а по очередности записи на носитель.

3.2.4.2 Встроенная функции микроконтроллера проигрывания wav файлов

Arduino имеет возможность воспроизводить аудиофайлы в формате wav. Схема для реализации этой функции была взята из открытого источника. Для этого потребуется усилитель, динамик, резистор и 2 конденсатора. Схема с характеристиками электронных компонентов указана ниже на рисунке 3.8.

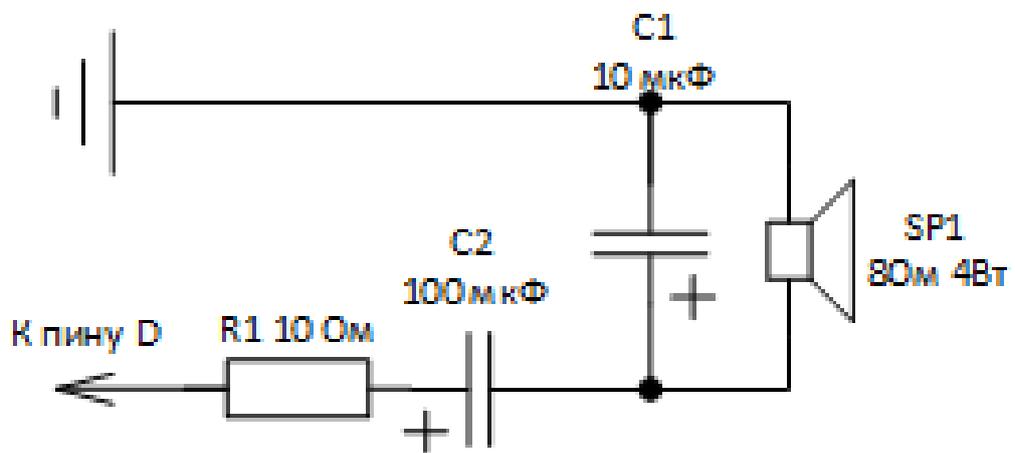


Рисунок 3.8 - Схема для реализации воспроизведения wav аудиофайлов

Достоинства:

- Низкая стоимость компонентов для реализации, даже с учетом стоимости усилителя;
- Отсутствие необходимости отдельного носителя microSD.

Недостатки:

- Маленькая громкость без использования усилителя;
- Необходимость конвертации аудиофайлов в формат wav;
- Необходимость уменьшать дискретность файла для возможной обработки микроконтроллером.

Поскольку громкость на выходе будет небольшой, требуется подключение усилителя. Учитывая, что пользователю потребуется изменять громкость в процессе прохождения теста, не прибегая к использованию дополнительных кнопок, подключенных к микроконтроллеру, была выбрана модель усилителя GF1002 на базе чипа РАМ8403. Данная модель имеет встроенный потенциометр, позволяющий регулировать громкость. Стоимость усилителя составляет 140 рублей вместе с динамиком. Внешний вид показан на рисунке 3.9.



Рисунок 3.9 - Усилитель GF1002 на базе чипа РАМ8403

Усилитель имеет следующие характеристики [16]:

- Частота преобразования усилителя (Класс D): 350 КГц (это позволяет не использовать индуктивности в схеме - их заменяет индуктивность динамиков, сведя к минимуму габариты и вес стереоусилителя);
- КПД: более 90% (не требует радиаторов охлаждения);
- Рабочее напряжение(V): 2.5-5;
- Мощность на канал: 3Вт.

Выбор был сделан в пользу встроенной функции проигрывания wav аудиофайлов, не только из за низкой стоимости, но и из за того, что в данном случае возможно хранить все необходимые файлы для корректной работы на одном microSD носителе. Также был подобран соответствующий усилитель с регулятором громкости, ввиду необходимости ее изменять в процессе прохождения теста и отсутствия в конструкции устройства предполагаемых регуляторов, которые будут свободны при выборе вариантов ответа.

3.2.5 Средство ввода

Оно требуется для выбора предоставленных вариантов ответа. Самым подходящим вариантом оказался аналоговый джойстик. Джойстик имеет преимущество перед обычными кнопками или кнопочными панелями, так как он занимает малую площадь при относительно большом диапазоне выдаваемых значений. Имеет относительно небольшое число выводов, пять основных положений, что в данном случае подходит для теста с четырьмя вариантами ответа. Стоимость варьируется в пределах 80-120 рублей. Пример выбранной модели изображен на рисунке 3.10.



Рисунок 3.10 - Джойстик аналоговый

Технические характеристики следующие [13]:

- Напряжение питания: номинальное 3.0...5,5 В;
- Выходной сигнал: цифровой (кнопка) и аналоговый (оси X и Y);
- Размеры: 26 мм x 40 мм x 22 мм;
- Диапазон выдаваемых значений потенциометров: 0...1023.

3.2.6 Хранение информации

Данная функция будет осуществляться на отдельном носителе, так как памяти микроконтроллера недостаточно для хранения предполагаемых объемов данных. Был выбран модуль для карт microSD, поскольку он имеет малые габариты относительно модуля для SD карт памяти, и способен на более быструю работу в отличие от носителей USB. Стоимость модуля около 50 руб. microSD карта памяти была оценена в 200 рублей, однако, она зависит от объема памяти и производителя. Пример используемого модуля показан на рисунке 3.11.



Рисунок 3.11 - Модуль для карт памяти microSD

Характеристики выбранной модели следующие [15]:

- Логика напряжения - 3.3В и 5В;
- Потребляемое напряжение - 3,3В;
- Поддерживаемые форматы – FAT32 (до 16Гб) и FAT16 (до 2Гб).

3.2.7 Вывод числа правильных и неправильных ответов

Вывод счета ответов на TFT экран возможен, но малоэффективен ввиду того, что при большом шрифте на экране не будет видно изображения, а при маленьком шрифте цифры будут плохо заметны. Поэтому данная функция была реализована с помощью блока семисегментных индикаторов модели tm1637 с четырьмя ячейками и разделительным двоеточием, что как раз подходит для вывода ответов. В таком случае, количество вопросов будет ограничено числом в 99 штук, так как на число суммы верных и на число суммы неверных ответов приходится по две цифровые ячейки. Стоимость данной модели около 50 руб. Пример такого модуля изображен на рисунке 3.12.



Рисунок 3.12 - Блок семисегментных индикаторов tm1637

3.2.8 Аккумулятор

Он должен заряжаться посредством распространенных блоков питания на 5В, предназначенных как правило для мобильных устройств. Исходя из этого, была выбрана модель контроллера заряда HW-357, способная осуществить коммутацию блока питания, аккумулятора и самого устройства. Также данная плата имеет возможность регулировать выходное напряжение. Внешний вид показан на рисунке 3.13.



Рисунок 3.13 - Внешний вид контроллера заряда HW-357

Данная модель имеет следующие характеристики [6]:

- Входное напряжение 4,5-8В;
- Напряжение заряда до 4,2В +/- 1%;
- Максимальный зарядный ток 1000мА;
- Расчетный ток разряда 2А;
- Пределы регулируемого выходного напряжения 4,3-27В.

Как можно заметить на рисунке 3.13, данная модель имеет разъем вида microUSB, что является эргономичным решением в плане коммутации, так как кабели с таким разъемом довольно распространены на сегодняшний день.

Контакты IN+ и IN- подключаются к входному питанию, контакты OUT+ и OUT- подключаются к нагрузке, а В+ и В- соединяются с соответствующими контактами аккумулятора. Как правило с данной моделью используются аккумуляторы формата 18650. Пример показан на рисунке 3.14.



Рисунок 3.14 - Аккумулятор модели 18650

Такие аккумуляторы имеют максимальный ток 4,2В, минимальный 3,7В, и обеспечивают выходной ток в размере 1А. Данная связка аккумулятора и контроллера вполне подходит для проектируемого устройства, так как потребление тока у нагрузки незначительное, и рабочее напряжение большинства модулей составляет 5В, которые обеспечивает контроллер заряда на выходе. Стоимость данного аккумулятора составляет примерно 450 руб. Стоимость контроллера HW-357 составляет 30 руб.

3.3 Вывод

В данном разделе были рассмотрены ближайшие аналоги проектируемого устройства и установлены их преимущества и недостатки, которые проектируемое устройство должно иметь или исключать соответственно. Также, в текущем разделе были выбраны необходимые компоненты и предварительно сравнены со своими аналогами с целью выявления модели с минимальной стоимостью и наличием необходимых характеристик.

4 Аппаратная часть

4.1 Определение входов/выходов микроконтроллера, используемых в проектировании устройства

Для дальнейшего подключения электронных компонентов устройства к микроконтроллеру, необходимо обратиться к документации микроконтроллера Arduino Mega и обозначить входы/выходы, которые будут задействованы далее. Последующие задачи будут ссылаться на схему расположения входов/выходов на рисунке 4.1 для определения входов/выходов, к которым будет подключаться тот или иной модуль [21].

ARDUINO MEGA 2560 PINOUT

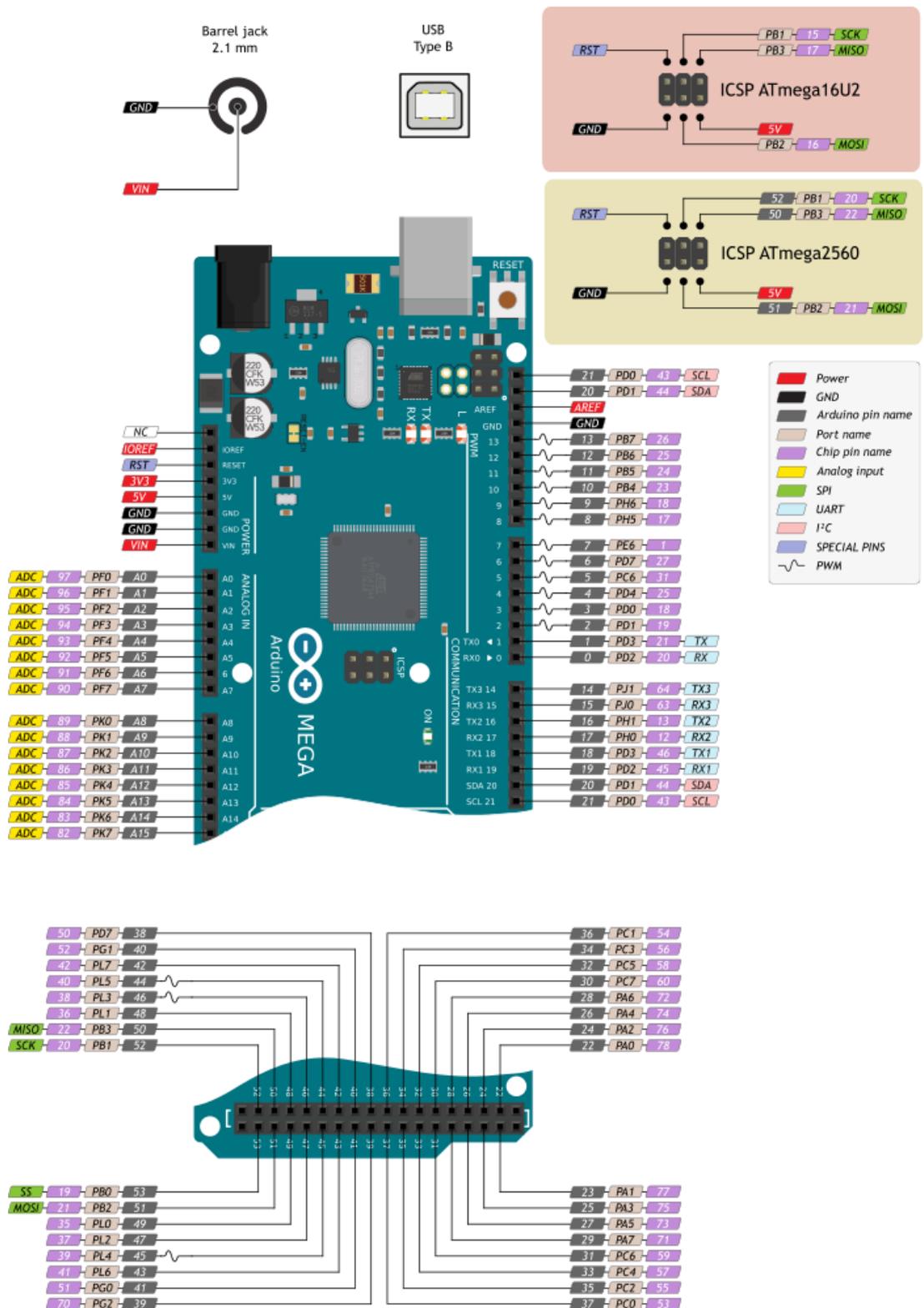


Рисунок 4.1 – Схема расположения входов/выходов микроконтроллера Arduino Mega

Как видно на рисунке 4.1, плата Arduino Mega имеет ряд различных входов/выходов, отвечающих за тот или иной интерфейс. Подключение модулей с использованием какого-либо из имеющихся интерфейсов будет осуществляться с данными входами/выходами микроконтроллера.

4.2 Работа с LCD дисплеем

Для тестирования данного и последующих модулей, на Персональный Компьютер необходимо установить специальную среду.

Микроконтроллеры Arduino как правило программируются посредством специальной кроссплатформенной среды программирования Arduino IDE.

Данная программа бесплатна и позволяет загружать программные библиотеки, упрощающие работу с видом тех или иных модулей.

Стоит упомянуть, что разработка программы не рассматривается в рамках данной задачи и в последующих задачах тестирования модулей, вплоть до специально поставленной. Поскольку для модулей требуется установка соответствующих программных библиотек, в файлах которых как правило имеются тестовые программы, позволяющие проверить работоспособность программируемого модуля.

После установки программной среды, можно переходить к тестированию модуля текстового дисплея.

Из-за большого количества подключаемых контактов выбранного дисплея может не хватить выводов микроконтроллера для присоединения нужных элементов в дальнейшем. Поэтому, следует использовать специальную шину, работающую по протоколу I2C.

I2C / ИС(Inter-Integrated Circuit) – это протокол, изначально создававшийся для связи интегральных микросхем внутри электронного устройства. Разработка принадлежит фирме Philips. В основе I2C протокола является использование 8-битной шины, которая нужна для связи блоков в управляющей электронике, и системе адресации, благодаря которой можно общаться по одним и тем же проводам с несколькими устройствами. Данные передаются то одному, то другому устройству, добавляя к пакетам данных идентификатор нужного элемента. [2]

Самая простая схема I2C может содержать одно ведущее устройство (чаще всего это микроконтроллер Arduino) и несколько ведомых (например, дисплей LCD). Каждое устройство имеет адрес в диапазоне от 7 до 127. Двух устройств с одинаковым адресом в одной схеме быть не должно.

Вся передача данных состоит из Стартовой посылки, битов и стоповой посылки. Порядок изменения уровня на шинах задает тип посылки. Временная диаграмма находится в Приложении Г.

Представление схемы подключения шины I2C к LCD дисплею не требуется, так как очередность контактов шины и дисплея совпадают. Пример приведен на рисунке 4.2.

