



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)
НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 Прикладная информатика

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

По дисциплине «Микропроцессорные системы»

НА ТЕМУ:

***МК-система управления приборами жилого
помещения***

Студент

ИУ6-75Б

(Группа)

(Подпись, дата)

В.Д. Шульман

(И.О. Фамилия)

Руководитель

(Подпись, дата)

В.Я. Хартов

(И.О. Фамилия)

Оглавление

Реферат	4
Обозначения и сокращения	5
Введение	6
Основная часть	7
1 Конструкторская часть	8
1.1 Проектирование МК-системы.....	8
1.1.1 Описание структурно-функциональной схемы микроконтроллерной системы.....	8
1.1.2 Выбор микроконтроллера	10
1.1.3 Описание архитектуры и технических характеристик микроконтроллера.....	11
1.1.4 Распределение адресного пространства ATmega8515	13
1.1.5 Особенности системы команд микроконтроллера ATmega8515 ..	14
1.2 Разработка функциональной схемы	15
1.2.1 Используемые модули ATmega8515	15
1.2.2 Пульт оператора	15
1.2.3 Блок передачи данных к ПЭВМ	18
1.2.4 Блок реле	19
1.2.5 LCD-дисплей.....	20
1.3 Разработка принципиальной схемы.....	23
1.3.1 Синтез принципиальной схемы	23
1.3.2 Конфигурация выводов микроконтроллера	23
1.3.3 Схема понижения входного напряжения до 5В.....	24
1.3.4 Подключение матричной клавиатуры.....	25
1.3.5 Подключение LCD-дисплея	27
1.4 Расчёт потребляемой мощности	28
1.5 Описание алгоритмов функционирования устройства.....	29
2 Технологическая часть	36
2.1 Характеристика использованных систем разработки.....	36
2.2 Оценка количества задействованной памяти микроконтроллера ATmega8515.....	36
2.3 Симуляция в Proteus 8	37
2.4 Способы программирования памяти микроконтроллера ATmega8515 39	
2.4.2 Алгоритм последовательного программирования через SPI.....	40

2.4.3	Опрос данных Flash памяти	44
2.4.4	Опрос данных EEPROM	44
Заключение		46
Список литературы		47

Реферат

Расчётно-пояснительная записка с. 42, рис. 18, табл. 6, источников 4, приложений 2.

МИКРОПРОЦЕССОР, МИКРОКОНТРОЛЛЕР, ATMEGA8515, СЕРВЕР РАСПИСАНИЙ, ПУЛЬТ ОПЕРАТОРА, UART, ТАЙМЕР, ДИСПЛЕЙ.

Объектом разработки курсовой работы является устройство управления приборами, получающее расписание их работы с удаленного сервера.

Цель работы – создание полного комплекса конструкторской документации для устройства управления приборами, создание программного обеспечения для микроконтроллера семейства AVR/

При проектировании решены следующие задачи:

- анализ объекта разработки на функциональном уровне;
- разработка функциональной схемы;
- выбор элементной базы для реализации объекта;
- разработка принципиальной схемы;
- расчет потребляемой мощности;
- разработка алгоритмов работы микроконтроллера;
- написания программного обеспечения для микроконтроллера.

Результатом проектирования является комплекс конструкторской документации для изготовления устройства, исходные коды программ для программирования памяти микроконтроллера.

Спроектированное устройство обладает следующими характеристиками:

- 1) управление до 8 приборами одновременно;
- 2) хранение до 127 записей для включения/выключения устройств;
- 3) получение актуального расписания и времени по UART;
- 4) управление устройствами вручную через пульт оператора;
- 5) установка текущего времени и расписания по умолчанию;
- 6) вывод текущего времени МК-системы на LCD-дисплей

Обозначения и сокращения

МК – микропроцессор

ПЭВМ – персональная электронно-вычислительная машина

СР – сервер расписания

РОН – регистры общего назначения

АЛУ – арифметико-логическое устройство

ПЗУ – постоянно запоминающее устройство

EEPROM – (Electrically Erasable Programmable Memory) электрически стираемое программируемое ПЗУ

Flash – перепрограммируемая память для хранения программ

РС – (Program Counter) программный счетчик

SREG – (Status Register) регистр статуса

MCUCR – (MCU Control Register) регистр управления

TIMSK – (Timer/Counter Interrupt Mask Register) регистр масок прерывания по таймерам/счетчикам

ISP – (In System Programming) внутрисхемное программирования

SPI – (Serial Peripheral Interface) последовательный периферийный интерфейс

UART – (Universal asynchronous receiver/transmitter) универсальный асинхронный приёмопередатчик

Введение

В данной работе на основании учебного плана кафедры ИУ6 производится разработка устройства управления для приборов жилого помещения, который осуществляет включение и выключение устройств по расписанию, получаемом от удаленного сервера по протоколу асинхронной передачи UART.

Для выполнения поставленной задачи используется высокопроизводительный 8-разрядный контроллер AVR ATmega8515. Внутренняя оперативная память SRAM данного микроконтроллера позволяет хранить до 512 байт данных, чего вполне достаточно для хранения расписания для многократного включения и отключения 8 приборов жилого помещения в течение суток. Модуль USART, 1 8-разрядный и 1 16-разрядный таймеры позволяют обеспечить необходимый функционал устройству для оперативного получения расписания и включения приборов в необходимые временные отрезки.

Для нештатных ситуаций, которые могут возникать в процессе работы устройства и сервера расписаний, в устройстве присутствует пуль управления оператором [ПУО]. ПУО позволяет манипулировать устройством управления напрямую.

Основная часть

В данной курсовой работе было разработано устройство управления 8 приборами жилого помещения на основе 8-разрядного высокопроизводительного микроконтроллера AVR ATmega8515

В техническом задании не предъявлялись специальные требования к выбору микроконтроллера и периферийных микросхем для создаваемого устройства управления. Был выбран контроллер ATmega8515, ввиду его функциональности и высокой частоты работы процессора.

Для решения задачи получения расписания с удаленного сервера было принято решения использовать протокол передачи данных RS-232 и модуль микроконтроллера USART. При этом было принято решения использовать асинхронный способ передачи данных по UART ввиду простоты и большей эффективности такого метода по сравнению с синхронной передачей.

Для хранения расписания было принято решения использовать 4-х байтовые сообщение в оперативной памяти SRAM в качестве меток включения или выключения устройств. В эти 4 байта входят номер устройства, время (часы, минуты, секунды), когда необходимо выключить/включить устройство, и флаг, характеризующий выключение или включение устройства.

Для ручного управления устройством был принято решения предусмотреть пульт оператора, с помощью которого можно включать и выключать устройства без расписания. В качестве ПУО используется матричная клавиатура 4x4 с 16 клавишами, 8 из которых – различные команды, ещё 8 – клавиши выбора прибора.

1 Конструкторская часть

1.1 Проектирование МК-системы

1.1.1 Описание структурно-функциональной схемы микроконтроллерной системы

Согласно заданию, нужно разработать устройство управления 8 приборами жилого помещения согласно расписанию, получаемому с удаленного сервера.

Из этого следует, что необходимо использовать модуль для приема и передачи данных USART микроконтроллера.

Для отсчета времени следует использовать имеющиеся таймеры T0 8-разрядный и T1 16-разрядный.

Для обеспечения повышенной точности при работе разрабатываемой системы, будет подключен внешний кварцевый генератор с частотой 8 MHz к разъемам XTAL1 и XTAL2.

Для дополнительного контроля над устройством будем возможность управлять им напрямую с помощью пульта оператора, состоящего из 16 кнопок.

Для подключения к ПЭВМ, которая и является сервером расписания, будем использоваться драйвер MAX232.

Для возможности оперативного получения расписания необходимо предусмотреть возможность отправки запроса на сервер расписания для получения последних данных о текущем расписании и времени.

Для отображения текущего времени в состав разрабатываемой МК-системы будет использоваться LCD-дисплей LM016L 16x2, где каждую секунду будет обновляться текущее время микроконтроллера.

