

Реализация отображения текста на китайском языке с использованием микроконтроллера ATmega328

Болтовский Гавриил Александрович

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

Целью данной статьи является создание устройства, способного выводить произвольный текст на китайском языке. Устройство способно выводить иероглифический текст, хранящийся на SD карте, на дисплей разрешением 128×64 пикселей, при этом, включена поддержка 131070 различных символов китайского алфавита. Результатом исследования станет устройство, с подробным описанием его создания.

Ключевые слова: встроенная разработка, ардуино, микроконтроллер

Implementation of Chinese text display using ATmega328 microcontroller

Boltovskiy Gavriil Aleksandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to create a device capable of outputting arbitrary text in Chinese. The device is capable of displaying hieroglyphic text stored on an SD card on a 128×64 pixel resolution display, while supporting 131,070 different Chinese characters. The result of the study will be a device with a detailed description of its creation.

Keywords: Embedded development, Arduino, microcontroller

1. Введение

1.1 Актуальность исследования

Вывод на дисплей, подключённому к микроконтроллеру, латинских и кириллических символов не является сложной задачей, так как все символы занимают мало места в памяти микроконтроллера. Начертание каждого символа может храниться так же в памяти дисплея (подобное реализовано в дисплеях LCD1602). Символы китайского алфавита невозможно сохранить в памяти микроконтроллер в силу их количества.

Отдельную сложность представляет создание начертания каждого иероглифа, подходящего для дальнейшего отображения на дисплее с малым разрешением.

Перечисленные ограничения требуют особых подходов к реализации отображения текста на китайском языке.

1.2 Обзор исследований

Проблема отображения китайских символов с использованием микроконтроллера слабо изучена.

Отображение графической информации исследовалось Г. Г. Громовым [1]. Им затронут вопрос хранения информации на низком уровне. Общие принципы работы с графикой изложены в исследовании А. И. Незванова на примере RGB матрицы [2]. Алгоритмы преобразования закодированного текста в текстовую информацию на дисплее, подключённому к микроконтроллеру изучалось Н. Н. Кошелева в её статье [3]

1.3 Цель исследования

Целью исследования является создание устройства, способного отображать произвольный текст на китайском языке.

1.4 Постановка задачи

Для достижения поставленной цели было собрано устройство на базе микроконтроллера, реализованы элементы управления, подключены внешние запоминающее устройство и дисплей. Каждый символ китайского алфавита был преобразован в формат, подходящий для работы с микроконтроллером. Были созданы программные решения, реализующие отображение и хранение текстовой информации.

2. Методы исследования

2.1 Реализация аппаратной части

Микроконтроллер выбран ATmega328, используемый в платах семейства Arduino. Данный микроконтроллер имеет 32 килобайта флеш-памяти и 2 килобайта оперативной памяти [4]. Для реализации элементов управления используется аналоговая клавиатура. Отображение производится на OLED дисплей, подключаемый по интерфейсу I2C и имеющий разрешение 128×64 пикселя. Внешнем носителем является SD карта, её подключение происходит по интерфейсу SPI.

Модуль аналоговой клавиатуры построен по схеме, представленной в GitHub репозитории библиотеки, позволяющей работать с ней [5].

Дисплей подключается к аппаратному I2C контроллера, то есть SDA и SCL подключаются к A4 и A4 выводам соответственно.

Метод подключения SD карты по интерфейсу SPI описан в [6].

Для проверки элементов собранного устройства использовалась специальная отладочная прошивка «for_check». из репозитория проекта [7]. В ней инициализируется SD карта, дисплей и элементы управления. Нажатия на клавиши аналоговой клавиатуры отображаются на дисплее и сохраняются на SD карте файле «DATA.TXT».

В отладочной прошивке используются библиотека «U8glib.h» [8], обладающая всем необходимым функционалом для работы с дисплеем. «SPI.h» и «SD.h» позволяют работать с SD картой, создавать файлы,

производить запись и чтение. «EncButton.h» [9] и «AnalogKey.h» реализуют обработку нажатий на аналоговой клавиатуре.

2.2 Создание специального шрифта

Установлено, что для отображения любого символа кириллического или латинского алфавита (а также цифр и специальных знаков) достаточно сетки 5 на 7 пикселей, то есть один символ занимает 5 байтов памяти (35 бита). Два алфавита будут занимать 290 байт, их можно записать в память микроконтроллера. Минимальная сетка для китайских иероглифов составляет 16 на 16 пикселей. Значит начертание одного иероглифа займёт в памяти 32 байта (256 бит). Восемь тысяч основных иероглифов займут 256000 байт (250 килобайт), загрузить такой объём в память микроконтроллера не представляется возможным.