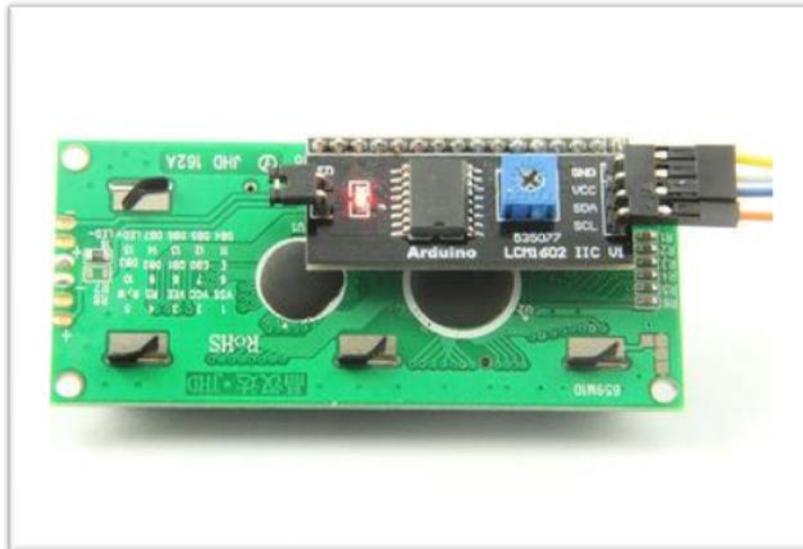


Рисунок 4.2 – Подключенная к дисплею шина I2C

Для использования выбранного дисплея в паре с микроконтроллером Arduino Mega необходимо установить специальную библиотеку в среду программирования Arduino IDE и отладить ее работу.

Была установлена библиотека LCD_1602_RUS [12], взятая из открытого источника. Библиотека составлена специально для дисплеев с вшитой кириллической таблицей.

Далее представлены функции библиотеки, используемые в коде тестовой программы.

- LCD_1602_RUS lcd(0x27, 20, 4); - Функция, описывающая входы, к которым подключен экран, и содержащая информацию о количестве строк и символов в них;
- lcd.begin(); - Функция, отвечающая за обращение программы к адресным входам экрана;
- lcd.backlight(); - Функция, включающая подсветку экрана;
- lcd.setCursor(0,0); - Функция, устанавливающая курсор в определенное положение определенной строки;

- `Icd.print("Message");` - Функция, отвечающая за вывод указанного в ней текста;
- `Icd.asciitf8();` - Функция, перекодирующая текст из формата UTF8 в формат ASCII. Используется в паре с предыдущей.

Данный дисплей соединяется с микроконтроллером посредством интерфейса SPI.

SPI (Serial Peripheral Interface), или последовательный периферийный интерфейс, был разработан компанией Motorola для организации быстрого и простого в реализации обмена данными между компонентами системы — микроконтроллерами и периферийными устройствами. На шине может быть одно ведущее устройство (master) и несколько ведомых (slave).

После инициализации выбирается ведомое устройство (при подключении нескольких, как в проектируемом в данной ВКР устройстве), переводя соответствующий вывод CS в состояние LOW. Затем ведомому устройству передаются данные. После передачи CS возвращается микроконтроллером в состояние HIGH [3]. Временная диаграмма указана в Приложении Г.

Кроме очевидно понятных по назначению двух контактов (GND – земля, VCC – питание +5В), присутствуют также два контакта SDA - двунаправленный канал передачи данных, SCL — тактовый сигнал от ведущего устройства. Данные контакты подключаются к определенным входам/выходам микроконтроллера, отвечающих за работу с SPI соединением. Как видно на схеме документации, представленной ранее, это цифровые входы/выходы под номерами 21 и 20 соответственно. Схема подключения показана на рисунке 4.3.

SPLC780D1

Upper 4 bit Lower 4 bit	LLLL	LLLH	LLHL	LLHH	LHLL	LHLH	LHHL	LHHH	HLLL	HLLH	HJLH	HJHH	HOLL	HOLH	HOOH	HOOH
LLLL				00P'P												
LLLH		!	1A0a9													
LLHL		"	2BRbr													
LLHH		#	3CSca													
LHLL		\$	4DTdt													
LHLH		%	5EUeu													
LHHL		&	6FUfu													
LHHH		'	7GWgw													
HLLL		(8HXhx													
HLLH)	9IYiy													
HHLH		*	: JZjz													
HLHH		+	; K[k]k													
HLLL		.	< L[l]l													
HLLH		-	= M[m]m													
HHLH		.	> N[n]n													
HHHH		/	? O[o]o													

Check Sum : 00003F61

Wed Dec 02 13:30:41 2009

Orise Technology Co., Ltd

Рисунок 4.4 - Таблица символов ASCII дисплея

Данная проблема была исправлена. Фрагмент кода библиотеки, отвечающего за сопоставление битов к символам представлен ниже:

```
void LCD_1602_RUS::printwc(const wchar_t _chr) {
    uint8_t rus_[8];
    switch (_chr)
    {
//для буквы а
        case 1072: //бит соответствующий букве а
            LiquidCrystal_I2C::print("\x61");//адрес в таблице символов
            break;
    }
}
```

Таким образом были прописаны функции к каждой букве русского алфавита, кроме заглавной буквы «Ё», так как ее байт совпадает с байтом символа Q.

Далее был запущен стандартный код-пример из файлов библиотеки для проверки корректной работы дисплея. На рисунке 4.5 представлен выведенный на экран кириллический алфавит.

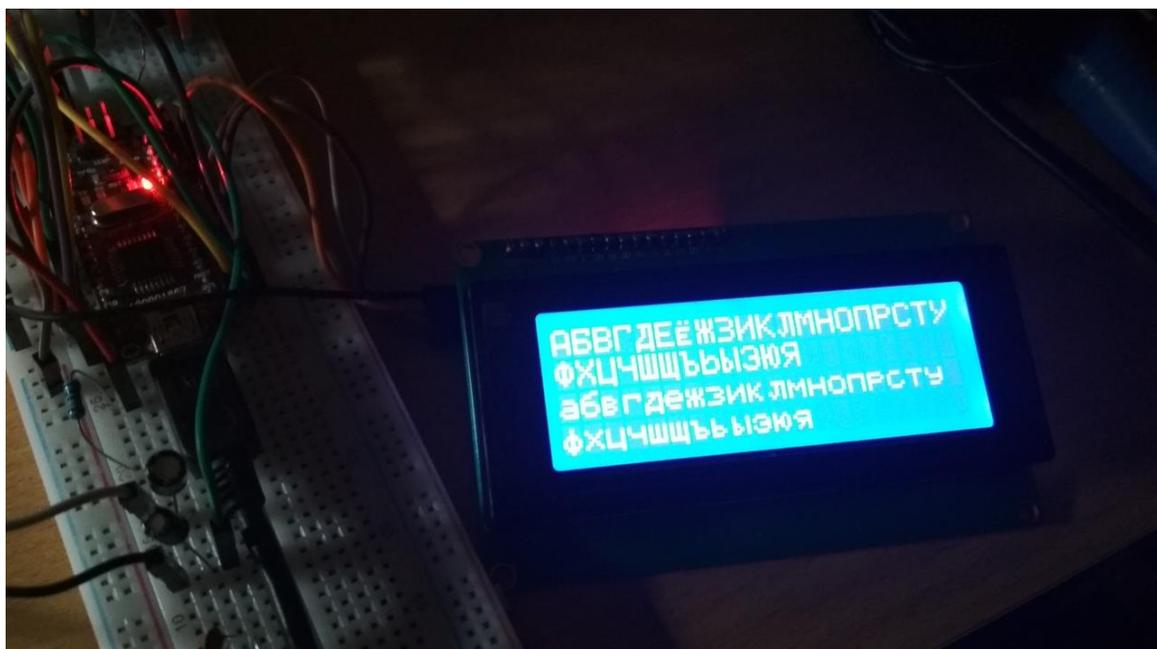


Рисунок 4.5 - Вывод текста на экран LCD

Как видно на рисунке 4.5, кириллический алфавит отображается корректно.

4.3 Работа с TFT дисплеем

Для модели дисплеев выбранной серии имеются две популярные библиотеки из открытых источников - Adafruit и Ucglib. Первая имеет более расширенный функционал, хотя и уступает второй в скорости работы с экраном, поэтому была использована библиотека Adafruit [19].

Данный дисплей, как и ранее рассматриваемый дисплей LCD, имеет интерфейс соединения типа SPI. Контакты дисплея имеют следующие наименования:

- RST – Сброс и отключения питания;
- CS – Выбор ведомого; устанавливается ведущим;

- DC – Передача команд;
- SDA – Двухнаправленный канал передачи данных;
- SCL - Тактовый сигнал (от ведущего);
- VDD – Питание +5В;
- GND – Земля;
- BLK – Подсветка, максимум +5В.

Обратившись к схеме дисплея [4], можно определять входы/выходы для использования:

- RST – 7,
- DC – 8,
- CS – 9,
- SDA – 51,
- SCL – 52.

Контакты SDA и SCL были подключены в режиме использования встроенного в микроконтроллер аппаратного SPI, так как данный режим позволяет выводить изображения значительно быстрее.

Для подключения дисплея к микроконтроллеру Arduino необходимо соединять большинство выводов посредством токоограничивающих резисторов. Это связано с разнящимся напряжением логики устройств.

После подключения в память микроконтроллера был загружен код из числа программ-примеров для тестирования работоспособности дисплея. Ниже на рисунке 4.6 видны результаты выполненной программы.

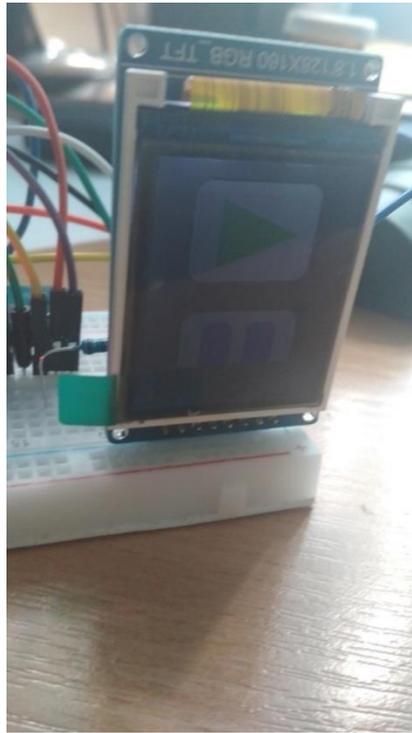


Рисунок 4.6 - Вывод тестовых изображений на дисплей

На данном этапе работа с TFT дисплеем не завершается, но этап можно считать успешно преодоленным.

4.4 Работа с модулем microSD

Для дальнейшего использования средств вывода по требуемому назначению, необходимо настроить и проверить работоспособность модуля для карт microSD.

Данный модуль способен считывать и записывать информацию на носитель microSD в связке с микроконтроллером. Модуль работает с логикой 3.3В и 5В, поэтому наличие токоограничивающих резисторов при коммутации не обязательно.

Сам носитель формата microSD должен отвечать определенным требованиям, а именно, иметь объем памяти до 16Гб и в следствии этого, быть отформатированным до вида FAT32.

Обратившись к схеме расположения входов/выходов, были определены номера цифровых входов/выходов для подключения данного модуля: CS – 53, MOSI – 51, SCK – 52, MISO – 50. Подключение модуля к микроконтроллеру показано на рисунке 4.7.

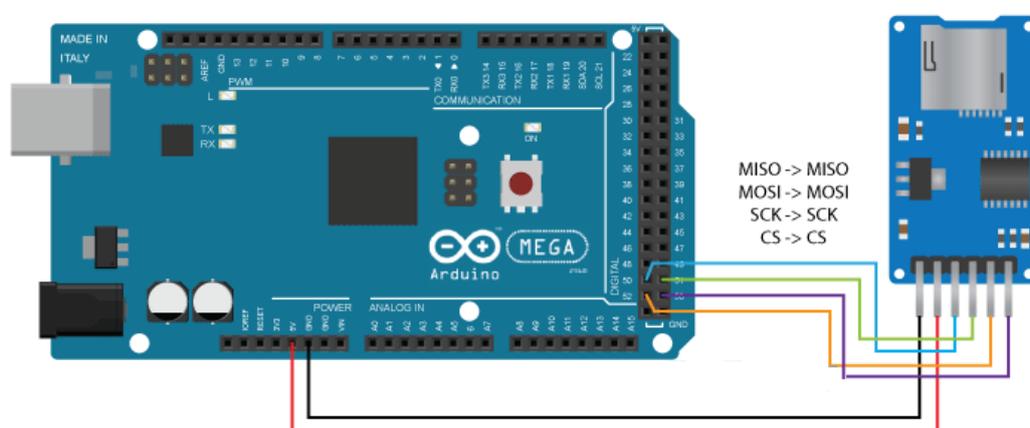


Рисунок 4.7 - Подключение модуля microSD к плате Arduino

В случае с моделью микроконтроллера Arduino Mega, номера выводов отличаются от используемых на других моделях. На схеме выше можно увидеть, какие номера выводов задействованы.

Далее следует открыть в среду Arduino IDE соответствующую стандартную библиотеку и запустить код-пример из файлов библиотеки, отвечающий за запись и считывание.

Ниже приведены команды, используемые в коде программы в дальнейшем:

- `SD.begin(chipSelect);` - Начать работу модуля с указанием на подключенный вывод CS. `chipSelect` – в данном случае вывод

микроконтроллера, к которому подключен вывод CS от модуля microSD;

- SD.open("test.txt"); - Открыть файл с именем test.txt;
- available() – Проверка файла на доступность к обработке;
- read(); - Считывание бита из файла с последующим перемещением указателя на следующий символ;
- close(); - Закрытие файла после работы с ним.

Работоспособность была проверена в связке с LCD экраном, так как он уже был протестирован на работоспособность. Однако, запись и считывание текстовой информации это только часть требований проектируемого устройства.

4.5 Вывод графический изображений на TFT дисплей

В открытых источниках авторов библиотеки Adafruit был найден код [5], отвечающий за вывод изображений, хранящихся на microSD носителе, на TFT дисплей в формате bmp. Поскольку в конструкции TFT дисплея не установлен разъем для карт памяти microSD, что и было задумано в пользу эргономичности, в решении данной задачи дисплей тестировался в связке с модулем microSD.

Также стоит упомянуть, что при выводе изображений на дисплей, вместо считывания, а потом окраски пикселей поочередно, считываются несколько пикселей сразу, и значение их цветов переносится в соответствующий буфер.

Изображение должно иметь формат bmp, глубину цвета пикселей 24 и разрешение 128x160 пикселей.

Основные функции, взаимодействующие с дисплеем и используемые в тестовой программе указаны ниже:

- `tft.pushColors(lcdbuffer, lcdidx, first);` - Выставление ряда закрасенных пикселей в соответствующую строку;
- `tft.setAddrWindow(x, y, x1, y1);` - Выбор отображаемой области изображения. Применяется в случае обработки изображения, превышающего разрешение экрана;
- `tft.setRotation (1);` - Установка начала координат. Т.н. поворот экрана.

Далее на рисунке 4.8 видно успешно отображенное изображение, которое было заранее записано на карту памяти.