Т. к. предполагается, что приборы будут питаться от стандартного напряжения в жилом помещении (220 В), то для включения и отключения питания устройств будет использоваться блок реле.

Итоговое устройство должно выводить на порт управления приборами текущее состояние каждого из приборов, где каждому прибору соответствует один бит, согласно принятому от ПВМ и записанному в оперативную память расписанию работы приборов.

Исходя из вышеперечисленного, итоговое устройство должно состоять из следующих блоков:

- 1) микроконтроллер;
- 2) блок обмена информацией с ПЭВМ;
- 3) пуль оператора;
- 4) блок реле для управления питанием приборов
- 5) LCD-дисплей

Обобщенная структура проектируемого устройства представлена на рисунке 1.

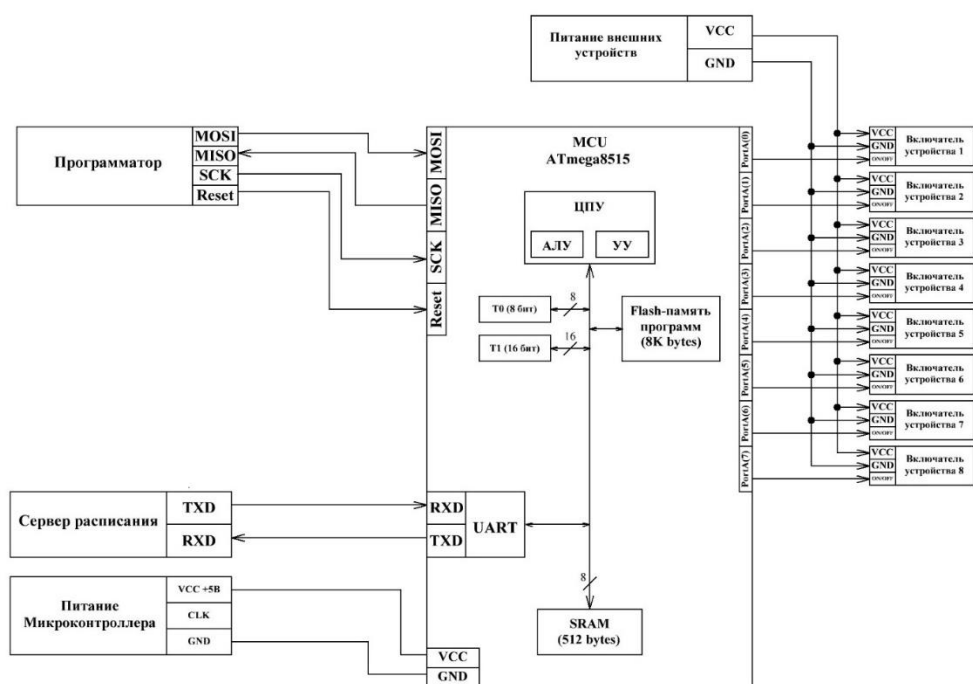


Рисунок 1 – Структурная схема устройства управления приборами жилого помещения

1.1.2 Выбор микроконтроллера

При выборе микроконтроллера важными параметрами были выбраны следующие:

- наличие модуля для асинхронной передачи данных UART;
- частота работы;
- объем оперативной памяти;
- количество выводов
- объем памяти программ
- количество таймеров и их разрядность

В таблице представлено сравнение некоторых микроконтроллеров AVR по важными для данной разрабатываемой системы параметрам.

Таблица 1 – Сводная таблица параметров различных МК AVR

МК	Пины	ПЗУ КБ	SRAM Б	Таймеры	Максим альная частота	Наличие модуля USART
ATmega8A	28	8	1024	2x8 бит 1x16 бит	16	Да
AT90LS2323	8	2	128	1x8 бит	4	Нет
AT90S4433	28	4	128	6x10 бит	8	Да
AT90S2343	8	2	128	1x8 бит	10	Нет
ATmega8515	40	8	512	1x8 бит 1x16 бит	16	Да
ATtiny2313	20	2	128	1x8 бит 1x16 бит	20	Да

Исходя из сводной таблицы видно сразу, что для поставленных целей подходят не все из представленных контроллеров.

Однозначно не подходят микроконтроллеры, у которых отсутствует USART, без которого осуществление асинхронного обмена данным с удаленным сервером не представляется возможным.

Микроконтроллер ATtiny2313 подходит по большинству параметров, однако он обладает достаточно небольшим объемом SRAM и малым количеством пинов, что приведет к наложению значительных ограничений при реализации МК-системы с использованием этого микроконтроллера.

Наиболее подходящие кандидаты – это ATmega8A и ATmega8515. В данном случае предпочтение отдается ATmega8515, т.к. он обладает большим количеством выводов. ATmega8A обладает большим объемом SRAM и таймеров, но в контексте поставленной задачи такое количество памяти и счётчиком является избыточным и не будет использоваться в полном объеме.

1.1.3 Описание архитектуры и технических характеристик микроконтроллера

В проектируемом устройстве используется 8-битный микроконтроллер AVR ATmega8515. Его функциональная схема представлена на рисунке 2.

