Содержание

1.2 Разработка принципиальной схемы

1.2.1 Драйвер дисплея (SN74LS48)

Для упрощения управления ССИ в составе дисплея, а также для сокращения числа использованных контактов микроконтроллера был использован драйвер SN74LS48, преобразующий 4 бита двоично-десятичного кода (один десятичный разряд) в сигналы для сегментов ССИ [3].

Условное графическое обозначение драйвера представлено на рисунке 4. (на изображении устройство отражено относительно вертикальной оси, как это сделано для удобства расположения на принципиальной схеме)

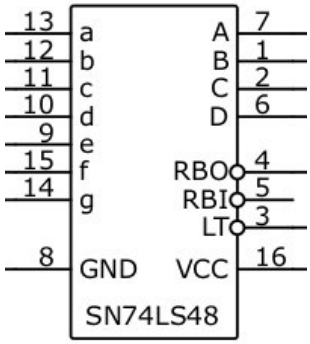

Рисунок 4 — условное графическое обозначение SN74LS48

Назначение контактов микросхемы:

Входы A, B, C, D — разряды двоично-десятичного числа.

Выходы a, b, ..., g — сигналы для сегментов ССИ.

Вход LT — тест светодиодов (активация всех светодиодов).

Вход RBI — не отображать 0.

Выход RBO — выключено отображение 0.

Состояние ССИ в зависимости от входного числа показано на рисунке 5.

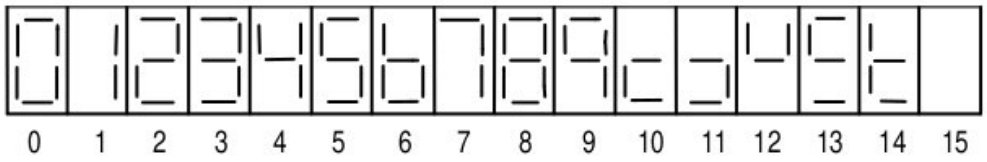

Рисунок 5 — состояния ССИ

1.2.2 Расчет транзисторных ключей

Так как дешифраторы, используемые для выбора активного светодиода в матрице имеют малые значения предельного выходного тока, для питания светодиодов необходимо использовать транзисторные ключи.

Схема такого с использованием n-p-n транзистора приведена на рисунке 6.

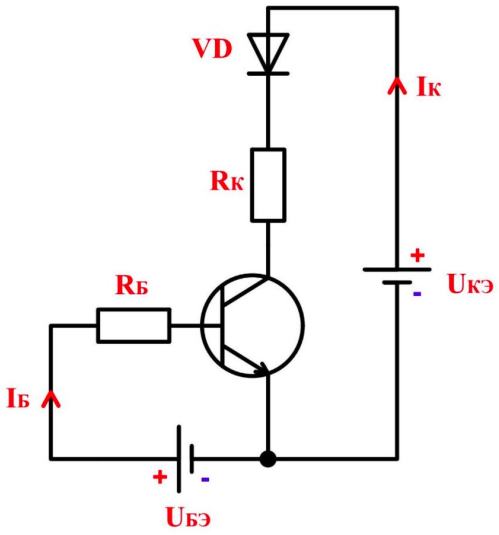

Рисунок 6 — транзисторный ключ

В основе транзисторного ключа используется транзистор 2N2222, который имеет коэффициент усиления по току $\beta = 100..300$, $\Delta U_{кэ} = 0.1$ и $\Delta U_{бэ} = 0.6$ [4].

Резистор со стороны коллектора ограничивает ток, протекающий в светодиоде ($I_{кэ}$ = 20 мА = 0.02 А), его номинал рассчитывается по формуле:

$R_к = (U_{vcc} - U_{vd} - \Delta U_{кэ}) / I_{кэ} = (5\text{-}2.5\text{-}0.1) / 0.02 = 120$ Ом

где U_{vcc} — напряжение питания;

U_{vd} — падение напряжения на светодиоде;

$\Delta U_{кэ}$ — падение напряжения на переходе коллектор-эммитер;

$I_к$ — предельный ток в цепи коллектора (ток через светодиод).

Номинал резистора со стороны базы рассчитывается по формуле:

$R_б = (U_{vcc} - \Delta U_{бэ}) / I_б = (U_{vcc} - \Delta U_{бэ}) / (I_{кэ} / \beta) = 4.4 / (0.02 / 100) = 22000$ Ом

где $\Delta U_{бэ}$ — падение напряжения на переходе база-эммитер;

$I_б$ — предельный ток в цепи базы;

2.2 Программирование микроконтроллера

В составе микроконтроллеров AVR наряду с Flash-памятью программ имеется энергонезависимая память для хранения данных EEPROM, которая также может быть запрограммирована перед началом работы. Микроконтроллеры этого семейства допускают более широкие возможности при программировании и верификации.

В процессе программирования микроконтроллеров AVR осуществляют:

- запись команд и констант в Flash-память программ;
- запись данных в память EEPROM;
- запись конфигурационных битов (fuse bits);
- запись битов защиты (lock bits).

В ходе этих работ можно дополнительно выполнить следующие операции:

- стирание памяти микроконтроллера;
- чтение ячеек Flash-памяти программ и памяти данных EEPROM для контроля правильности записанной информации (верификация);
- чтение конфигурационных ячеек для определения состояния микроконтроллера;
- чтение битов защиты;
- чтение ячеек идентификатора для определения типа микроконтроллера и ячейки калибровочного байта при настройке внутреннего RC-генератора.

Программирование микроконтроллеров AVR в зависимости от применяемого класса и типа модели может быть выполнено четырьмя способами:

- последовательное программирование при высоком напряжении питания (+12 В);
- параллельное программирование при высоком напряжении;

выполняется по нарастающему фронту сигнала SCK, а «защелкивание» выходных данных — по спадающему.

Программирование памяти программ микроконтроллеров семейства Mega осуществляется постранично. Сначала содержимое страницы побайтно заносится в буфер по командам «Загрузка страницы Flash-памяти». При этом в каждой команде указываются младшие разряды адреса записываемой ячейки (положение ячейки на странице) и записываемое значение. Каждая ячейка памяти загружается в следующей последовательности: сначала младший байт, затем — старший. Фактическое программирование страницы Flash-памяти выполняется после завершения загрузки буфера по команде «Запись страницы Flash-намяти». В команде указываются старшие разряды адреса, определяющие номер страницы.

Программирование памяти данных EEPROM осуществляется с помощью команд «Запись в память EEPROM». В каждой команде указывается адрес записываемой ячейки и записываемое значение. Для большинства микроконтроллеров можно также программировать конфигурационные биты (fuse) для выбора настроек тактирования, длительности задержки при старте и порога детектора понижения напряжения (BOD).

Команды, используемые для программирования ATmega8535 показаны в таблице 8.

Таблица 8 — команды программирования ATmega8535

Инструкция	Формат инструкции			
	Байт 1	Байт 2	Байт 3	Байт 4
Разрешить программирование	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx
Стереть чип	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx
Прочитать из памяти программ	0010 H000	0000 aaaa	bbbb bbbb	oooo oooo