Рисунок 4.8 – Вывод изображения на экран

Кроме вывода на экран изображений с внешнего носителя microSD, было принято решение синтезировать лицо некоего персонажа с помощью функции отрисовки форм. Оно будет иметь несколько вариантов эмоций, что позволит создать приятное впечатление у проходящего тест, ведь будет выглядеть так, что именно этот персонаж задает вопросы и реагирует на ответы. Это повысит

интерес и создаст благоприятную эмоциональную атмосферу, которая необходима человеку для лучшей концентрации.

Используя стандартные команды, была составлена программа отрисовки шести лиц персонажа, речь о которой пойдет в дальнейшем в пункте 5.1 «Разработка ПО устройства». Код, о котором идет речь в упомянутой задаче, изменяет значение координат фигур, при получении соответствующей переменной. На рисунке 4.9 представлено одно из лиц.

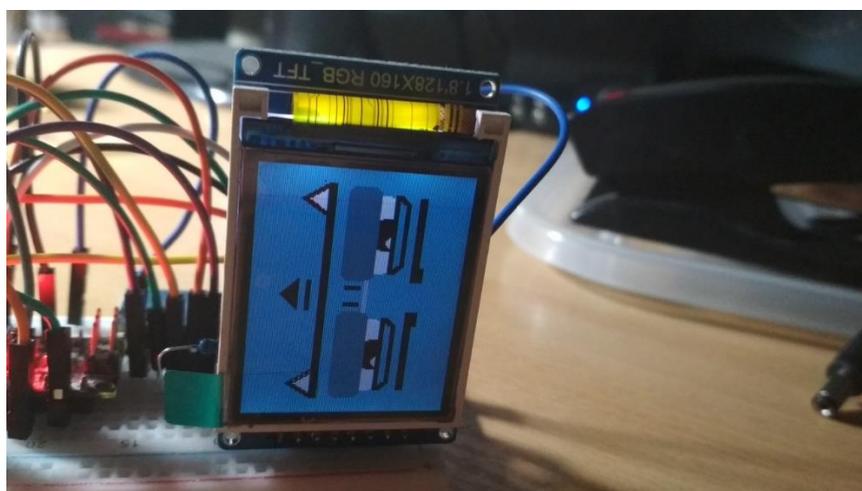


Рисунок 4.9 - Отрисовка лица персонажа

Основные команды библиотеки Adafruit, используемые для отрисовки форм в данном случае следующие:

- `tft.fillScreen(ST77XX_BLACK);` - Заливка экрана определенным цветом. При отсутствии цвета в стандартной палитре библиотеки, его следует внести в системный файл библиотеки в формате RGB и задать имя;
- `tft.fillRoundRect (x,y,x1,y1,r,ST77XX_WHITE);` - Отрисовка скругленного прямоугольника, заполненного цветом. Указываются координаты, цвет и радиус скругления;

- `tft.fillCircle(x,y,r,ST77XX_BLACK);` - Отрисовка заполненного цветом круга. Указываются координаты, радиус и цвет;
- `tft.drawLine(x, y, x1, y1, ST77XX_BLACK);` - Отрисовка линии. Указываются начальные и конечные координаты, а также цвет;
- `tft.fillTriangle(x,y,x1,y1,x2,y2,ST77XX_BLACK);` - Заполненный цветом треугольник. Указываются три координаты вершин и цвет;
- `tft.drawFastHLine(x, y, l, ST77XX_BLACK);` - Сокращенная функция отрисовки горизонтальной линии. Указываются начальные координаты и длина;
- `tft.drawFastVLine(x, y, l, ST77XX_BLACK);` - Сокращенная функция вертикальной линии. Аналогична предыдущей функции.

4.6 Настройка воспроизведения аудио

Для проведения аудирования, и просто в целях экономии места для текста, необходимо осуществить функцию воспроизведения аудиофайлов посредством встроенной функции микроконтроллера.

Для этого на макетной плате была собрана электрическая цепь на основе схемы, приведенной на рисунке 4.10.

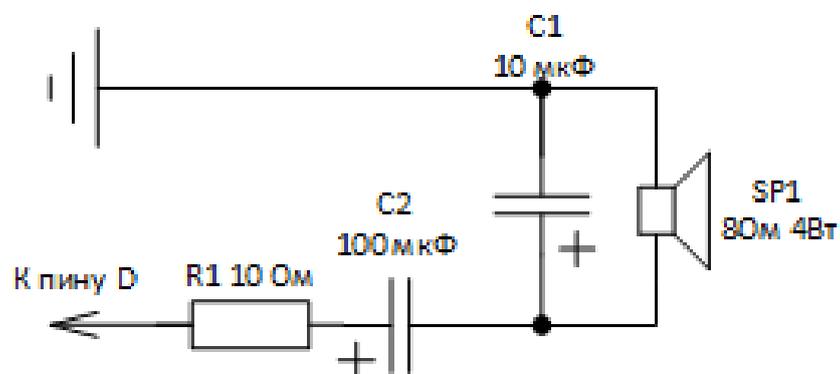


Рисунок 4.10 - Схема цепи, реализующей вывод аудио

«Пин D » на данной схеме – любой свободный цифровой вход/выход микроконтроллера, способного к Широтно Импульсной Модуляции. В данном случае был выбран цифровой выход под номером 46.

Далее из открытого источника была загружена специальная библиотека [20] и установлена в программную среду Arduino IDE [7]. В файлах библиотеки, как и в большинстве общедоступных, была тестовая программа, с помощью которой и была проверена работоспособность данного модуля в связке с модулем для карт памяти microSD.

Для исключения повышенной нагрузки на процессор микроконтроллера, файл wav необходимо привести к виду:

- Частота - 16 КГц,
- Количество бит – 8,
- Канал – Моно.

Для этого был использован онлайн конвертер [11].

Основные команды данной библиотеки, используемые в коде, приведены ниже:

- `TMRpcm tmrpcm;` - Создание объекта с обращением к библиотеке;
- `tmrpcm.speakerPin = 10;` - Определение вывода микроконтроллера;
- `tmrpcm.play("File.wav");` - Проигрывание файла с именем File.wav.

После загрузки кода в память микроконтроллера, аудиофайл начал проигрываться, как и планировалось программой. На этом этапе

задачу можно считать решенной. В прототипе устройства данная схема использоваться не будет, так как вместо нее будет установлен усилитель.

4.7 Разработка и расчет электрической схемы устройства

Перед конструированием устройства была составлена электрическая принципиальная схема, в которой подключены выводы всех модулей к входам/выходам микроконтроллера, согласно особенностям каждого модуля, описанных в предыдущих задачах. Схема изображена в Приложении В.

Схема была составлена в программе Microsoft Visio, так как она удобна в установке и освоении, и позволяет загружать в базу и редактировать графические объекты, в том числе и условно графические изображения, отвечающие ГОСТ 2.710-8 [1]. Также стоит упомянуть, что программа бесплатная.

На схеме можно увидеть, чем является тот или иной элемент и отследить, к каким входам/выходам микроконтроллера подключены используемые модули, речь о которых шла в предыдущих задачах.

Сразу стоит отметить модули, контакты которых могут быть подключены к произвольным аналоговым или цифровым входам/выходам микроконтроллера.

Аналоговые выводы джойстика были подключены к произвольным аналоговым входам микроконтроллера под номерами А4 и А5, а вывод нажимной кнопки к цифровому входу D4, так как микроконтроллер Arduino имеет широкий диапазон аналоговых и цифровых входов/выходов, не отвечающих за интерфейс SPI, речь о котором шла в предыдущих задачах.

По такой же причине контакты модуля семисегментных индикаторов CLK и DIO были подключены к входам/выходам D5 и D6 соответственно [17].

Контакты контроллера заряда подключены в соответствии со схемой, указанной в анализе исходных данных. Конденсатор С отвечает за стабилизацию питания усилителя, гнезда XS1 и XS2 отвечают за подключение питания к контроллеру заряда, XS3-XS5 являются выходом для наушников, подключенным через делители напряжения, выключатель SW1 отвечает за включение и выключение питания, а выключатели SW2 и SW3 отвечают за переключение выхода на наушники и выхода на динамик.

4.8 Вывод

В данном разделе была проведена проверка работоспособности LCD дисплея, TFT дисплея, воспроизведения звука с помощью микроконтроллера и модуля microSD и самого модуля microSD. Были рассмотрены виды соединений I2C и SPI, используемые в проектируемом устройстве. Также были определены рабочие входы/выходы микроконтроллера для использования в электрической принципиальной схеме. Была составлена электрическая принципиальная схема, обеспечивающая связь всех требуемых узлов.

5 Программная часть

5.1 Разработка ПО для устройства

На данном этапе необходимо составить программу для комплексного функционирования модулей и обеспечения корректной работы проектируемого устройства. Программная среда для программирования микроконтроллеров Arduino уже была рассмотрена ранее и установлена.

От составляемой программы требуется выполнение следующих функций:

- Выводить меню выбора вариантов ответа на текстовый LCD экран и осуществлять переход между режимами меню в зависимости от контекста;
- Реагировать на выбор варианта ответа с помощью принимаемых с аналогового джойстика сигналов;
- В случае обработки вопроса, содержащего графическое изображение, выводить таковое на дисплей TFT. Также выводить лица созданного персонажа в зависимости от контекста;
- Воспроизведение аудиофайлов wav с вопросами и фразами персонажа;
- Вести счет верных и неверных ответов и выводить их на модуль семисегментных индикаторов;
- Обращаться к карте памяти microSD с целью считывания файлов.

Для примерного обозначения структуры программы была построена блок-схема, описывающая основные элементы программы и переходы между ними. Структурная схема указана в Приложении А.

Как описывалось в предыдущих задачах, работа с большинством модулей в данном устройстве подразумевает включение программных библиотек в код:

- `#include <SPI.h>` - Библиотека, отвечающая за соединение SPI;
- `#include <LCD_1602_RUS.h>` - Библиотека для работы с LCD дисплеем с I2 шиной и кириллическими символами;
- `#include <SD.h>` - Библиотека для работы с модулем карт памяти;
- `#include <Wire.h>` - Библиотека для модулей с использованием I2C;
- `#include <TM1637.h>` - Библиотека, упрощающая работу с семисегментным индикатором с I2C соединением;
- `#include <TMRpcm.h>` - Библиотека для воспроизведения файлов wav;
- `#include <SPI.h>` - Библиотека для модулей с SPI соединением;
- `#include <Adafruit_GFX.h>` - Библиотека ядра графики Adafruit;
- `#include <Adafruit_ST7735.h>` - Библиотека с ПО для моделей TFT дисплеев ST7735.

Здесь же объявляются контакты блока семисегментных индикаторов.

Далее необходимо ввести переменные, отвечающие за следующие функции:

- Изменение координат отрисовки элементов лица персонажа в зависимости от контекста;
- Счет пройденных вопросов в текущей категории;
- Счет пройденных категорий;
- Счет верных и неверных ответов;
- Маркировка категорий как пройденные;
- Хранение байта с цифрой правильного ответа для сравнения;
- Хранение информации с джойстика;

- Определения номера режима меню;
- Наименование возможных событий при различных сигналах с джойстика;
- Хранение случайно сгенерированного числа;
- Счет символов для разделения текста на строки.

Стоит упомянуть о предстоящей работе с кодом для модуля microSD. В обычных случаях в команду прочтения файла путь к файлу пишется вручную. Данный вариант здесь идет в ущерб эргономичности, так как количество файлов не определено, к тому же храниться они будут отсортированными по видам расширений в отдельных папках. Соответственно, прописывать путь для каждого файла отдельно является далеко не продуктивным и эргономичным решением. Ввиду этого, были составлены функции, формирующие массив с прописанным путем к файлу на карте памяти. Для этих функций необходимо объявить отдельные строки: для текстовых, графических и звуковых файлов. Функции представлены далее.

```
char* string2char(String tring){
```

```
    int len = tring.length()+1;
```

```
    char *array_char = new char[len];
```

```
    tring.toCharArray(array_char, len);
```

```
    return array_char;}

```

```
String createPathtxt(int num, int num2) {
```

```
    return String(num) + "/" + "txt" + "/" + String(num2) + ".txt";}
```

```
String createPathbmp(int num3, int num4) {
```

```
return String(num3) + "/" + "pic" + "/" + String(num4) + ".bmp";}
```

```
String createPathwav(int num5, int num6) {
```

```
return String(num5) + "/" + "wav" + "/" + String(num6) + ".wav";}
```

Функции `createPath()` будут присваиваться строкам соответствующих типов файлов. Ис составленных строк с помощью функции `string2char()` будет формироваться массив, который будет помещен в необходимые функции вместо вручную написанного пути к файлу.

В теле `void setup{ }` прописываются команды инициализации TFT дисплея, контакта CS для карты памяти, блока семисегментных индикаторов и LCD дисплея. Объявляются вывод на динамик, аналоговый вход для генератора случайных чисел (необходим для воспроизведения персонажем случайных фраз), а также аналоговые и цифровой входы для джойстика.

Также объявляются номера входов/выходов микроконтроллера для подключения TFT дисплея, семисегментного индикатора, LCD дисплея, аналогового джойстика, модуля microSD, выхода на динамик. Подробнее указано в Приложении E.

В теле `void loop{ }` прописывается инструкция для предотвращения дребезга контактов. В этой инструкции присутствует функция `getPressedButton()`, которая отвечает за определение, при каких значениях потенциометров джойстика считается, что он повернут в ту или иную сторону. Также функция определяет, когда кнопка джойстика нажата. После этого начинается обработка режимов основного меню.

```
switch(Mode){
```

```
case(0):
```

```
lcd.setCursor(6,0);  
lcd.print("Разбуди");  
lcd.setCursor(7,1);  
lcd.print("Зверя");  
break;  
case(1):  
lcd.setCursor(0,0);  
lcd.print("Пройти тест");  
lcd.setCursor(0,1);  
lcd.print("Проверить звук");  
lcd.setCursor(0,2);  
lcd.print("Авторы");  
lcd.setCursor(0,3);  
lcd.print("Стик влево-правила");  
break;  
case(2):  
//меню отрисовки выбора категории  
break;  
case(3):  
//меню отрисовки и выбора ответа  
break;  
case(4):
```

```
//меню регулировки громкости  
  
break;  
  
}
```

В данном фрагменте показана команда, осуществляющая переключение между режимами меню. Case(0) обозначает первый режим, в котором необходимо совершить определенные действия, завершить исполнение и ожидать переключения режима. В случаях с цифрами 2, 3 и 4 тело case() остается пустым, так как от этих режимов требуется лишь ожидание.

Ранее указанная инструкция подавления дребезга контактов содержит в себе функцию ButtonEventProc(), которая возвращает информацию о выбранном положении джойстика: вверх, вниз, влево, вправо и нажатая кнопка (т.н. EN). В этой функции описано, какие команды следует выполнить в текущем режиме при нажатии определенной кнопки. Например, если получено значение EN и режим (Mode) равняется 0, то на экране выводится приветственная надпись, появляется лицо персонажа и воспроизводится фраза в аудиоформате, после чего режим переключается на значение 1, где начинается отрисовка пунктов меню. Также с помощью этой функции можно определить, какой вариант ответа дал пользователь. Номера вариантов ответа сопоставлены следующим образом: UP (вверх) – 1й вариант, RG (вправо) – 2й вариант, DW (вниз) – 3й вариант и LF (влево) – 4й. То есть нумерация осуществляется по часовой стрелке.