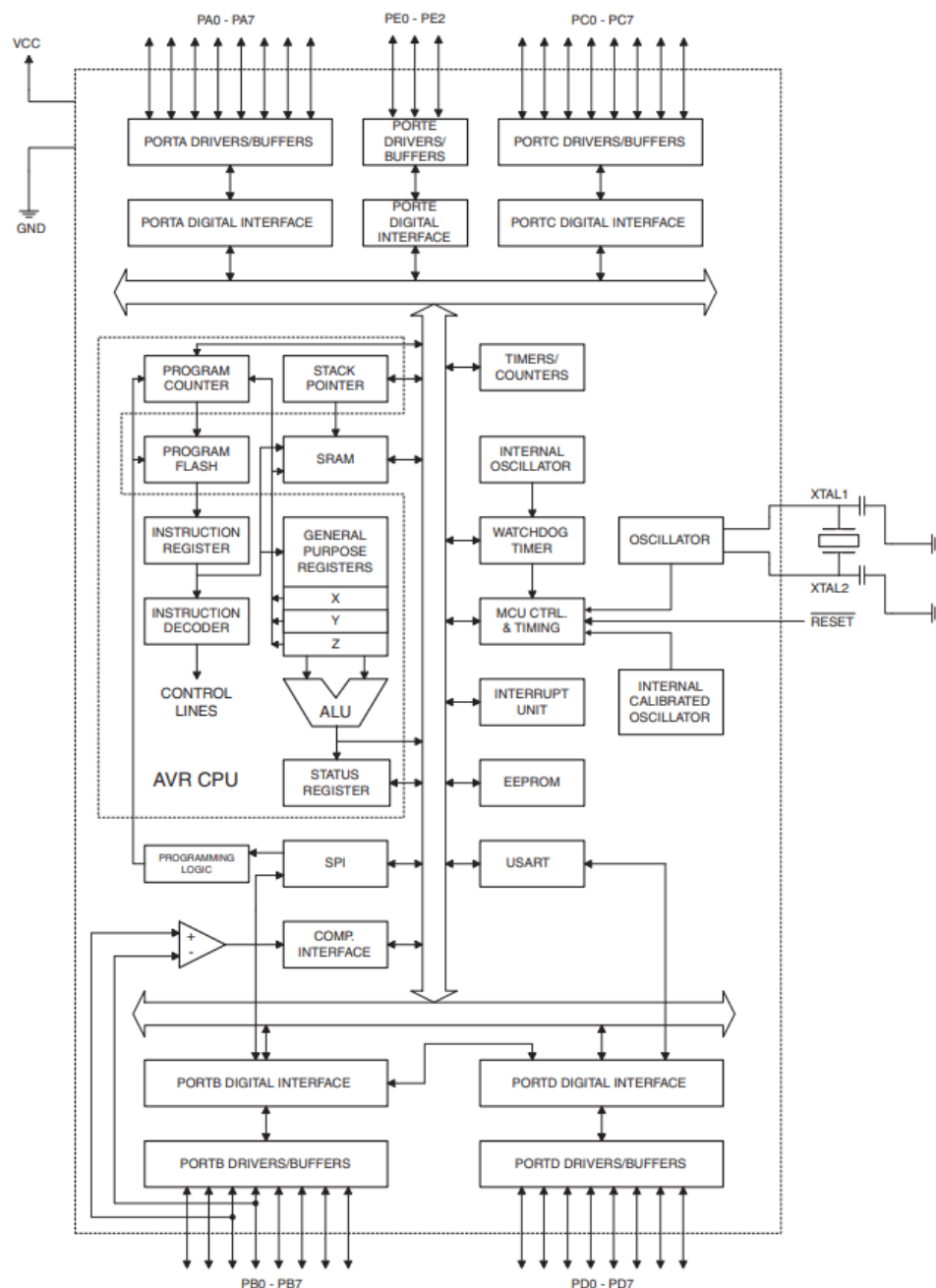


Рисунок 2 – Функциональная схема микроконтроллера ATmega8515

Из функциональной схемы видно, что микроконтроллер обладает четырьмя 8-разрядными портами ввода-вывода, один из которых имеет АЦП с мультиплексором; дополнительный 3-разрядный порт PE; аппаратными интерфейсами USART, SPI; встроенным компаратором, встроенным генератором (осциллятором); счетчиками (один 8-разрядный и один 16-разрядный); сторожевым таймером; блоком прерываний; энергонезависимой и энергозависимой памятью.

Семейство микроконтроллеров Mega – это 8-битные микроконтроллеры, представляющие собой одну из лучших основ для создания экономных и высокопроизводительных устройств различного назначения.

Микроконтроллеры этого семейства изготавливаются по RISC-архитектуре, согласно которой, инструкции, выполняемые процессором микроконтроллера должны быть как можно более простыми. Такой подход позволяет получить оптимальное соотношение между стоимостью, быстродействием и энергопотреблением.

1.1.4 Распределение адресного пространства ATmega8515

В микроконтроллерах AVR используется Гарвардская архитектура. Согласно этой архитектуре память программ и память данных находится в разных адресных пространствах. Способу адресации и доступа к этим областям также различны. Такая архитектура обеспечивает центральному процессору работать одновременно с памятью программ и с памятью данных. Это существенно повышает производительность МК.

Память данных МК разделена на три части:

- регистровая память;
- оперативная память;
- энергонезависимая память.

Регистровая и оперативная память находится в одном адресном пространстве, в отличие от энергонезависимой, которая обладает собственным адресным пространством.

Изображение адресных пространств МК ATmega8515 представлено на рисунке .

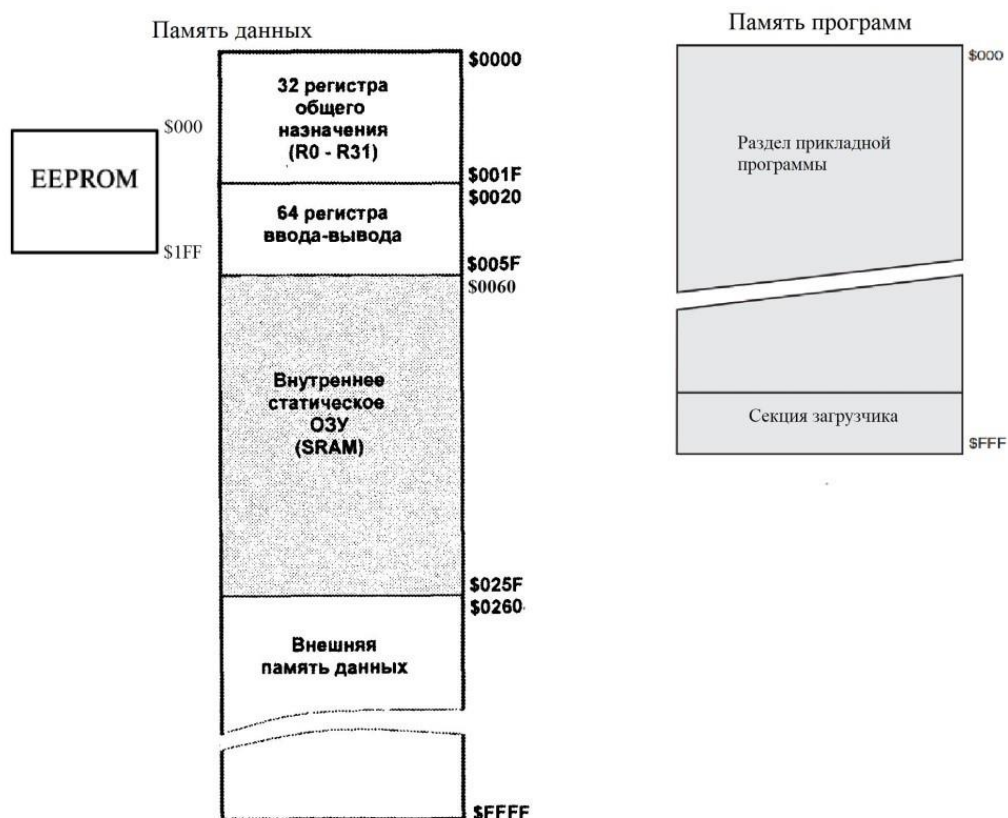


Рисунок 3 – Адресные пространства МК ATmega8515

Регистровая память включает 32 регистра общего назначения и 64 регистра ввода-вывода.

Для хранения данных имеется память RAM объемом 512 байт. Есть возможность подключение внешней памяти SRAM, позволяющее расширить оперативную память микроконтроллера до 64 Кбайт.

Для долгосрочного хранения данных в МК присутствует 512 байт памяти EEPROM.

1.1.5 Особенности системы команд микроконтроллера ATmega8515

Система команд микроконтроллера ATmega8515 выполнена по RISC архитектуре и состоит из 130 инструкций, большинство из которых выполняется за один такт.

Система команд обладает полностью статическим функционированием. Производительность составляет до 16 млн. операций в секунду при тактовой частоте 16 МГц.

1.2 Разработка функциональной схемы

1.2.1 Используемые модули ATmega8515

Микроконтроллер ATmega8515 является основным узлом в данной разрабатываемой системе.

В результате разработки МК-системы используются следующие компоненты и модули микроконтроллера;

- модуль USART в асинхронном режиме;
- модуль SPI для программирования микроконтроллера;
- таймер T0 для отсчета времени
- система прерываний
- 8 выводов порта A для управления приборами
- 8 выводов порта C для функционирования ПУО
- 3 вывода порта B для программирования
- 2 вывода порта D для передачи данных по USART

Для эффективной и быстрой передачи данных, а также для оперативного включения и отключения приборов используется тактовая частота в 8 МГц.

1.2.2 Пульт оператора

Пульт оператора представляет из себя матричную клавиатуру, которая состоит из 16 кнопок.

Он позволяет оказывать воздействие на работу устройства в случае непредвиденных ситуаций (отказ работы сервера расписания, необходимость срочного включения или отключения одного из приборов и т. п.).

Все 16 клавиш пульт оператора задействованы и несут в себе определенную функцию. Функционирование части клавиш зависит от клавиш, которые были нажаты на пульте оператора в предыдущий момент времени. Функциональное назначение кнопок пульта оператора представлено на рисунке 4.

7	8	ВСЕ	ВКЛП
4	5	6	ВЫКЛП
1	2	3	ЗСР
СВПР /ОТМ	УРПУ	УВПУ	ЗСВ

Рисунок 4 – Пульт управления оператора

Клавиши на пульте оператора можно условно разделить на 2 категории: функциональные клавиши, выполняющие какую-то операцию (ВКЛП, ЗСР, СВПР и т.д.), и контекстные клавиши, предназначенные для выбора прибора, над которым необходимо совершить, выбранное с помощью функциональных клавиш, действие (1, 2, ..., 7, ВСЕ).