Для решения проблемы нехватки места, было решено хранить данные о символах на SD карте.

Отрисовка 8 тысяч иероглифов является трудоёмкой задачей, поэтому было решено растривать существующий векторный шрифт таким образом, чтобы он уместился в сетку 16 на 16.

Скрипт для растеризации шрифта создав в программной среде Processing [10] и представлен в GitHub репозитории проекта. В коде скрипта необходимо указать название шрифта и блок Юникода (рис. 1).

```
137 int fontSize = 16;
138
139
140 int displayFontSize = 16;
141
142
143 static final int[] unicodeBlocks = {
144
145     0x31C0, 0x31EF,
146     0x31F0, 0x31FF,
147     0x3200, 0x32FF,
148     0x3300, 0x33FF,
149     0x3400, 0x4DBF,
150     0x4DC0, 0x4DFF,
151     0x4E00, 0x9FFF,
152
153 };
```

Рисунок 1 – Скрипт для растеризации шрифта

Шрифт для растеризации выбран «STXINGKA». Китайские иероглифы находятся блоке юникода по адресу 0x31C0 до 0x9FFF. Полученный файл с расширением «.v1w» следует представить в виде битового изображения. Для этого использовался онлайн инструмент [11].

Полученные данные сохранены в «font.txt» сохранён на SD карте.

2.3 Реализация программной части

Микроконтроллер ATmega328 обладает лишь одним вычислительным ядром. Поставленная задача подразумевает параллельный доступ к файлам на SD карте. Алгоритмы, позволяющая реализовать мультизадачность единственном вычислительном ядре, называются псевдомультитзадачными. В данном случае параллельный доступ к файлам осуществляется с помощью мультицикла, в котором доступ к файлам будет поочерёдным, но благодаря высокой скорости работы пользователю не будет этого заметно.

Работа прошивки построена следующим образом: файл с текстом на китайском языке открывается на чтение, так как дисплей может уместить текст, состоящий только из двадцати четырёх иероглифов за раз, то создаётся массив, содержащий адреса каждого символа (по три байта на символ), затем с этой информацией происходит обращение к файлу со шрифтом, откуда элементы выгружаются на дисплей. При прокрутке текста снова открывается файл, содержащий текст, и в массив адресов записываются новые значения. Ознакомиться с файлом прошивки можно в репозитории проекта на GitHub.

Результат представлен на рисунке (рис. 2).



Рисунок 1 – Вывод текста на китайском языке

3. Выводы

Таким образом, было создано устройство, позволяющее выводить текст на китайском языке на дисплей с поддержкой всех иероглифов. Описанные методы могут быть использованы и отображения символов из других языков в условиях ограниченной памяти микроконтроллера.

Библиографический список

1. Громов Г. Г. Графические технологии разработки программного обеспечения для микроконтроллеров // Компоненты и Технологии. 2001. №10.
2. Незванов, А. И. Разработка схемы и программирование микроконтроллера Atmel attiny45 для управления RGB-матрицей // Сборник научных трудов Новосибирского государственного технического университета. 2016. № 4(86). С. 144-159.
3. Кошелева Н. Н. Устройство формирования опознавательных сигналов на основе микроконтроллера // Вестник Воронежского государственного технического университета. 2017. Т. 13, № 6. С. 69-73.
4. Arduino Uno URL: <https://arduino.ru/Hardware/Arduinoboarduno> (дата обращения: 29.06.2023)
5. Github. URL: <https://github.com/GyverLibs/GyverButton/> (дата обращения: 29.06.2023)
6. Михайлов В. Г. Запись бинарных данных на SD карту Arduino Due // Системный анализ и прикладная информатика. 2016. № 3. С. 40-49.
7. Github. URL: <https://github.com/Gavriilbolt/ChineseCharacters> (дата обращения: 29.06.2023)
8. Github. URL: <https://github.com/olikraus/u8glib> (дата обращения: 29.06.2023)
9. Github. URL: <https://github.com/GyverLibs/EncButton/> (дата обращения: 29.06.2023)
10. Processing. URL: <https://processing.org/> (дата обращения: 29.06.2023)
11. File to hexadecimal converter. URL: https://tomeko.net/online_tools/file_to_hex.php (дата обращения: 29.06.2023)