Все действия можно проследить на блок-схеме или в полном коде программы, которые находятся соответственно в Приложениях А и Е.

Отдельно стоит отметить функцию отрисовки графических файлов формата bmp, которая была взята из открытого источника, а именно стандартного тестового файла библиотеки Adafruit.

Перед формированием основного кода также было необходимо прописать функцию отрисовки лица персонажа. Основные команды были описаны ранее, в которой проводилось тестирование работоспособности дисплея TFT. Стоит отметить, что для отрисовки различных эмоций одного и того же лица было принято решение ввести ряд переменных, отвечающих за корректировку координат отрисовки большинства элементов лица. Таким образом появилась возможность изменять эмоции на лице персонажа. Два лица, у одного из которых изменено выражение, показаны на рисунках 5.1 и 5.2.

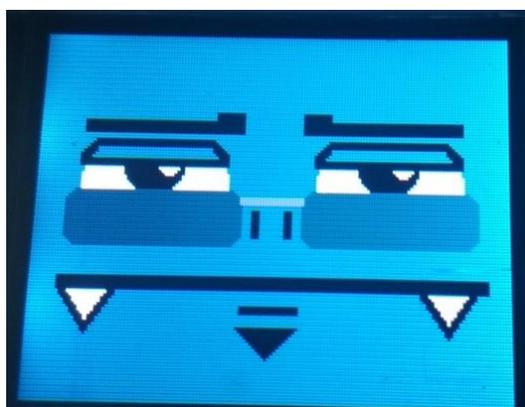


Рисунок 5.1 - Первоначальная эмоция «стандартная»

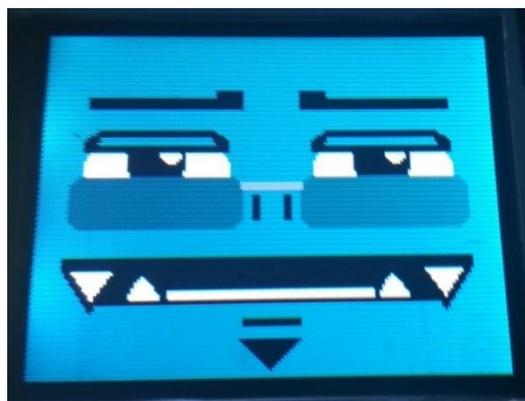


Рисунок 5.2 - Измененная эмоция «радостная»

Для вызова на экран определенного лица, необходимо в функции face(N); на место символа N вписать цифру от 1 до 6, каждая из которых отвечает за изменение корректирующих переменных, и, как следствие, изменение выражения лица.

Определив команды для основного меню, стоит перейти к определению команд функции ButtonEventProc(), в которой будут описаны действия в случае получения определенных значений джойстика в текущем режиме.

5.1.1 Значение EN (ENTER).

Значение получено в режиме 0:

- На LCD дисплей выводятся приветственные надписи;
- Воспроизводятся приветственные фразы;
- На TFT дисплей выводятся соответствующие лица персонажа.

5.1.2 Значение UP.

Значение получено в режиме 1 (Начать тест):

- Включается блок семисегментных индикаторов;
- На LCD дисплей выводятся просьба выбрать категорию вопросов, а затем названия четырех категорий, которые записаны в отдельных текстовых файлах. Если какая-либо категория отсутствует в файловой системе, то вместо ее наименования выводится прочерк;
- Воспроизводится соответствующая фраза;
- Отображается соответствующее лицо;
- Осуществляется переход в режим 2, где ожидается выбор категории.

Значение получено в режиме 2 (Выбрана категория 1):

- Проверяется условие, если категория уже пройдена, то выполняется соответствующая функция, в которой выводится контекстное лицо и пользователю сообщается, что эта категория уже пройдена, и осуществляется переход обратно в режим 2;

- Если категория еще не пройдена, она отмечается переменной как проходимая на данный момент;
- В папке файловой системы проверяется наличие вопроса с текущим номером. Если такового нет, значение счетчика вопросов будет расценено как максимальное, и категория будет считаться пройденной;
- Если вопрос с текущим номером есть, то проверяется наличие изображений для этого вопроса. В случае наличия изображения, оно отображается на TFT дисплее;
- Воспроизводится аудиофайл с номером текущего вопроса;
- На LCD дисплей выводятся варианты ответа;
- Осуществляется переход в режим 3, где ожидается ввод ответа.

Значение получено в режиме 3 (Выбран 1й вариант ответа):

- Бит с цифрой верного варианта ответа сравнивается с битом цифры, записанной в текстовом файле с вариантами и стоящей в самом начале;
- Если ответ совпадает с цифрой в файле, то на блоке семисегментных индикаторов в ячейках для верных ответов значение увеличивается на 1;
- Если ответ не совпадает, значение для неверных ответов увеличивается на 1;
- Выполняется ранее описанный алгоритм по обработке следующего вопроса и переход в режим 3.

5.1.3 Значение DW (DOWN).

Значение получено в режиме 1 (Автор проекта):

- На LCD дисплей выводится информация об авторе проекта;
- Выводится соответствующее лицо;
- Осуществляется переход обратно к режиму 1.

Значение получено в режиме 2 (Выбрана категория 3) - тот же алгоритм, как в случае с UP, но с другими данными.

Значение получено в режиме 3 (Выбран 3й вариант ответа) - тот же алгоритм, как в случае с UP, но с другими данными.

5.1.4 Значение LF (LEFT).

Значение получено в режиме 1:

- На LCD выводится текст с правилами прохождения теста;
- Осуществляется переход в режим 4, где ожидается сигнал с джойстика, для перехода назад к основному меню.

Значение получено в режиме 2 (Выбрана категория 4) - тот же алгоритм, как в случае с UP, но с другими данными.

Значение получено в режиме 3 (Выбран 4й вариант ответа) - тот же алгоритм, как в случае с UP, но с другими данными.

5.1.5 Значение RG (RIGHT).

Значение получено в режиме 1 (Проверка звука):

- На LCD дисплей выводится информация, сообщающая, что происходит проверка звука для регулировки;
- Выводится соответствующее лицо;
- Воспроизводится случайный аудиофайл для проверки;
- Осуществляется переход в режим 1

Значение получено в режиме 2 (Выбрана категория 2) - тот же алгоритм, как в случае с UP, но с другими данными.

Значение получено в режиме 3 (Выбран 2й вариант ответа) - тот же алгоритм, как в случае с UP, но с другими данными.

На этом краткое описание программного кода можно считать завершенным.

В описанных режимах выполняются отдельно описанные функции, такие как: отрисовка лиц, воспроизведение аудио, генерация

случайных чисел, проверка байтов текстового файла, разбиение получаемого текста на строки, формирование строк с путем к тому или иному файлу, проверка пройденных категорий и вопросов, подсчет верных и неверных ответов.

Для подробного рассмотрения каждой команды в Приложении Е есть комментарии.

5.2 Вывод

В данном разделе была составлена блок-схема, по которой написан программный код, обеспечивающий корректную работу модулей и проектируемого устройства в целом. Были рассмотрены основные используемые команды и упрощенная структура кода. Также были описаны процессы, происходящие в тех или иных режимах.

6 Выбор остальных элементов устройства

Перед сборкой прототипа проектируемого устройства, необходимо произвести подбор элементов, необходимых для полноценного функционирования устройства.

6.1 Блок питания

Для зарядки аккумулятора необходим блок питания, обеспечивающий выходное напряжение 4.5-8В. В пользу доступности, был выбран блок питания РКР-00027 на 5В и 1А с разъемом для кабелей USB. Этот блок питания схож с блоками для зарядки мобильных телефонов, и поэтому легко может быть заменен любым другим блоком питания, удовлетворяющим требуемым характеристикам. Стоимость таких блоков варьируется в пределах 150-300 рублей.

6.2 Провод зарядки

Также потребуются провод с штекером USB на одном конце и штекером microUSB на другом для коммутации блока питания и устройства тестирования. Цена варьируется в широких пределах.

6.3 Наушники

Для возможности использования нескольких устройств без шумового засорения учебной аудитории, необходимо приобрести пару наушников. Рекомендуется использовать наушники с сопротивлением

от 16 Ом, поскольку качество воспроизводимых аудиозаписей значительно урезано в пользу производительности. Следовательно, при увеличении громкости появятся большие искажения звука, которые на малом сопротивлении особенно заметны. Цена наушников варьируется в трудноопределимых пределах. От 60 рублей до нескольких тысяч рублей.

6.4 Вывод

В данном разделе были подобраны остальные элементы проектируемого устройства, необходимые для его полноценного функционирования. Также был сделан вывод, что в цену прототипа не стоит включать подобранные в данном разделе компоненты, так как их ценовой диапазон довольно широк.

7 Корпус устройства

7.1 Разработка корпуса для устройства

Корпус устройства был спроектирован в программе КОМПАС 3D. В качестве материала корпуса был выбран прозрачный пластик. Данное решение позволит использовать клейкие пористые прокладки в качестве крепления модулей и отказаться от лишних отверстий. Также, в качестве крепления крышки были использованы отверстия для цилиндров. На чертежах были отмечены необходимые отверстия под разъем microUSB, mini Jack 3.5 и потенциометра усилителя с учетом эргономичности и оптимальным расположением внутренней составляющей. На противоположной стороне располагаются два отверстия для выключателей питания и динамика. Подробнее в Приложении Д.

Было принято решение не проделывать отверстия в крышке корпуса, так как его материал прозрачен. Вместо болтовых креплений модули будут крепиться на губчатые прокладки с клейкой поверхностью, что поможет избежать повреждения плат при внешнем воздействии на устройство.

7.2 Вывод

В данном разделе был подобран корпус, отвечающий требуемым размерам и позволяющий расположить в себе модули в необходимом порядке. Были определены расположения отверстий для необходимых модулей устройства.

Заключение

В конечном итоге, спроектированное устройство и его прототип отвечают заявленным требованиям, имеют требуемый функционал при относительно небольшой стоимости компонентов, а именно 2420 рублей без учета проводов и средств для сборки. Программный код отлажен и позволяет осуществить проведение индивидуального теста со всеми требуемыми функциями.

Несмотря на объем проделанной работы, устройство имеет недостатки и может быть модернизировано. Имеются следующие предложения:

- Осуществить передачу данных от данного устройства к некоему мастер-устройству, способного обновлять данные посредством передачи данных по беспроводной сети. Это позволит отказаться от извлечения и перезаписи карты памяти устройства вручную;
- После каждого прохождения теста отправлять результаты теста с устройства тестирования на какое-либо мастер-устройство. Это поможет быстрее собирать статистику прохождения при одновременном использовании большого количество устройств тестирования;
- Увеличить количество строк и категорий. Как следствие, требуется усовершенствовать способ навигации по меню.

В данной ВКР были проанализированы известные решения, обозначены требования к прототипу, составлена его структурная схема, проведен поиск и подбор компонентной базы, протестирована работоспособность модулей, синтезированы электрическая принципиальная схема, программный код, и корпус.

Список используемой литературы

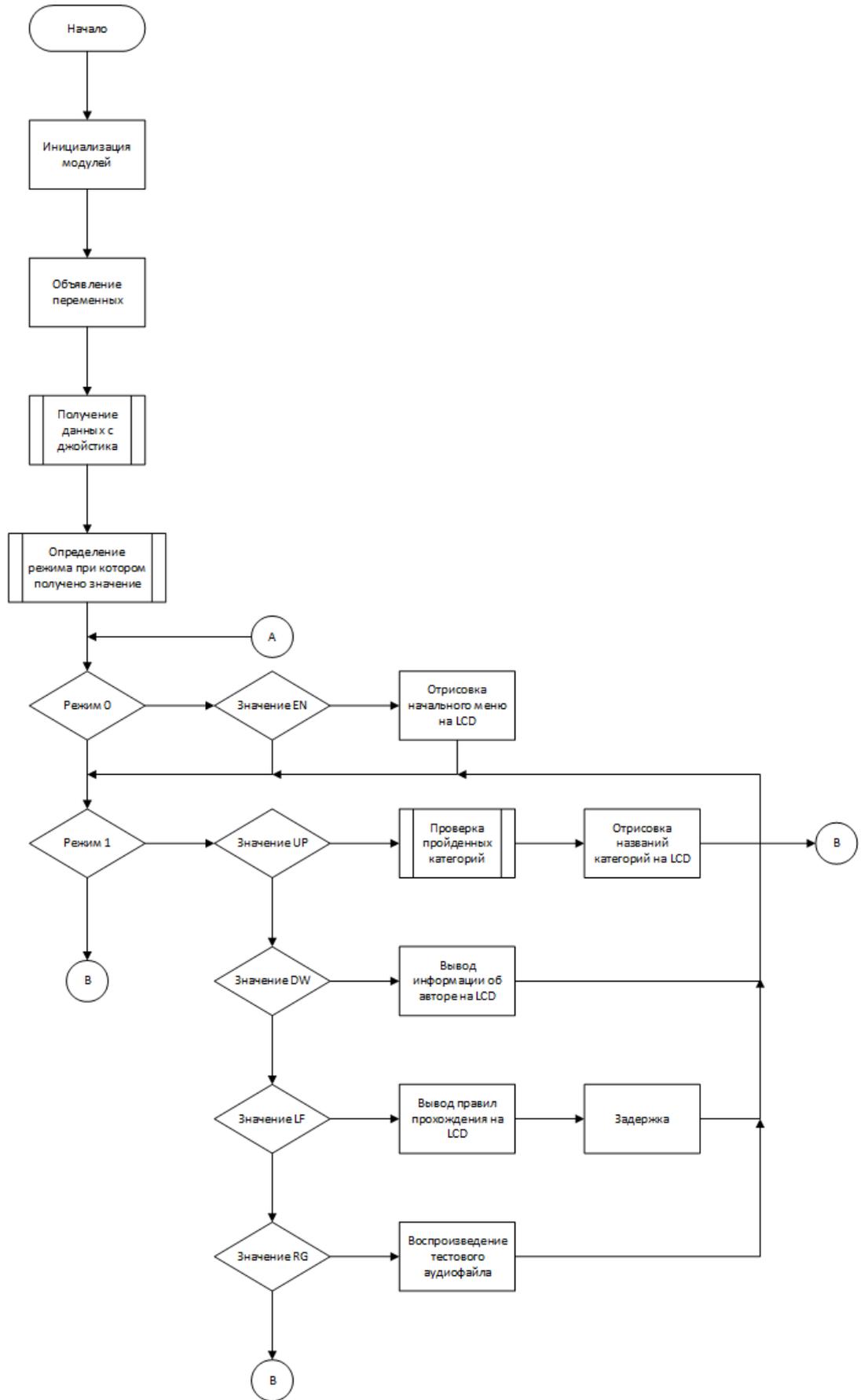
1. Единая система конструкторской документации, ОБОЗНАЧЕНИЯ БУКВЕННО-ЦИФРОВЫЕ В ЭЛЕКТРИЧЕСКИХ СХЕМАХ [Электронный ресурс]: ГОСТ 2.710-81. URL:<http://www.pntd.ru/2.710.htm> (дата обращения: 30.04.2020).
2. Описание интерфейса I2C [Электронный ресурс]: Интерфейсная шина ИС (I2C). URL:<http://easyelectronics.ru/interface-bus-iic-i2c.html> (дата обращения: 15.03.2020).
3. Описание интерфейса SPI [Электронный ресурс]: Интерфейс SPI и Arduino. URL:<https://soltau.ru/index.php/en/arduino/item/382-interfejs-spi-i-arduino> (дата обращения: 15.03.2020).
4. Подключение TFT дисплея [Электронный ресурс]: Гид по TFT LCD дисплеям для Ардуино. URL: <https://arduinoplus.ru/arduino-tft-lcd-display/> (дата обращения: 19.04.2020).
5. Программный код отрисовки изображений [Электронный ресурс]: Arduino:Примеры/soft_spitftbitmap. URL:http://wikihandbk.com/wiki/Arduino:%D0%9F%D1%80%D0%B8%D0%BC%D0%B5%D1%80%D1%8B/soft_spitftbitmap (дата обращения: 19.04.2020).
6. Технические характеристики контроллера заряда [Электронный ресурс]: HW-357. URL:https://www.alibaba.com/product-detail/HW-357-3-7V-liter-5V9V_62295137412.html (дата обращения: 25.02.2020).
7. Arduino IDE [Электронный ресурс]: Download the Arduino IDE. URL:<https://www.arduino.cc/en/main/software> (дата обращения: 20.02.2020).