Подробное описание функции, которую несет каждая из клавиш, представлено в таблице .

Таблица 2 – Назначения и расшифровка клавиш пульта оператора

Название клавиши	Группа	Расшифровка и назначение
ВКЛП	Функциональная	Включить принудительно. Принудительно включает одно из выбранных устройств и переводит его в принудительный режим.
ВЫКЛП	Функциональная	Выключить принудительно. Принудительно выключает одно из выбранных устройств и переводит его в принудительный режим

ЗСР	Функциональная	Запросить серверное расписание. Запрашивает у сервера новое расписание.
ЗСВ	Функциональная	Запросить серверное время. Запрашивает у сервера текущее время.
УРПУ	Функциональная	Установить расписание по умолчанию. Устанавливает расписание по умолчанию.
УВПУ	Функциональная	Установить время по умолчанию. Устанавливает время по умолчанию.
СВПР/ОТМ	Функциональная	Сброс всех принудительных режимов/Отмена. Выводит все устройства из принудительного режима. Может использоваться как клавиша отмены операции на этапе выбора прибора.
1	Контекстная	Выбирает первый прибор
2	Контекстная	Выбирает второй прибор
3	Контекстная	Выбирает третий прибор
4	Контекстная	Выбирает четвертый прибор
5	Контекстная	Выбирает пятый прибор
6	Контекстная	Выбирает шестой прибор
7	Контекстная	Выбирает седьмой прибор
8	Контекстная	Выбирает восьмой прибор
ВСЕ	Контекстная	Выбирает все приборы

Один из возможных сценариев использования пульта оператора:

- 1) возникла необходимость получения более актуального расписания;
- 2) оператор нажимает кнопку «ЗСР»;
- 3) ответ от сервера расписания не приходит. Оператор решает самостоятельно выключить один из приборов;

- 4) оператор нажимает клавишу «ВКЛП»;
- 5) оператор выбирает первое устройство нажатием клавиши «1»;
- 6) устройство выключается;
- 7) оператор снова запрашивает расписание клавишей «ЗСР»;
- 8) от сервера приходит новое расписание;
- 9) оператор нажимает клавишу «СВПР» для вывода приборов из принудительного режима;
- 10) приборы продолжают работать по новому расписанию.

На этапе 5 можно было осуществить нажатие на клавишу «ОТМ», что отменило бы действие клавиши «ВКЛП»

1.2.3 Блок передачи данных к ПЭВМ

Блок передачи данных состоит из драйвера MAX232 и COM-порта, соединяющего само устройство и удаленный сервер.

Сигнал TxD с выхода микроконтроллера поступает на схему формирования уровней сигналов интерфейса RS-232 (MAX232), далее через разъем, в усиленном состоянии, по уходит на линию связи.

Сигнал RxD, поступающий от сервера расписания, проходят через преобразователь, ослабляется, и попадает на вход микроконтроллера RxD

Усиление и ослабление сигнала необходимо, т. к. передача данных по кабелю требует большого уровня сигнала из-за затухания. Без усиления посылаемого сигнала он может в процессе достижения своей цели настолько ослабнуть, что ПЭВМ его не сможет воспринять. В случае с сигналами, которые поступают на микроконтроллер, их необходимо ослаблять во избежание сгорания микроконтроллера.

Модуль передачи данных USART настроен в данной разрабатываемой МК-системе следующим образом;

- скорость передачи данных 9600 бит в секунду;
- 8 бит данных в кадре;
- 1 стоповый бит;

– бит четности отключен.

Кадр UART изображен на рисунке 5.

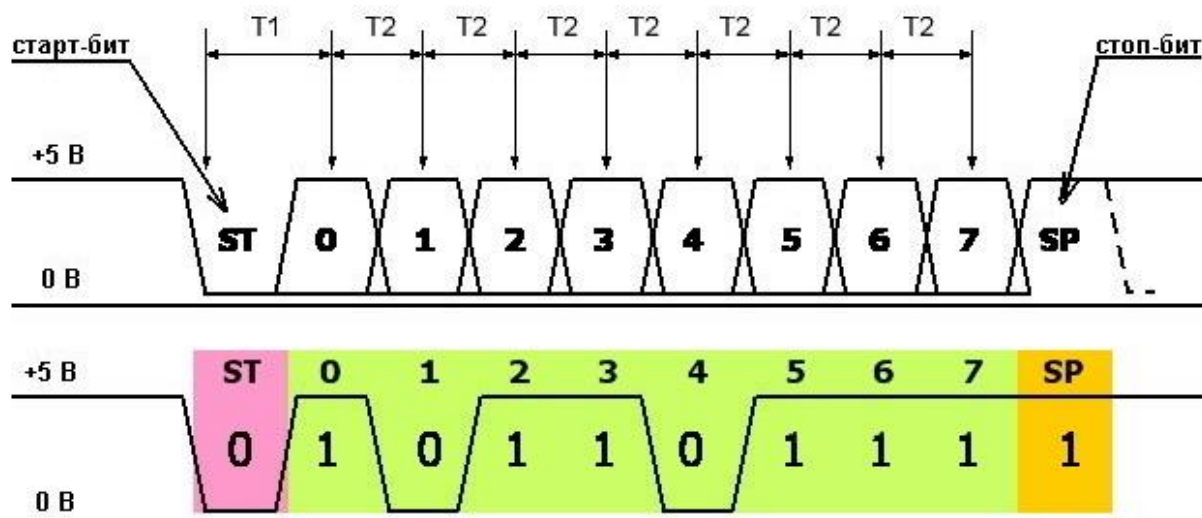


Рисунок 5 – Кадр UART

Отключение бита четности и отключение второго стопового бита обусловлено устремлением к большей скорости передачи данных.

Модуль USART ATmega8515 может принимать в одном кадре до 9 информационных бит, однако в данном случае, для упрощения алгоритмов обработки данных и более наглядного вида передаваемых данных было принято решение использовать 8 бит – размер байта памяти данных.

Получившаяся итоговая конфигурация является достаточно простой для понимания и отладки и одновременно высокопроизводительной.

1.2.4 Блок реле

Для управление блоком реле используется порт А. Каждый вывод порта А подключён к соответствующему реле для управления питанием прибора. При этом уровень логической единицы означает, что прибор в данный момент находится во включенном состоянии, а уровень логического нуля, соответственно, означает, что прибор в данный момент времени находится в выключенном состоянии.

Для вывода состояния приборов на порт А внутри программы микроконтроллера используется алгоритм, который по записанному в память SRAM расписанию определяет

1.2.5 LCD-дисплей

LCD дисплей представляет из себя жидкокристаллический индикатор, сделанный на основе жидких кристаллов. С его помощью, в простых устройства, можно отображаться простые графические объекты (буквы, цифры, специальные символы и т.д.).

В данной курсовой работе для разработки МК-системы используется LCD-дисплей LM016L с монохромным экраном, сделанный на базе контроллера HD44780. В данном дисплее используется монохромный экран с двумя строками вывода.

Видимая область экрана составляет по 16 символов для каждой строки, а общий объем памяти 40 символов для каждой строки. В данной курсовой работе для вывода текущего времени микроконтроллера достаточно одной видимой области экрана, потому сдвиг зоны отображения информации на экране не производится.

Схема отображения символов из ячеек памяти отображена на рисунке 6. Ячейки пронумерованы в 16-ричной системе счисления.