8. Arduino Mega [Электронный ресурс]: Arduino Mega. URL: <https://store.arduino.cc/usa/mega-2560-r3> (дата обращения: 20.02.2020).
9. Arduino Nano [Электронный ресурс]: Arduino Nano. URL: <https://store.arduino.cc/usa/arduino-nano> (дата обращения: 20.02.2020).
10. Arduino Uno [Электронный ресурс]: Arduino Uno. URL: <https://store.arduino.cc/usa/arduino-uno-rev3> (дата обращения: 20.02.2020).
11. Converter for wav files [Электронный ресурс]: Convert audio to WAV URL: <https://audio.online-convert.com/convert-to-wav> (дата обращения: 10.02.2020).
12. Cyrillic LCD library [Электронный ресурс]: LCD_1602_RUS. URL: https://github.com/ssilver2007/LCD_1602_RUS/blob/master/README.md (дата обращения: 05.01.2020).
13. Documentation for analog joystick [Электронный ресурс]: Analog joystick. URL: https://docs.isy.liu.se/pub/VanHeden/DataSheets/Analog_Joystick.pdf (дата обращения: 10.02.2020).
14. Documentation for LCM2004A [Электронный ресурс]: LCM2004A. URL: <https://www.vistronica.com/images/Documentos/2004a.pdf> (дата обращения: 10.01.2020).
15. Documentation for microSD module [Электронный ресурс]: microSD card module. URL: <http://datalogger.pbworks.com/w/file/fetch/89507207/Datalogger%20-%20SD%20Memory%20Reader%20Datasheet.pdf> (дата обращения: 01.02.2020).
16. Documentation for PAM8403 [Электронный ресурс]: PAM8403. URL: <https://www.diodes.com/assets/Datasheets/PAM8403.pdf> (дата обращения: 01.02.2020).

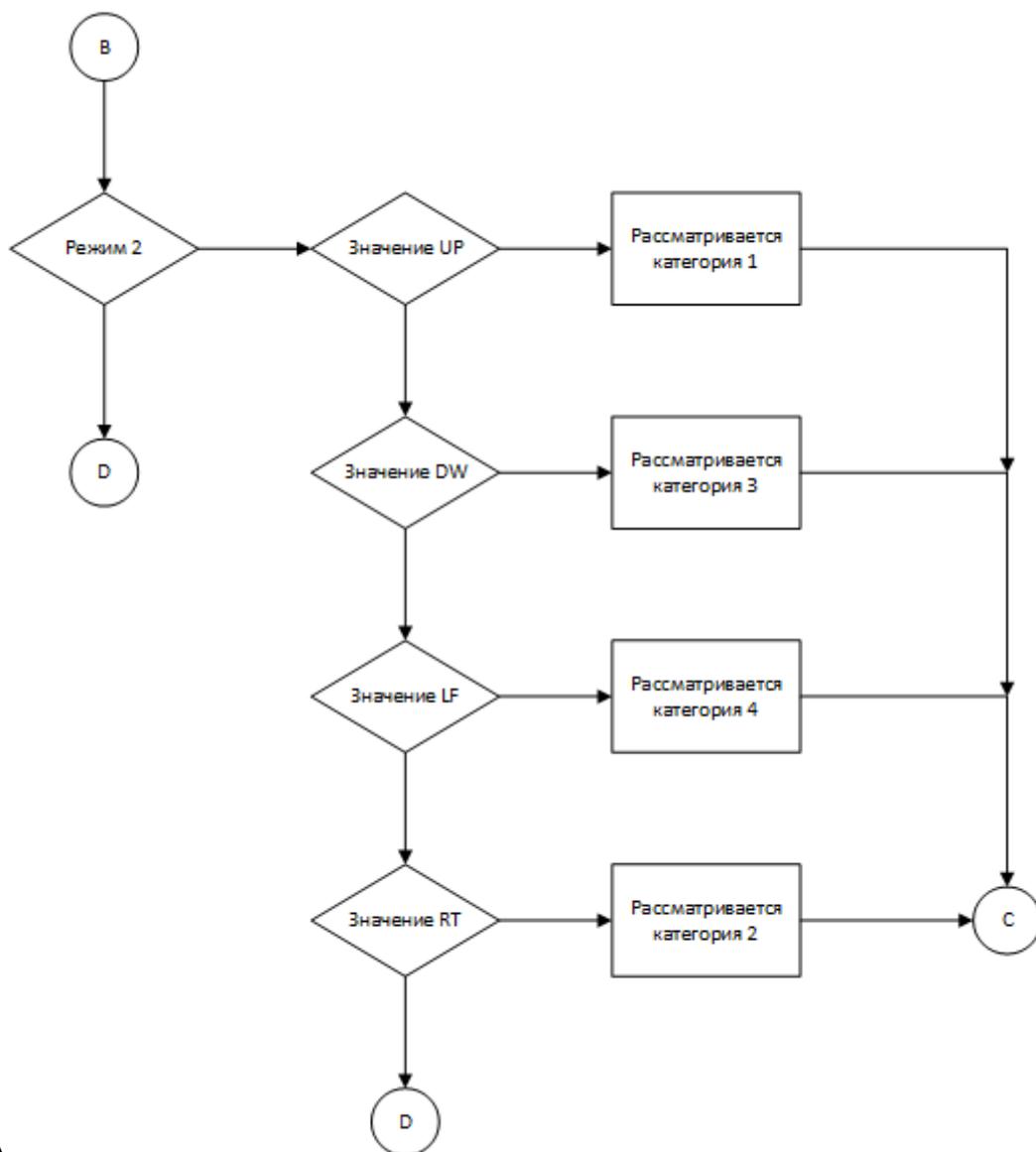
17. Documentation for TM1637 [Электронный ресурс]: Datasheet TM1637. URL:https://www.mcielectronics.cl/website_MCI/static/documents/Datasheet_TM1637.pdf (дата обращения: 15.03.2020).
18. Documentation for TFT 1.8 [Электронный ресурс]: 1.8" 128x160 TFT LCD with SPI Interface. URL:https://www.hotmcu.com/18-128x160-tft-lcd-with-spi-interface-p-314.html?cPath=6_16&zenid=b814flh4b6qo29fov1b2715a13 (дата обращения: 25.02.2020).
19. Library for TFT [Электронный ресурс]: Adafruit-ST7735-Library. URL:<https://github.com/adafruit/Adafruit-ST7735-Library> (дата обращения: 25.02.2020).
20. Library for wav playing [Электронный ресурс]: TMRpcm. URL:<https://github.com/TMRh20/TMRpcm> (дата обращения: 10.02.2020).
21. Mapping for Arduino Mega [Электронный ресурс]: ATmega2560-ArduinoPinMapping. URL:<https://www.arduino.cc/en/Hacking/PinMapping2560> (дата обращения: 05.01.2020).