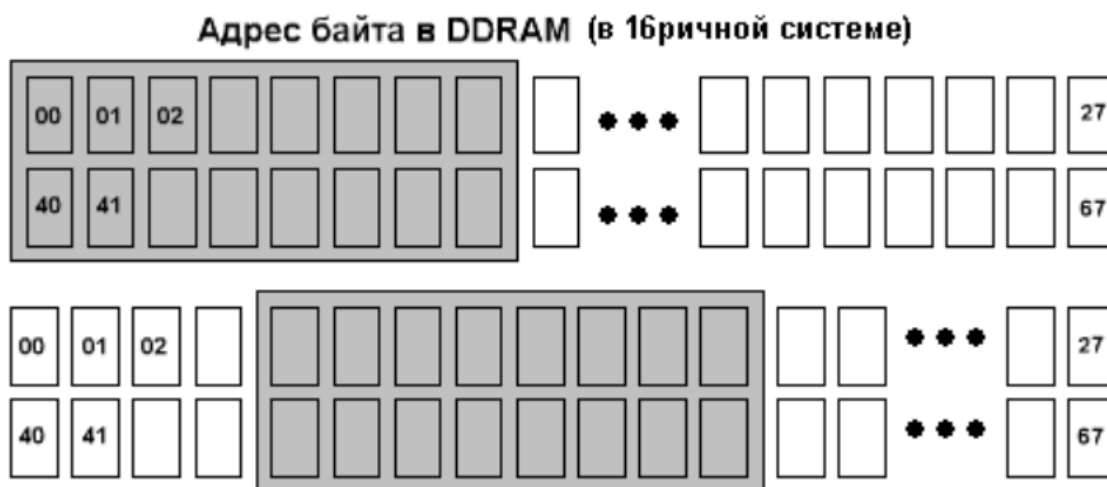


Рисунок 6 – Окно отображения дисплея LM016L

На представленном выше рисунке видно, что окно, сдвигаемой на некоторое количество ячеек в сторону, позволяет отобразить информацию, которая скрыта за областью отображения. Это позволяет хранить информацию в дисплее и отображать ее в случае необходимости.

Данный дисплей обладает 8 информационными и 3 управляющими входами. В зависимости от сигналов управления, последовательность бит, пришедшая на информационные входы воспринимается как очередной символ или одна из команд.

Перечень управляющих сигналов:

- **E** — стробирующий вход. Отрицательным перепадом напряжения на этой линии мы даем понять дисплею что нужно забирать/отдавать данные с/на шину данных;

- **RW** — определяет в каком направлении у нас движутся данные. Если 1 — то на чтение из дисплея, если 0 то на запись в дисплей;

- **RS** — определяет что у нас передается, команда ($RS=0$) или данные ($RS=1$). Данные будут записаны в память по текущему адресу, а команда исполнена контроллером.

Допустимый набор команды для дисплея LM016L представлен в таблице 3.

Таблица 3 – Команды управления дисплеем LM016L

DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Значение
0	0	0	0	0	0	0	1	Очистка экрана. Счетчик адреса на 0 позицию DDRAM
0	0	0	0	0	0	1	—	Адресация на DDRAM сброс сдвигов, Счетчик адреса на 0
0	0	0	0	0	1	I/D	S	Настройка сдвига экрана и курсора
0	0	0	0	1	D	C	B	Настройка режима отображения
0	0	0	1	S/C	R/L	—	—	Сдвиг курсора или экрана, в зависимости от битов
0	0	1	DL	N	F	—	—	Выбор числа линий, ширины шины и размера символа
0	1	AG	AG	AG	AG	AG	AG	Переключить адресацию на SGRAM и задать адрес в SGRAM
1	AD	AD	AD	AD	AD	AD	AD	Переключить адресацию на DDRAM и задать адрес в DDRAM

Значения каждого бита следующие:

1) I/D — инкремент или декремент счетчика адреса. По дефолту стоит 0 — Декремент. Т.е. каждый следующий байт будет записан в n-1 ячейку. Если поставить 1 — будет Инкремент;

2) S — сдвиг экрана, если поставить 1 то с каждым новым символом будет сдвигаться окно экрана, пока не достигнет конца DDRAM;

3) D — включить дисплей. Если поставить туда 0 то изображение исчезнет, а если 1 – изображение наоборот появится;

4) C — включить курсор в виде прочерка. Для включения курсора необходимо, чтобы бит был равен 1;

5) B — сделать курсор в виде мигающего черного квадрата;

6) S/C сдвиг курсора или экрана. Если стоит 0, то сдвигается курсор. Если 1, то экран. По одному разу за команду;

7) R/L — определяет направление сдвига курсора и экрана. 0 — влево, 1 — вправо;

8) D/L — бит определяющий ширину шины данных. 1-8 бит, 0-4 бита;

- 9) N — число строк. 0 — одна строка, 1 — две строки;
- 10) F — размер символа 0 — 5x8 точек. 1 — 5x10 точек;
- 11) AG — адрес в памяти CGRAM;
- 12) AD — адрес в памяти DDRAM.

Совокупность посланных информационных сигналов воспринимается как команды в случае, если управляющий вход RS=0.

1.3 Разработка принципиальной схемы

1.3.1 Синтез принципиальной схемы

По результату проектирования МК-системы и разработке функциональной схемы, а также анализу возможностей и требований к реализуемому устройству, был сформулирован перечень необходимых компонентов, а также способы и их подключения. На основе выделенных компонентов была разработана принципиальная схема.

В результате принципиальная схема может быть представлена следующими основными компонентами:

- 1) Микроконтроллер ATmega8515;
- 2) Блок реле для управления питанием приборов;
- 3) Матричная клавиатура 4x4;
- 4) LCD-дисплей для индикации текущего времени МК-системы;
- 5) Преобразователь входного напряжения.

1.3.2 Конфигурация выводов микроконтроллера

Основным узлом разрабатываемого устройства в данной курсовой работе является микроконтроллер ATmega8515. Конфигурация выводов микроконтроллера в корпусе PDIP приведена на рисунке 7.

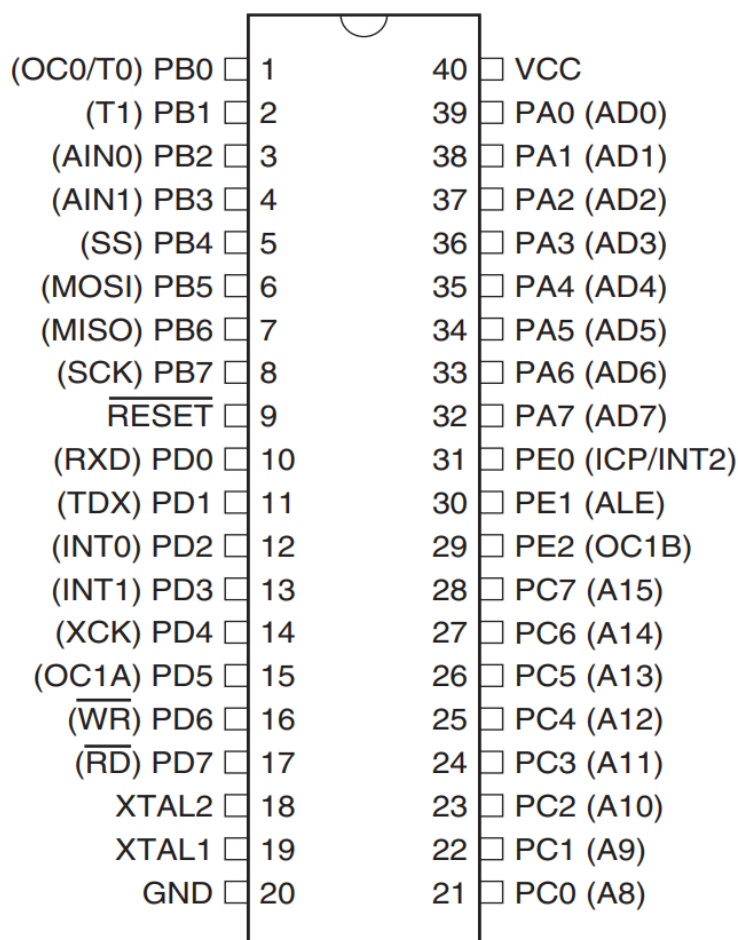


Рисунок 7 – Микроконтроллер ATmega8515 в корпусе PDIP

1.3.3 Схема понижения входного напряжения до 5В

На схему устройства подается напряжение 12В. Сам микроконтроллер и другие устройства принципиальной схема работают от напряжения питания 5В. Для преобразования 12В в 5В требуется использовать устройство, понижающее напряжение до необходимого уровня.

Для решения данной задачи используется стабилизатор напряжения КР142ЕН5А. Для нормальной работы данный стабилизатор шунтируется конденсаторами на входе и выходе. Схема подключения представлена на рисунке 8.

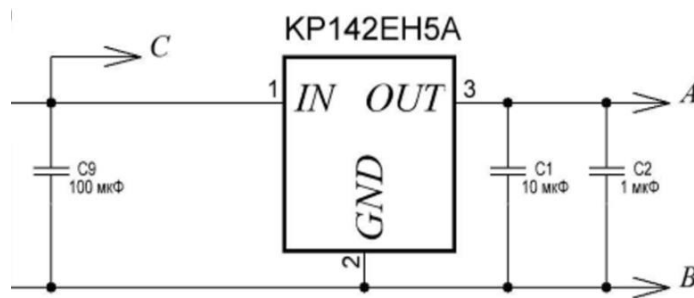


Рисунок 8 – Схема подключения стабилизатора напряжения KP142EH5A

В данной выше схеме подключения стабилизатора напряжения конденсатор C9 используется для сглаживания скачков напряжения питания на входе, а конденсаторы C1 и C2 используются в качестве фильтров, сглаживая подаваемое напряжения на устройства.

1.3.4 Подключение матричной клавиатуры

Матричная клавиатура представляет из себя блок кнопок, в котором клавиши размещены в виде матрицы на пересечении горизонтальных и вертикальных линий связи.

В данном разрабатываемом устройстве вертикальные линии подключены к входному регистру, а горизонтальный ряд к выходному регистру. На входной регистр подается код, который содержит 0 уровень сигнала на одном из разрядов и 1 на всех остальных. При замыкании кнопки вертикального ряда, на котором в данный момент присутствует сигнал 0, этот сигнал поступает на горизонтальную линию и по ней на выходной регистр. Проверяя состояние выходного регистра, микроконтроллер может идентифицировать строку и номер замкнутой кнопки.

С помощью последовательного опроса кодами вида: 1110, 1101, 1011 и 0111 – можно опросить состояния всех столбцов клавиатуры и установить номер замкнуто кнопки.

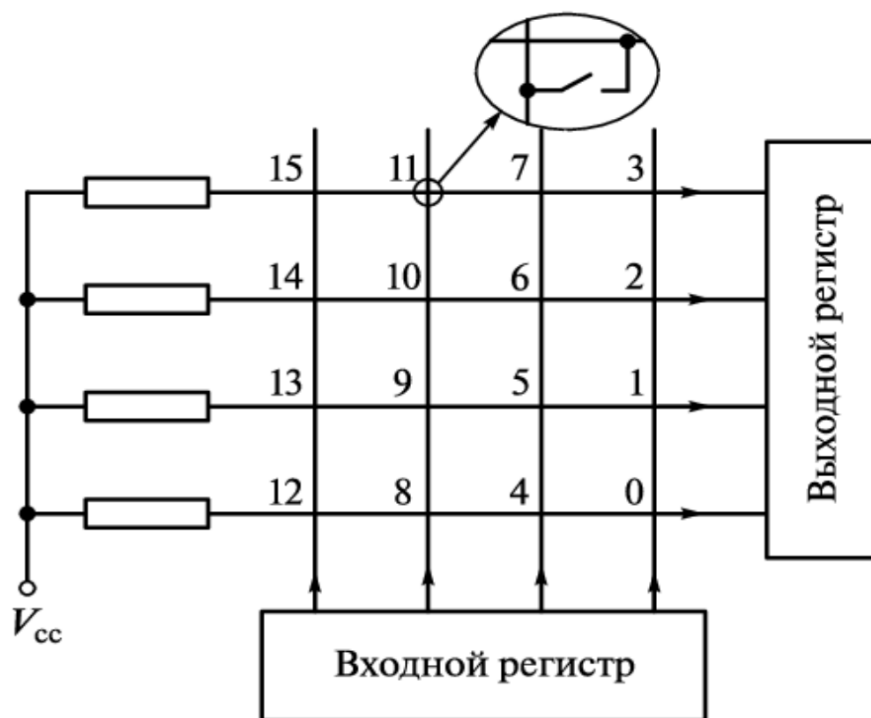


Рисунок 9 – Матричная клавиатура

Входной регистр данной матричной клавиатуры подключается к входам 4, 5, 6, 7 порта С.

Выходной регистр подключается к входам 0, 1, 2, 3 того же порта С микроконтроллера АТmega8515.

Таким образом, все 16 кнопок обрабатываются с помощью 8 пинов порта С микроконтроллера. В случае, если бы каждая кнопка подключалась простейшим образом (1 кнопка – 1 вывод порта), то пришлось бы использовать в 2 раза больше выводов, что могло привести к дефициту выводов микроконтроллера.

Подключение клавиатуры матричным способом позволят довольно сильно экономить на количестве необходимых выводов для считывания состояний кнопок. При этом чем больше размерность матричной клавиатуры – тем более эффективный такой способ подключения.

При этом стоит отметить, что минимальный размер матричной клавиатуры, в котором есть смысл – это 3x2 или 2x3.

1.3.5 Подключение LCD-дисплея

LCD-дисплей LM016L может работать в 2-х режимах:

- 1) 8-битный режим. В этом режиме данные передаются сразу одним байтом за один такт. При этом используются сразу 8 контактов;
- 2) 4-битный режим. В этом режиме данные передаются полубайтами, используя только 4 контакта. Для передачи байта в данном режиме необходимо передавать байта данных в 2 такта.

В целях экономии выводов микроконтроллера в данной курсовой работе используется 4-битный режим.

Схема подключения этого дисплея показана на рисунке 10.

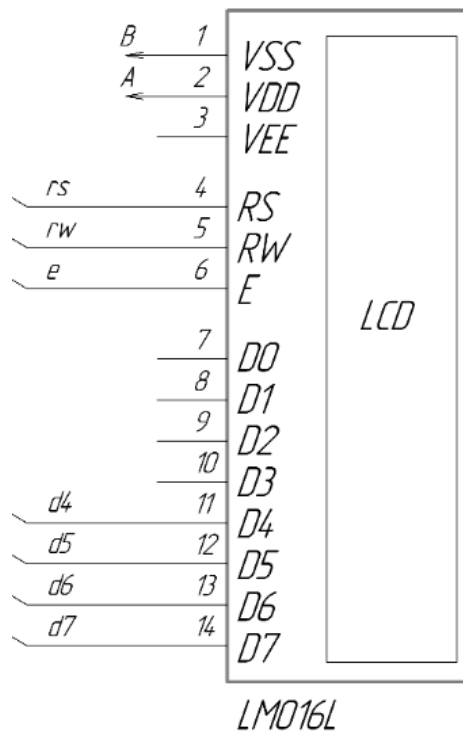


Рисунок 10 – Подключение LCD-дисплея LM016L

Как видно из рисунка выше, информационные входы D0 – D3 не используются, т.к. данный дисплей используется в 4-битном режиме. Вход VEE (контрастность) также не используется, поскольку нет необходимости динамически менять контрастность дисплея в процессе его работы.

1.4 Расчёт потребляемой мощности

Оценка мощности, потребляемой микроконтроллером, будет производиться при условии пикового режима его работы.

Для подсчета потребляемой мощности, спроектированной МК-системы, воспользуемся графиком потребляемого тока микроконтроллера ATmega8515 в зависимости от частоты и питающего напряжения. Сам график приведен на рисунке 11.