Приложение А
Блок-схема

программы

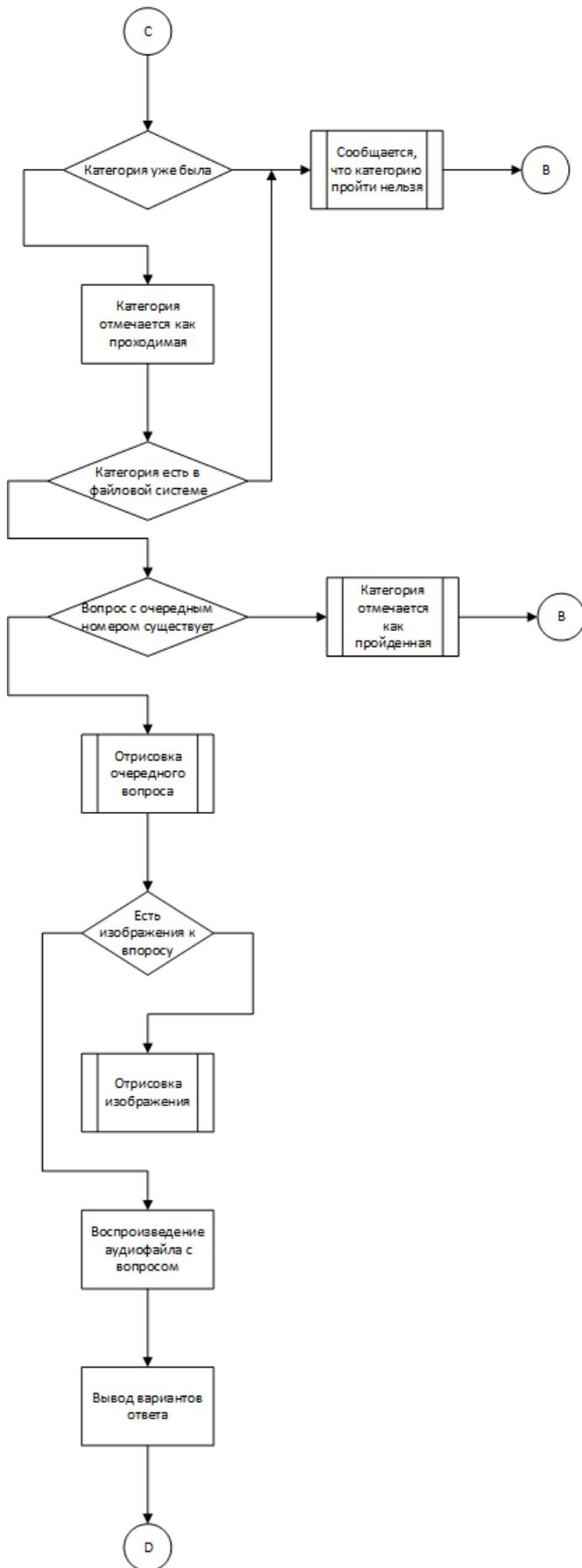


Продолжение Приложения



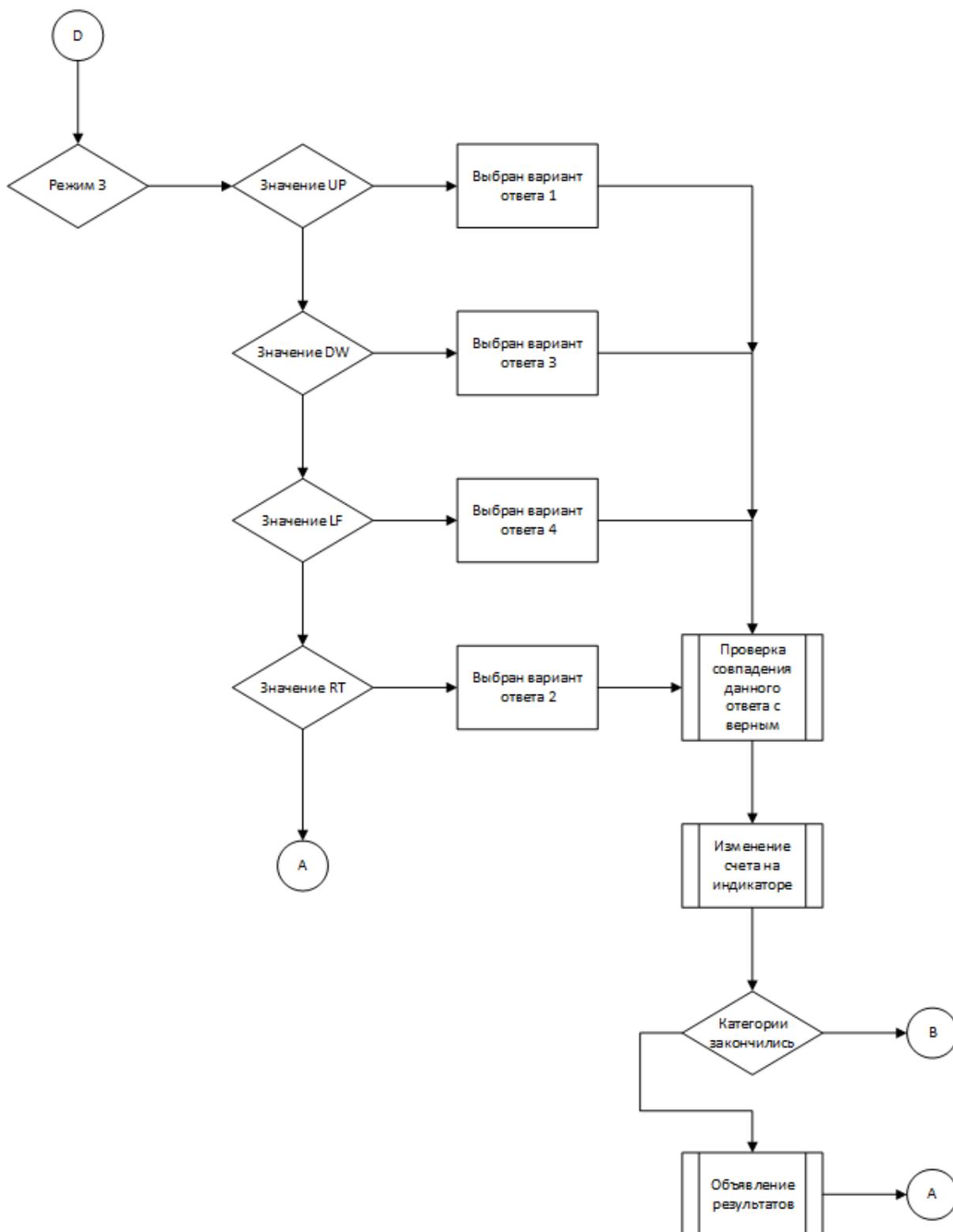
A

Продолжение Приложения А

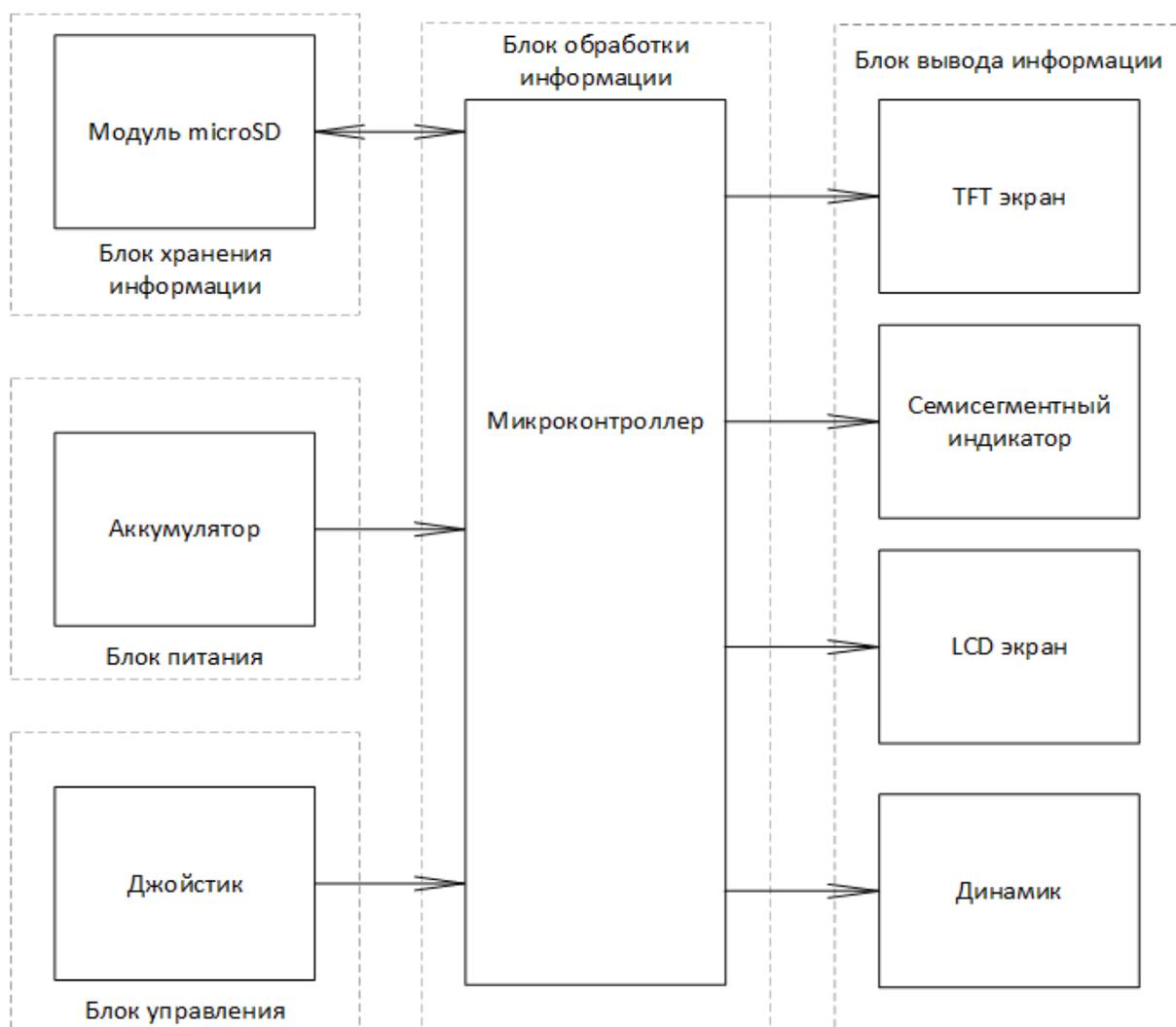


Продолжение Приложения

A

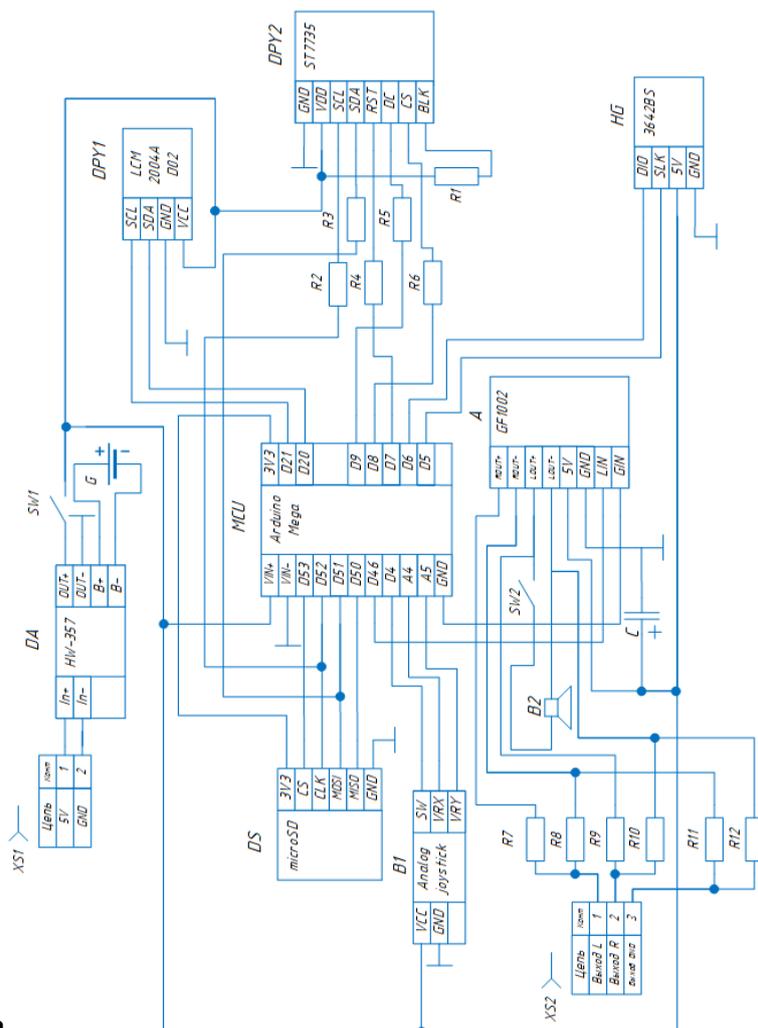


Приложение Б
Структурная схема
устройства



Приложение В
Электрическая принципиальная схема

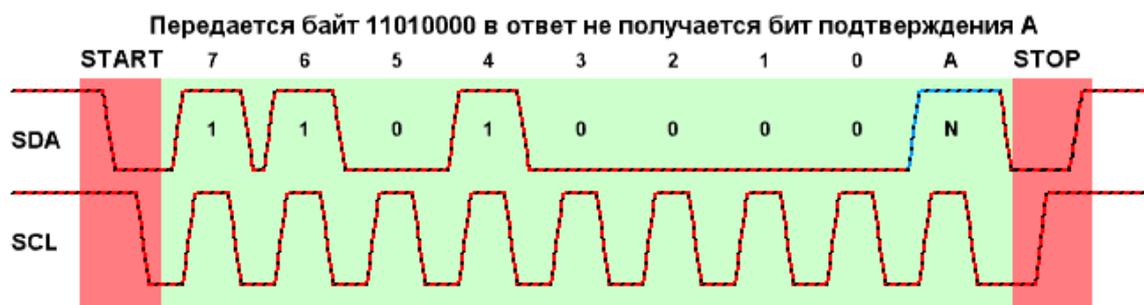
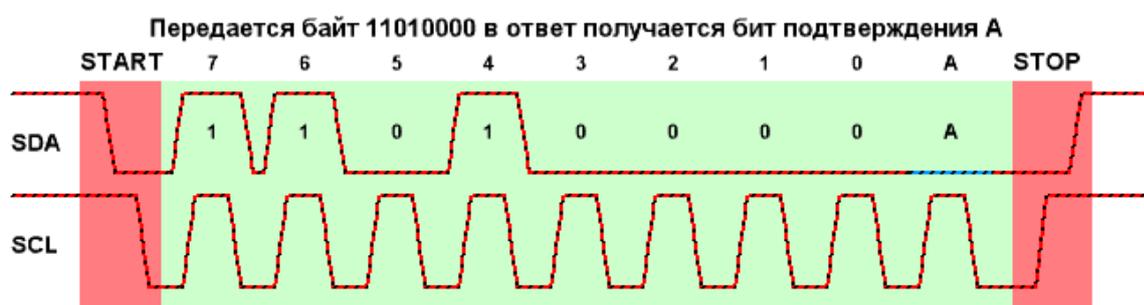
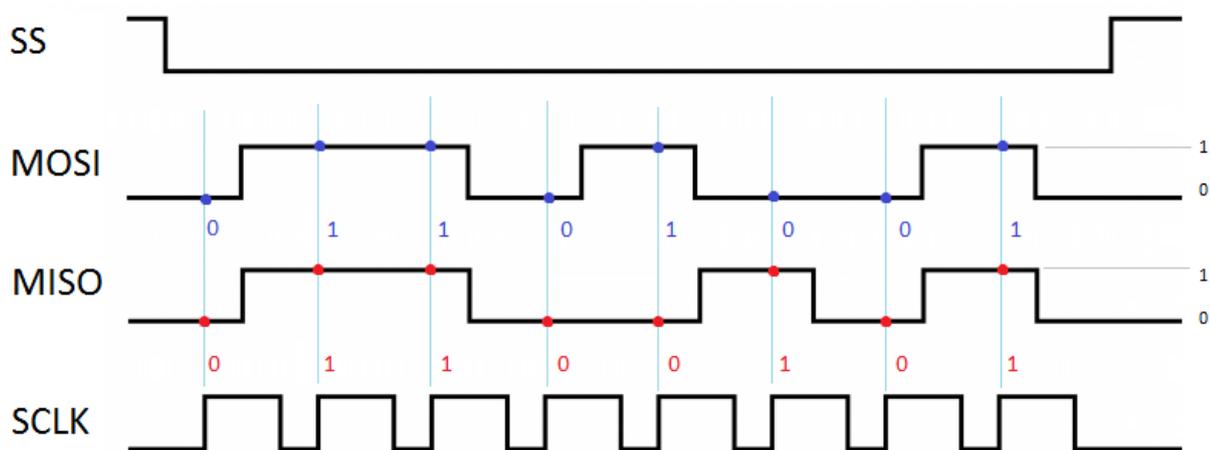
устройства



Поз обозначение	Наименование	Кол.	Примечание
A	Плата усилитель GF002	1	
B1	Модуль аналоговый джойстик	1	
B2	Динамик 8 Ом 3 Вт	1	
C	Конденсатор ECAP (K50-35), 1000 мкФ, 6.3 В	1	
DA	Плата HW-357	1	
DPY1	Модуль дисплей LCM 2004-A D02	1	
DPY2	Модуль дисплей TFT-ST7735	1	
DS	Модуль microSD	1	
G	Аккумулятор Samsung ICR18650	1	
HG	Модуль семисегментный индикатор 3642BS	1	
MCU	Плата Arduino Mega	1	
R1..R6	Резистор MF-25 (C2-23) 0.25 Вт, 1 кОм, 1%	6	
R7..R10	Резистор MF-25 (C2-23) 0.25 Вт, 100 Ом, 1%	4	
R11..R12	Резистор MF-25 (C2-23) 0.25 Вт, 20 Ом, 1%	2	
SW1..SW2	Переключатель выжимной, КВВ10-2Р3W	2	ON-OFF-ON (1.5A 250V) DPDT 6P
XS1	Гнездо microUSB	1	
XS2	Гнездо TRS Mini Jack 3.5 мм	1	

Приложение Г

Временные диаграммы



Уровень определяет ведомый

Уровень определяет ведущий

easyelectronics.ru

Приложение Е

Программный код

```
#include <SPI.h>
#include <LCD_1602_RUS.h>
#include <SD.h>
#include <Wire.h>
#include <TM1637.h>

#include <TMRpcm.h>
#include <SPI.h>

#include <Adafruit_GFX.h>
#include <Adafruit_ST7735.h>

int bb;
int c;
int d;
int e;
int f;

#define TFT_RST 7 // reset line
#define TFT_CS 8 // chip select line
#define TFT_DC 9 // data/command line
#define BUFFPIXEL 20

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);
String picture = "";

int maxnumber = 0;
int foldname = 0;
int question = 0;//счетчик пройденных вопросов

TMRpcm tmrpcm;
String sound = "";
unsigned long time = 0;

LCD_1602_RUS lcd(0x26, 20, 4); //адрес текстового экрана

File myFile;
```

Продолжение Приложения Е

```
File entry;
String path = "";

#define CLK 5
#define DIO 6
TM1637 tm1637(CLK,DIO); //семисегментный индикатор

short int sum = 0; //счетчик пройденных категорий
short int ans = 0; //код цифры правильного ответа для проверки в файле с ответами
short int answer = 0; //переменная для записи кода цифры верного ответа из файла
//счетчики пройденных категорий:
short int onecat = 0;
short int twocat = 0;
short int threecat = 0;
short int fourcat = 0;
int win = 0; //единицы числа верных ответов
int bigwin = 0; //десятки числа верных ответов
int totalwin = 0; //число верных ответов
int lose = 0; //единицы числа неверных ответов
int biglose = 0; //десятки числа неверных ответов
int totallose = 0; //число неверных ответов
int getPressedButton();
void ButtonEventProc(int analogValue);
short int Mode = 0; //номер режима отрисовки того или иного меню
int currentButtonState, previousButtonState;
//все для джойстика:
int l_r,p,d_u;
int button;
const int UP = 1;
const int DW = 2;
const int EN = 3;
const int LF = 4;
const int RG = 5;
long randNumber; //случайно генерируемое число для воспроизведения случайных фраз
int bCount = 0; //счетчик строк для текстового экрана
```

Продолжение Приложения Е

```
void setup()
{
  tft.initR(INTR_BLACKTAB);
  tft.setTextWrap(false);
  tft.setRotation (1);
  tft.fillScreen(ST77XX_BLACK);

  SD.begin(53);
  delay(100);
  randomSeed(analogRead(0));
  delay(100);

  tmrpcm.speakerPin = 10;

  lcd.begin();
  lcd.backlight();
  lcd.clear();
  tm1637.init();
  tm1637.set(BRIGHT_TYPICAL);
  pinMode(A1,INPUT);
  pinMode(A2,INPUT);
  pinMode(10,INPUT);
  digitalWrite(10,HIGH);//Active low key

  lcd.clear();
  lcd.setCursor(0, 0);
}

void bmpDraw(char *filename, uint8_t x, uint16_t y) {

  File  bmpFile;
  int   bmpWidth, bmpHeight; // W+H in pixels
  uint8_t bmpDepth;          // Bit depth (currently must be 24)
  uint32_t bmpImageoffset;   // Start of image data in file
  uint32_t rowSize;          // Not always = bmpWidth; may have padding
```