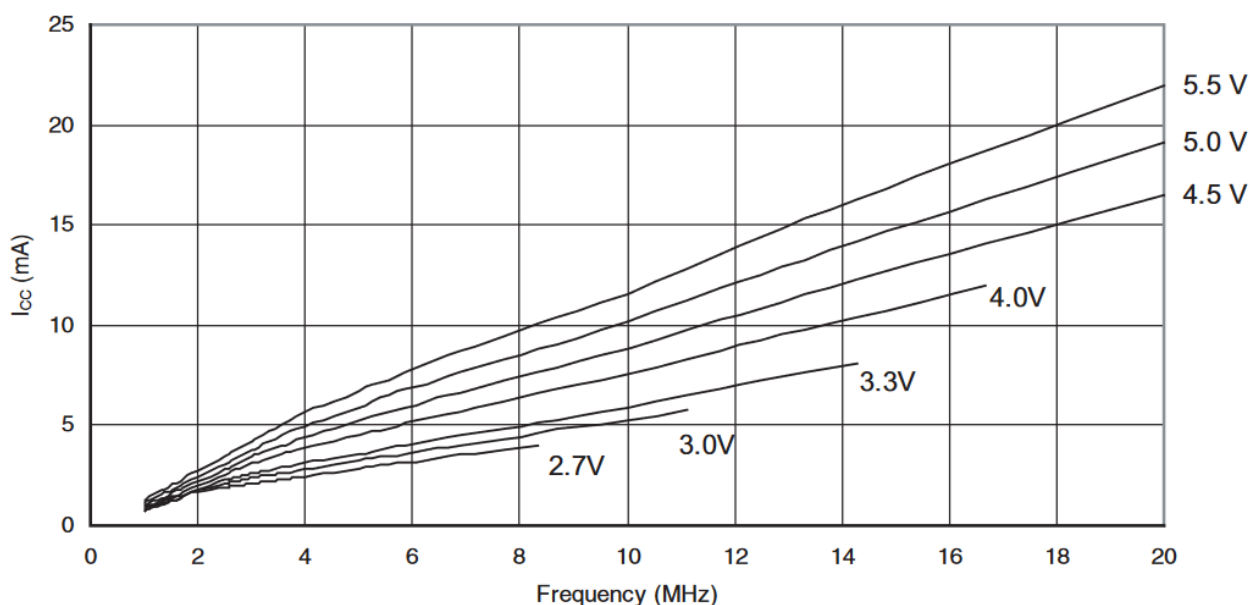


Рисунок 11 – График зависимости потребляемого тока для ATmega8515

По графику определяем ток $I_{cc}(8MHz, 5V) \approx 8$ мА.

Потребление LCD-дисплея LM016L при напряжении питания 5В составляет в пике 3 мА, согласно datasheet. Потребление дайвера MAX232 при передачи данных может составлять максимум 22 мА при скорости передачи в 9600 бод/с. Рассеиваемая мощность на стабилизаторе напряжения КР142ЕН5А равна падению напряжения на нем 7В, умноженному на суммарный ток, проходящий через него 33 мА.

Потребляемую мощность устройствами можно определить по следующей формуле: $P = I_{cc} * U_{пит}$.

Потребляемая мощность различных устройств представлена в таблице .

Таблица 4 – Потребляемая мощности компонентов МК-системы

Микросхема	P_{one} , мВт	Количество	P_{summ} , мВт
ATmega8515	40	1	40
KP142EH5A	231	1	231
MAX232	110	1	110
LM016L	15	1	15

Суммарная потребляемая мощность составляет около 396 мВт.

Основной потребитель – стабилизатор напряжения и драйвер MAX232. Однако вычислимая мощность является максимально возможной в штатном режиме работы интенсивность передачи данных по драйверу MAX232 будет на порядок меньше, чем максимально возможная.

1.5 Описание алгоритмов функционирования устройства

На рисунке 12 представлена обобщенная схема-алгоритмы работы разработанной программы для микроконтроллера ATmega8515. Данная схема дает общее представление о принципах работы разработанного устройства управления приборами жилого помещения и как оператор может взаимодействовать с разработанной МК-системой с помощью пульта оператора.