Продолжение Приложения Е

```
uint8_t sdbuffer[3*BUFFPIXEL]; // pixel buffer (R+G+B per pixel)
uint8_t buffidx = sizeof(sdbuffer); // Current position in sdbuffer
boolean goodBmp = false; // Set to true on valid header parse
boolean flip = true; // BMP is stored bottom-to-top
int w, h, row, col;
uint8_t r, g, b;
uint32_t pos = 0, startTime = millis();

if((x >= tft.width()) || (y >= tft.height())) return;

Serial.println();
Serial.print(F("Loading image "));
Serial.print(filename);
Serial.println("");

// Open requested file on SD card
if ((bmpFile = SD.open(filename)) == NULL) {
  Serial.print(F("File not found"));
  return;
}

// Parse BMP header
if(read16(bmpFile) == 0x4D42) { // BMP signature
  Serial.print(F("File size: ")); Serial.println(read32(bmpFile));
  (void)read32(bmpFile); // Read & ignore creator bytes
  bmpImageoffset = read32(bmpFile); // Start of image data
  Serial.print(F("Image Offset: ")); Serial.println(bmpImageoffset, DEC);
  // Read DIB header
  Serial.print(F("Header size: ")); Serial.println(read32(bmpFile));
  bmpWidth = read32(bmpFile);
  bmpHeight = read32(bmpFile);
  if(read16(bmpFile) == 1) { // # planes -- must be '1'
    bmpDepth = read16(bmpFile); // bits per pixel
    Serial.print(F("Bit Depth: ")); Serial.println(bmpDepth);
    if((bmpDepth == 24) && (read32(bmpFile) == 0)) { // 0 = uncompressed
```

Продолжение Приложения E

```
goodBmp = true; // Supported BMP format -- proceed!
Serial.print(F("Image size: "));
Serial.print(bmpWidth);
Serial.print('x');
Serial.println(bmpHeight);

// BMP rows are padded (if needed) to 4-byte boundary
rowSize = (bmpWidth * 3 + 3) & ~3;

// If bmpHeight is negative, image is in top-down order.
// This is not canon but has been observed in the wild.
if(bmpHeight < 0) {
    bmpHeight = -bmpHeight;
    flip    = false;
}

// Crop area to be loaded
w = bmpWidth;
h = bmpHeight;
if((x+w-1) >= tft.width()) w = tft.width() - x;
if((y+h-1) >= tft.height()) h = tft.height() - y;

// Set TFT address window to clipped image bounds
tft.startWrite();
tft.setAddrWindow(x, y, w, h);

for (row=0; row<h; row++) { // For each scanline...

    // Seek to start of scan line.  It might seem labor-
    // intensive to be doing this on every line, but this
    // method covers a lot of gritty details like cropping
    // and scanline padding.  Also, the seek only takes
    // place if the file position actually needs to change
    // (avoids a lot of cluster math in SD library).

    if(flip) // Bitmap is stored bottom-to-top order (normal BMP)
```

```
pos = bmpImageoffset + (bmpHeight - 1 - row) * rowSize;
```

Продолжение Приложения E

```
else // Bitmap is stored top-to-bottom
  pos = bmpImageoffset + row * rowSize;
if(bmpFile.position() != pos) { // Need seek?
  tft.endWrite();
  bmpFile.seek(pos);
  buffidx = sizeof(sdbuffer); // Force buffer reload
}

for (col=0; col<w; col++) { // For each pixel...
  // Time to read more pixel data?
  if (buffidx >= sizeof(sdbuffer)) { // Indeed
    bmpFile.read(sdbuffer, sizeof(sdbuffer));
    buffidx = 0; // Set index to beginning
    tft.startWrite();
  }

  // Convert pixel from BMP to TFT format, push to display
  b = sdbuffer[buffidx++];
  g = sdbuffer[buffidx++];
  r = sdbuffer[buffidx++];
  tft.pushColor(tft.color565(r,g,b));
} // end pixel
} // end scanline
tft.endWrite();
Serial.print(F("Loaded in "));
Serial.print(millis() - startTime);
Serial.println(" ms");
} // end goodBmp
}

bmpFile.close();
if(!goodBmp) Serial.println(F("BMP format not recognized.));
}
```

Продолжение Приложения E

```
uint16_t read16(File f) {
    uint16_t result;
    ((uint8_t *)&result)[0] = f.read(); // LSB
    ((uint8_t *)&result)[1] = f.read(); // MSB
    return result;
}
```

```
uint32_t read32(File f) {
    uint32_t result;
    ((uint8_t *)&result)[0] = f.read(); // LSB
    ((uint8_t *)&result)[1] = f.read();
    ((uint8_t *)&result)[2] = f.read();
    ((uint8_t *)&result)[3] = f.read(); // MSB
    return result;
}
```

```
void printDirectory(File dir, int numTabs) {
    while (true) {

        File entry = dir.openNextFile();
        if (! entry) {
            // no more files
            break;
        }

        for (uint8_t i = 0; i < numTabs; i++) {
            Serial.print('\t');
        }
        Serial.print(entry.name());
        if (entry.isDirectory()) {
            Serial.println("/");
            printDirectory(entry, numTabs + 1);
        } else {
            // files have sizes, directories do not
            Serial.print("\t\t");
        }
    }
}
```

```
Serial.println(entry.size(), DEC);
```

Продолжение Приложения Е

```
    }  
    entry.close();  
}  
}  
  
void face(int inp){  
    if ((inp <= 6) && (inp >= 1)){  
        tft.fillScreen(ST77XX_DARKDARKCYAN);  
        if (inp == 3){bb = -7;}  
        else if (inp == 5){bb = 3;}  
        else if (inp == 6){bb = 3;}  
        else {bb = 0;}  
  
        //глазные яблоки  
        tft.fillRoundRect (15,65,50,20,10,ST77XX_WHITE); //  
        tft.fillRoundRect (95,65,50,20,10,ST77XX_WHITE); //  
        tft.fillCircle(40,77+bb,10,ST77XX_BLACK);  
        tft.fillCircle(120,77+bb,10,ST77XX_BLACK);  
        tft.fillCircle(35,80,3,ST77XX_WHITE); //  
        tft.fillCircle(115,80,3,ST77XX_WHITE); //  
  
        if (inp == 2){  
            bb = 2;  
            c = 1;  
            d = 0;  
            e = 0;  
            f = 0;  
        }  
        else if (inp == 3){  
            bb = 0;  
            c = 0;  
            d = 0;  
            e = 0;  
            f = 0;  
        }  
        else if (inp == 4){  
            bb = 0;  
            c = 0;
```

Продолжение Приложения Е

```
d = 2;

e = 1;
f = 0;
}
else if (inp == 5){
    bb = 0;
    c = 0;
    d = 0;
    e = 0;
    f = 1;
}
else if (inp == 6){
    bb = 0;
    c = 0;
    d = -2;
    e = 0;
    f = 5;
}
else {
    bb = 0;
    c = 0;
    d = 0;
    e = 0;
    f = 0;
}
//веки
tft.fillRect (15,80+bb+d,60,10-bb+d+f,ST77XX_DARKDARKCYAN);//
tft.fillRect (95,80+d,60,10+d+f,ST77XX_DARKDARKCYAN);
if (inp == 6){f = 4;}
else {f = 0;}
for (uint16_t a=0; a<3; a++)
{ tft.drawFastHLine(15, 80+a+c+e-f, 50, ST77XX_BLACK);}
for (uint16_t a=0; a<3; a++)
{ tft.drawFastHLine(95, 80+a-bb+e-f, 50, ST77XX_BLACK);}

for (uint16_t a=0; a<5; a++)
```

```
{ tft.drawLine(15, 80+a+c+e-f, 20, 85+a, ST77XX_BLACK);} //
```

Продолжение Приложения E

```
for (uint16_t a=0; a<5; a++)  
{ tft.drawLine(65, 80+a+c+e-f, 60, 85+a, ST77XX_BLACK);} //  
for (uint16_t a=0; a<5; a++)  
{ tft.drawLine(95, 80+a-bb+e-f, 100, 85+a, ST77XX_BLACK);} //  
for (uint16_t a=0; a<5; a++)  
{ tft.drawLine(145, 80+a-bb+e-f, 140, 85+a, ST77XX_BLACK);} //  
for (uint16_t a=0; a<3; a++)  
{ tft.drawFastHLine(20, 87+a, 40, ST77XX_BLACK);}  
for (uint16_t a=0; a<3; a++)  
{ tft.drawFastHLine(100, 87+a, 40, ST77XX_BLACK);}
```

```
//брови
```

```
if (inp == 2) {  
  bb = 2;  
  c = 2;  
  d = 0;  
  e = 0;  
  f = 0;  
}  
else if (inp == 3) {  
  bb = 0;  
  c = 0;  
  d = -2;  
  e = 0;  
  f = 0;  
}  
else if (inp == 4) {  
  bb = 3;  
  c = 0;  
  d = 0;  
  e = 3;  
  f = 3;  
}  
else if ((inp == 5) || (inp == 6)) {  
  bb = -3;
```

Продолжение Приложения Е

```
c = 0;

d = 0;
e = -3;
f = -3;
}
else {
    bb = 0;
    c = 0;
    d = 0;
    e = 0;
    f = 0;
}
for (uint16_t a=0; a<5; a++)
{ tft.drawFastHLine(15, 95+a+c+d+e, 30, ST77XX_BLACK);}
for (uint16_t a=0; a<5; a++)
{ tft.drawFastHLine(45, 95+a+d+f, 25, ST77XX_BLACK);}
for (uint16_t a=0; a<5; a++)
{ tft.drawFastHLine(90, 95+a+bb+d, 55, ST77XX_BLACK);}
for (uint16_t a=0; a<10; a++)
{ tft.drawFastVLine(60+a, 95+d+e, 8, ST77XX_BLACK);}
for (uint16_t a=0; a<10; a++)
{ tft.drawFastVLine(90+a, 95+c+d+e, 8, ST77XX_BLACK);}

//очки
if ((inp >= 2) || (inp <= 6)) {
    bb = -3;
}
else {bb = 0;}
tft.fillRoundRect (10,55+bb,60,20,5,ST77XX_DARKCYAN);
tft.fillRoundRect (90,55+bb,60,20,5,ST77XX_DARKCYAN);
tft.fillRect (69,70+bb,22,3,ST77XX_GRAY);

//нос
for (uint16_t a=0; a<3; a++)
{ tft.drawFastVLine(84+a, 55, 10, ST77XX_BLACK);} //
for (uint16_t a=0; a<3; a++)
```

```
{ tft.drawFastVLine(73+a, 55, 10, ST77XX_BLACK);} //
```

Продолжение Приложения E

```
//pot  
if ((inp == 4) || (inp == 6))  
{  
  bb = 3;  
  e = 0;  
  c = 0;  
  if (inp == 6){c = -3;e = 10;}  
  tft.fillRect (20,25,120,15+c,ST77XX_BLACK);  
  tft.fillTriangle(38,34,42,27,33,27,ST77XX_WHITE);  
  tft.fillTriangle(123,34,118,27,127,27,ST77XX_WHITE);  
  tft.fillRect (45,27,70,3,ST77XX_WHITE);  
}  
else if (inp == 5) {  
  bb = 3;  
}  
else {bb = 0;c = 0;e = 0;}  
tft.fillTriangle(20,25-bb,30,40-bb,10,40-bb,ST77XX_WHITE);  
tft.fillTriangle(140,25-bb,150,40-bb,130,40-bb,ST77XX_WHITE);  
for (uint16_t a=0; a<5; a++)  
{ tft.drawFastHLine(10, 40+a-bb, 20, ST77XX_BLACK);}  
for (uint16_t a=0; a<5; a++)  
{ tft.drawFastHLine(130, 40+a-bb, 20, ST77XX_BLACK);}  
  
for (uint16_t a=0; a<5; a++)  
{ tft.drawLine(140, 25+a-bb+c, 150, 40+a-bb+c, ST77XX_BLACK);}  
for (uint16_t a=0; a<5; a++)  
{ tft.drawLine(140, 25+a-bb+c, 130, 40+a-bb+c, ST77XX_BLACK);}  
for (uint16_t a=0; a<5; a++)  
{ tft.drawLine(20, 25+a-bb+c, 30, 40+a-bb+c, ST77XX_BLACK);}  
for (uint16_t a=0; a<5; a++)  
{ tft.drawLine(20, 25+a-bb+c, 10, 40+a-bb+c, ST77XX_BLACK);}  
for (uint16_t a=0; a<5; a++)  
{ tft.drawFastHLine(10+e, 40+a-bb, 140-0.9*e, ST77XX_BLACK);}
```

```
//борода
```

Продолжение Приложения E

```
if ((inp == 4) || (inp == 6)) {  
    bb = -15;  
}  
else if (inp == 5) {  
    bb = -3;  
}  
else {  
    bb = 0;  
}  
for (uint16_t a=0; a<3; a++)  
{ tft.drawFastHLine(70, 32+a+bb, 20, ST77XX_BLACK);}  
tft.fillTriangle(90,27+bb,70,27+bb,80,17+bb,ST77XX_BLACK);  
bb = 0;  
c = 0;  
d = 0;  
e = 0;  
f = 0;  
}  
else{ }  
}
```

```
char* string2char(String tring){  
    int len = tring.length()+1;  
    char *array_char = new char[len];  
    tring.toCharArray(array_char, len);  
    return array_char;  
}
```

```
String createPathtxt(int num, int num2) {  
    return String(num) + "/" + "txt" + "/" + String(num2) + ".txt";  
}
```

```
String createPathbmp(int num3, int num4) {  
    return String(num3) + "/" + "pic" + "/" + String(num4) + ".bmp";  
}
```

Продолжение Приложения Е

```
String createPathwav(int num5, int num6) {
    return String(num5) + "/" + "wav" + "/" + String(num6) + ".wav";
}

//определение направления джойстика
int getPressedButton()
{
    l_r=analogRead(A1);
    d_u=analogRead(A2);
    p=digitalRead(10);
    if (l_r > 800)
        return LF;
    else if (l_r < 100)
        return RG;
    else if (p==LOW)
        return EN;
    else if (d_u > 800)
        return DW;
    else if (d_u < 100)
        return UP;
    else return 0;
}

//функция обработчик события нажатой кнопки
void ButtonEventProc(int analogValue)
{
    switch(analogValue)
    {
    case UP:
        switch(Mode)
        {
            //если в корневом меню выбрали ИГРА - включаем сем. индикатор и начинаем тест
            case(1):
                tm1637.point(POINT_ON);
```

```
tm1637.display(0,0);
```

Продолжение Приложения Е

```
tm1637.display(1,0);
```

```
tm1637.display(2,0);
```

```
tm1637.display(3,0);
```

```
menu();
```

```
Mode = 2; // возвращение в соответствующий режим
```

```
break;
```

```
case(2):
```

```
if (onecat == 2)
```

```
{
```

```
    //если эта категория уже была, пишем что нельзя ее пройти
```

```
    //если не было то запускаем вопросы
```

```
    porp();
```

```
}
```

```
else{
```

```
letssee();
```

```
onecat = 1;
```

```
delay(500);
```

```
select();
```

```
Mode = 3;
```

```
}
```

```
break;
```

```
case (3):
```

```
ans = 49; //бит 49 соответствует единице
```

```
check(); //сравниваем данный ответ с правильным
```

```
back(); // возвращаемся к отрисовке следующих вариантов ответа
```

```
break;
```

```
}
```

```
break;
```

```
case DW:
```

```
switch(Mode)
```

```
{
```

```
case(1):
```

```
face(4);
```

```
lcd.clear();
```

Продолжение Приложения Е

```
lcd.setCursor(2,0);
```

```
// в корневом меню при выборе ВНИЗ пишем авторов
```

```
lcd.print("Каргин П.А.");
```

```
lcd.setCursor(4,1);
```

```
lcd.print("ЭЛб-1601а");
```

```
lcd.setCursor(5,2);
```

```
lcd.print("ТГУ 2020г.");
```

```
face(3);
```

```
delay(3000);
```

```
lcd.clear();
```

```
Mode = 1;
```

```
break;
```

```
case(2):
```

```
if (threecat == 2)
```

```
{
```

```
  nope();
```

```
}
```

```
else{
```

```
  letssee();
```

```
  threecat = 1;
```

```
  delay(500);
```

```
  select();
```

```
  Mode = 3;
```

```
}
```

```
break;
```

```
case (3):
```

```
  ans = 51;
```

```
  check();
```

```
  back();
```

```
break;
```

```
}
```

```
break;
```

```
case EN:
```

```
  switch(Mode)
```

Продолжение Приложения Е

```
{  
  
case(0):  
    // при нажатии на кнопку стика приветствуем и входим в меню  
    lcd.clear();  
    lcd.setCursor(7,0);  
    lcd.print("ЗВЕРЬ");  
    lcd.setCursor(5,1);  
    lcd.print("ПРОСЫПАЙСЯ");  
    face(6);  
    sound = createPathwav(0,1);  
    tmrpcm.play(string2char(sound));  
    while(tmrpcm.isPlaying());  
    face(4);  
    lcd.clear();  
    lcd.setCursor(7,0);  
    lcd.print("Зверь");  
    lcd.setCursor(2,1);  
    lcd.print("приветствует вас");  
    randNumber = random(2,24);  
    sound = createPathwav(0,randNumber);  
    tmrpcm.play(string2char(sound));  
    while(tmrpcm.isPlaying());  
    face(1);  
    lcd.clear();  
    Mode = 1;  
    break;  
}  
break;  
case LF:  
    switch(Mode)  
    {  
        case(1):  
            lcd.clear();  
            lcd.setCursor(0, 0);  
            lcd.print("Выбор вопросов по");  
            lcd.setCursor(0,1);
```

```
lcd.print("часовой.Сверху 1й.");
```

Продолжение Приложения Е

```
lcd.setCursor(0,2);
lcd.print("Выйти до конца");
lcd.setCursor(0,3);
lcd.print("теста нельзя.");
delay(3000);
lcd.clear();
Mode = 1;
break;
case(2):
  if (fourcat == 2)
  {
    nope();
  }
  else{
    letssee();
    fourcat = 1;
    delay(500);
    select();
    Mode = 3;
  }
  break;
case (3):
  ans = 52;
  check();
  back();
  break;
}
break;
case RG:
  switch(Mode)
  {
    case(1):
      // в корневом меню если нажать ВПРАВО - проверяется звук
      lcd.clear();
      lcd.setCursor(0, 0);
```

```
lcd.print("Зверь говорит");
```

Продолжение Приложения E

```
delay(1000);
```

```
face(4);
```

```
randNumber = random(24,32); // в этом диапазоне на носителе расположены фразы для проверки звука
```

```
sound = createPathwav(0,randNumber);
```

```
tmrpcm.play(string2char(sound));
```

```
while(tmrpcm.isPlaying());
```

```
face(1);
```

```
lcd.clear();
```

```
Mode = 1;
```

```
break;
```

```
case(2):
```

```
if (twocat == 2)
```

```
{
```

```
  nope();
```

```
}
```

```
else{
```

```
letssee();
```

```
twocat = 1;
```

```
delay(500);
```

```
select();
```

```
Mode = 3;
```

```
}
```

```
break;
```

```
case (3):
```

```
ans = 50;
```

```
check();
```

```
back();
```

```
break;
```

```
}
```

```
break;
```

```
}
```

```
}
```

```
// функция считывает названия категорий из файлов и выводит на экран
```

//если категория пройдена, вычеркиваем ее

Продолжение Приложения Е

```
void menu()
{
    face(3);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Выберите категорию");
    lcd.setCursor(0, 1);
    lcd.print("вопросов");
    sound = createPathwav(0,61);
    tmrpcm.play(string2char(sound));

    while(tmrpcm.isPlaying());
    lcd.clear();
    lcd.setCursor(0, 0);
    if (onecat == 0)
    {
        foldname = 1;
        path = createPathtxt(foldname,question);
        if (SD.exists(string2char(path)) == true){
            myFile = SD.open(string2char(path));
            printcateg();
        }
        else{lcd.print("---");}
    }
    else if (onecat == 2)
    {
        lcd.print("---");
    }
    lcd.setCursor(0, 1);
    if (twocat == 0)
    {
        foldname = 2;
        path = createPathtxt(foldname,question);
        if (SD.exists(string2char(path)) == true){
            myFile = SD.open(string2char(path));
```

```

    printcateg();

}
else{lcd.print("---");}
}
else if (twocat == 2)
{
    lcd.print("---");
}
lcd.setCursor(0, 2);
if (threecat == 0)
{
    foldname = 3;
    path = createPathtxt(foldname,question);
    if (SD.exists(string2char(path)) == true){
        myFile = SD.open(string2char(path));
        printcateg();
    }
    else{lcd.print("---");}
}
else if (threecat == 2)
{
    lcd.print("---");
}
lcd.setCursor(0, 3);
if (fourcat == 0)
{
    foldname = 4;
    path = createPathtxt(foldname,question);
    if (SD.exists(string2char(path)) == true){
        myFile = SD.open(string2char(path));
        printcateg();
    }
    else{lcd.print("---");}
}
else if (fourcat == 2)
{

```

Продолжение Приложения Е

```
lcd.print("---");
```

Продолжение Приложения E

```
}
```

```
}
```

```
// функция для проверки правильности ответа и вывода баллов на индикатор
```

```
void check()
```

```
{
```

```
    myFile.seek(0);
```

```
    myFile.peek();
```

```
    answer = myFile.read();
```

```
    myFile.close();
```

```
    delay(100);
```

```
    if (answer == ans)
```

```
    {
```

```
        face(4);
```

```
        lcd.clear();
```

```
        lcd.setCursor(5, 0);
```

```
        lcd.print("Все верно");
```

```
        randomNumber = random(32,42);
```

```
        sound = createPathwav(0,randomNumber);
```

```
        tmrpcm.play(string2char(sound));
```

```
        while(tmrpcm.isPlaying());
```

```
        delay(1000);
```

```
        win++;
```

```
        if (win == 10)
```

```
        {
```

```
            bigwin++;
```

```
            totalwin = totalwin + 10;
```

```
            tm1637.display(0,bigwin);
```

```
            tm1637.display(1,0);
```

```
            win = 0;
```

```
        }
```

```
        else
```

```
        {
```

```
            tm1637.display(1,win);
```

Продолжение Приложения Е

```
    }  
  
    }  
    else  
    {  
        face(5);  
        lcd.clear();  
        lcd.setCursor(6, 0);  
        lcd.print("Неверно");  
        randomNumber = random(42,54);  
        sound = createPathwav(0,randomNumber);  
        tmrpcm.play(string2char(sound));  
  
        while(tmrpcm.isPlaying());  
        delay(1000);  
        lose++;  
        if (lose == 10)  
        {  
            biglose++;  
            totallose = totallose + 10;  
            tm1637.display(2,biglose);  
            tm1637.display(3,0);  
            lose = 0;  
        }  
        else  
        {  
            tm1637.display(3,lose);  
        }  
    }  
    lcd.clear();  
    lcd.setCursor(0, 0);  
}  
  
// функция вывода вариантов ответа с носителя на экран  
void wans()  
{  
    myFile.seek(1);
```

```
lcd.setCursor(0,0);
```

Продолжение Приложения Е

```
while (myFile.available()) {
  int incoming = myFile.read();
  //bCount считает строки и если считывается знак ";" строка переносится
  if((bCount == 2) && (incoming == 59))
  {
    lcd.setCursor(0,3);
  }
  if((bCount == 1) && (incoming == 59))
  {
    bCount++;
    lcd.setCursor(0,2);
  }
  if((bCount == 0) && (incoming == 59))
  {
    bCount++;
    lcd.setCursor(0,1);
  }
  if (incoming != 59)
  {
    lcd.print(lcd.asciiutf8(incoming));
  }
  if (incoming == 59)
  {
    //если считывается знак ";" - ничего не печатаем
  }
  }
  bCount = 0;
  Mode = 3;
}

// функция для перекодировки поступающего сигнала
//чтобы можно было читать русские символы
void printcateg()
{
  while (myFile.available()) {
```

```
lcd.print(lcd.asciiutf8(myFile.read()));
```

Продолжение Приложения Е

```
    }  
    myFile.close();  
}  
  
// проверяем какая категория выбрана и печатаем оттуда вопросы  
void select()  
{  
    face(1);  
    if (onecat == 1)  
    {  
        foldname = 1;  
    }  
    else if (twocat == 1)  
    {  
        foldname = 2;  
    }  
    else if (threecat == 1)  
    {  
        foldname = 3;  
    }  
    else if (fourcat == 1)  
    {  
        foldname = 4;  
    }  
    question++;  
    path = createPathtxt(foldname,question);  
    if (SD.exists(string2char(path)) == true)  
    {  
        picture = createPathbmp(foldname,question);  
        if (SD.exists(string2char(picture)) == true){  
            entry = SD.open(string2char(picture));  
            uint8_t nameSize = String(entry.name()).length(); // get file name size  
            String str1 = String(entry.name()).substring( nameSize - 4 ); // save the last 4 characters (file extension)  
  
            if ( str1.equalsIgnoreCase(".bmp") ) // if the file has '.bmp' extension
```

Продолжение Приложения Е

```
    bmpDraw(entry.name(), 0, 0);    // draw it

    entry.close(); // close the file
}
sound = createPathwav(foldname,question);
tmrpcm.play(string2char(sound));

while(tmrpcm.isPlaying());
myFile = SD.open(string2char(path));
wans();
}
else {
    question--;
    maxnumber = question;
    back();}
}

//проверяем результат теста
//выносим вердикт
//возвращаемся к началу программы
void back()
{
    if (question == maxnumber)
    {
        sum++;
        if (sum == 4)
        {
            sum = 0;
            lcd.clear();
            face(3);
            sound = createPathwav(0,61);
            tmrpcm.play(string2char(sound));

            while(tmrpcm.isPlaying());
            totalwin = totalwin + win;
            totallose = totallose + lose;
```

Продолжение Приложения Е

```
if (totalwin > totallose)
{
    face(4);
    lcd.setCursor(3,0);
    lcd.print("Верных ответов");
    lcd.setCursor(5,1);
    lcd.print("больше 50%");
    randomNumber = random(64,66);
    sound = createPathwav(0,randomNumber);
    tmrpcm.play(string2char(sound));

    while(tmrpcm.isPlaying());
}
if (totalwin < totallose)
{
    face(5);
    lcd.setCursor(3,0);
    lcd.print("Верных ответов");
    lcd.setCursor(5,1);
    lcd.print("меньше 50%");
    randomNumber = random(65,67);
    sound = createPathwav(0,randomNumber);
    tmrpcm.play(string2char(sound));

    while(tmrpcm.isPlaying());
}
if (totalwin == totallose)
{
    face(2);
    lcd.setCursor(3,0);
    lcd.print("Верных ответов");
    lcd.setCursor(5,1);
    lcd.print("ровно 50%");
    randomNumber = random(62,64);
    sound = createPathwav(0,randomNumber);
    tmrpcm.play(string2char(sound));
```

Продолжение Приложения Е

```
    while(tmrpcm.isPlaying());
}
question = 0;
win = 0;
bigwin = 0;
totalwin = 0;
lose = 0;
biglose = 0;
totallose = 0;
onecat = 0;
twocat = 0;
threecat = 0;
fourcat = 0;
tm1637.point(POINT_OFF);
lcd.clear();
lcd.setCursor(3,0);
lcd.print("Зверь засыпает");
face(6);
randNumber = random(54,59);
sound = createPathwav(0,randNumber);
tmrpcm.play(string2char(sound));

while(tmrpcm.isPlaying());
tft.fillScreen(ST77XX_BLACK);
lcd.clear();
Mode = 0;
}
else if (sum != 4)
{
if (onecat == 1)
{
    onecat = 2;
}
if (twocat == 1)
{
```

Продолжение Приложения Е

```
    twocat = 2;
}
if (threecat == 1)
{
    threecat = 2;
}
if (fourcat == 1)
{
    fourcat = 2;
}
menu();
question = 0;
Mode = 2;
}
}
else
{
    select();
}
}
```

//в случае выбора уже пройденной категории, говорит что нельзя, возвращаемся к выбору категории

```
void porp()
{
    face(2);
    lcd.clear();
    lcd.setCursor(2, 0);
    lcd.print("Пройти категорию");
    lcd.setCursor(7, 1);
    lcd.print("нельзя");
    sound = createPathwav(0,60);
    tmrpcm.play(string2char(sound));

    while(tmrpcm.isPlaying());
    lcd.clear();
    menu();
}
```

Продолжение Приложения Е

```
    Mode = 2;
}

//выводится лицо и произносится фраза ТАК ПОСМОТРИМ
void letssee()
{
    face(3);
    sound = createPathwav(0,34);
    tmrpcm.play(string2char(sound));

    while(tmrpcm.isPlaying());
    lcd.clear();
    lcd.setCursor(0, 0);
}

void loop()
{
    //инструкции подавления дребезга контактов кнопки
    currentButtonState = getPressedButton();
    if(currentButtonState != previousButtonState)
    {
        delay(5);
        currentButtonState = getPressedButton();
        if(currentButtonState != previousButtonState)
        {
            ButtonEventProc(currentButtonState);
            previousButtonState = currentButtonState;
        }
    }
    switch(Mode)
    {
        //точка начала отрисовки основного экрана
        case(0):
            //инструкции для отображения отрисовки начального экрана
            lcd.setCursor(6,0);
            lcd.print("Разбуди");
```

Продолжение Приложения Е

```
lcd.setCursor(7,1);
lcd.print("Зверя");
lcd.setCursor(4,2);
lcd.print("(нажми стик)");
break;
//режим отрисовки корневого меню
case(1):
  lcd.setCursor(0,0);
  lcd.print("Начать тест");
  lcd.setCursor(0,1);
  lcd.print("Проверить звук");
  lcd.setCursor(0,2);
  lcd.print("Авторы проекта");
  lcd.setCursor(0,3);
  lcd.print("Стик влево-правила");
  break;
case(2):
  //меню отрисовки выбора категории
  break;
case(3):
  //меню отрисовки и выбора ответа
  break;
}
}
```