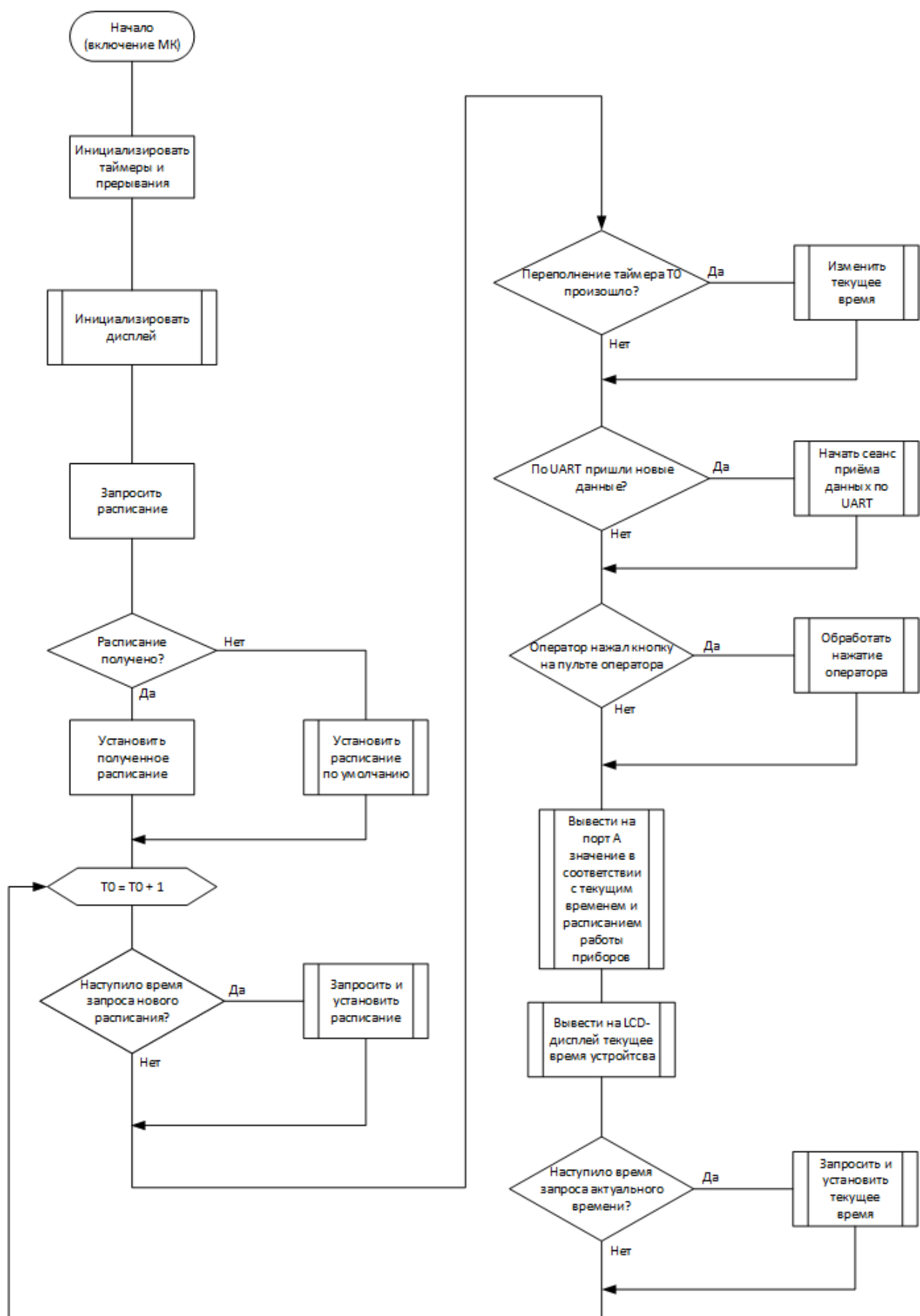


Рисунок 12 – Обобщенная схема-алгоритма работы программы

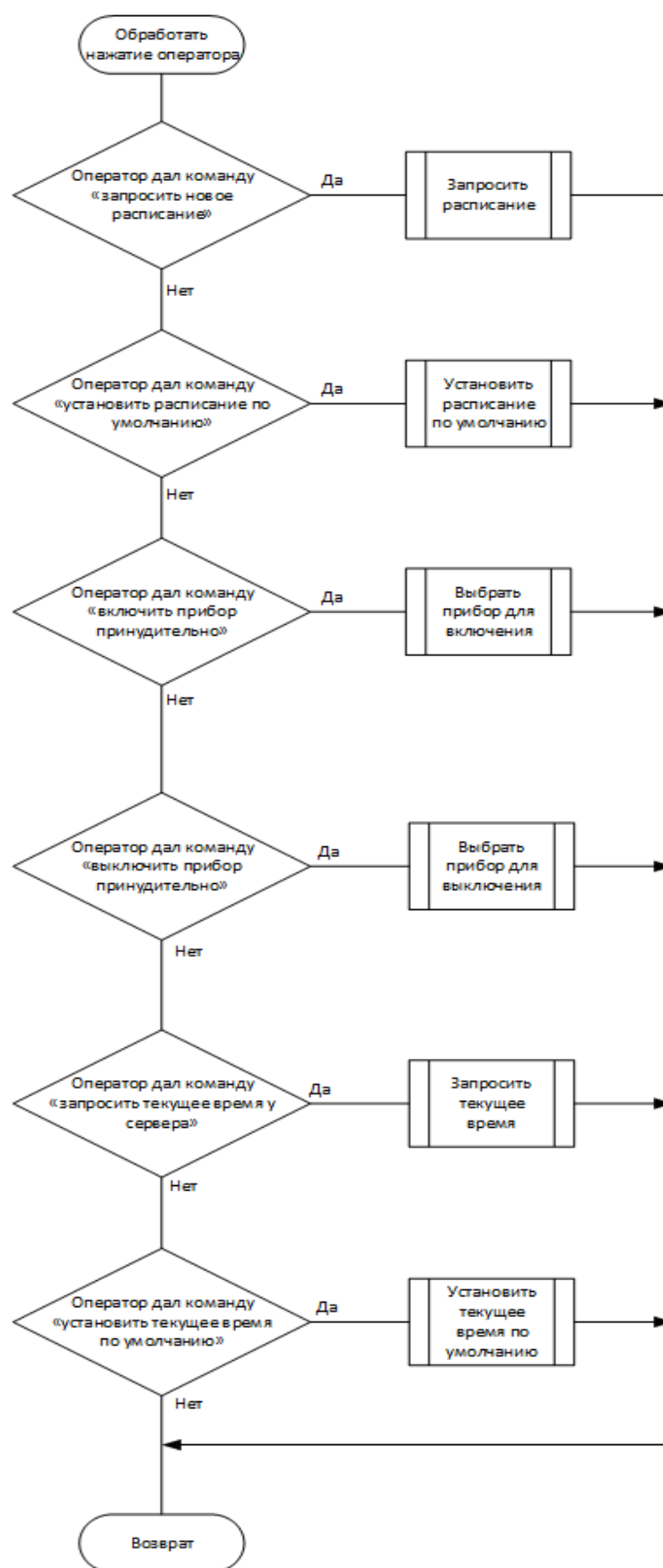


Рисунок 13 – Схема-алгоритм обработки нажатия клавиши на пульте оператора

Алгоритм обработки нажатия кнопки на пульте оператора представлена на рисунке 13. Здесь, в зависимости от нажатой кнопки, производится либо некоторая операция, либо ожидается следующее контекстное нажатие. Не предполагается нажатие сразу нескольких клавиш или их долгое удержание в качестве дополнительной функциональности.

Само срабатывание клавиши происходит в момент отпускания клавиши, а не сразу при её нажатии.

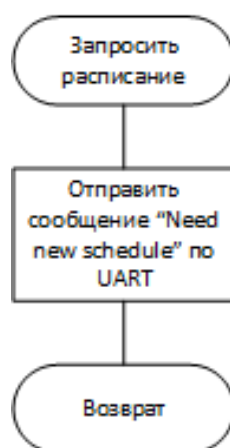


Рисунок 14 – Схема-алгоритм отправки запроса нового расписания

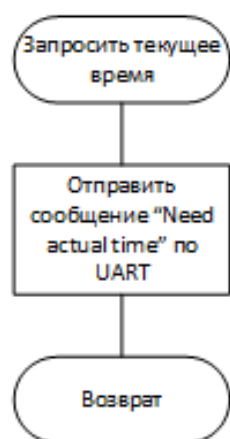


Рисунок 15 – Схема-алгоритм отправки запроса текущего времени

При запросе нового расписания или текущего времени программа не переходит в режим какого-либо ожидания приёма, а продолжает работу в штатном режиме. Сеанс приема расписания начинается в момент начала передачи ответа от сервера.

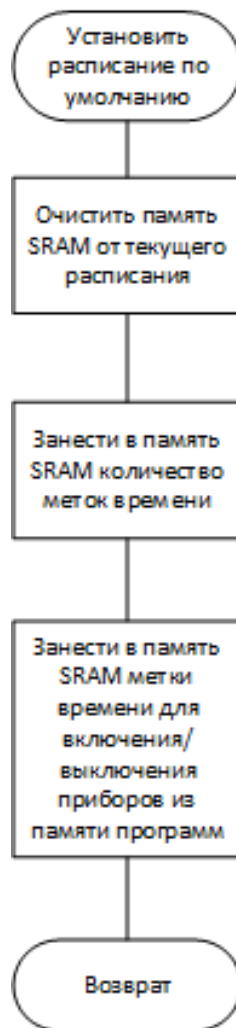


Рисунок 16 – Схема-алгоритм установки расписания по умолчанию

Установка расписания по умолчанию осуществляется посредством занесения вместо временных меток для включения и выключения приборов по расписанию, получаемому от удаленного сервера, временных меток из памяти программ, определенными заранее на этапе программирования микроконтроллера.

Установка времени по умолчанию осуществляется немного иным образом. Информация о текущем времени в секундах, минутах и часах находится в соответствующих регистрах `time_seconds`, `time_minutes` и `time_hours`. При невозможности получения времени от сервера в эти регистры можно установить расписание, зашитое заранее в память программ на этапе программирования микроконтроллера.



Рисунок 17 – Схема-алгоритм установки времени по умолчанию

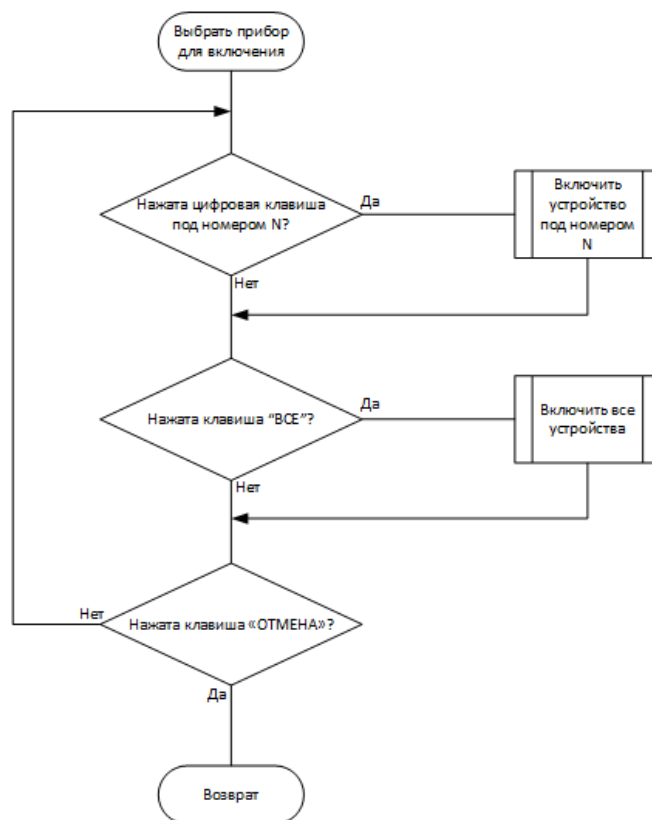


Рисунок 18 – Схема-алгоритм выбора прибора для включения

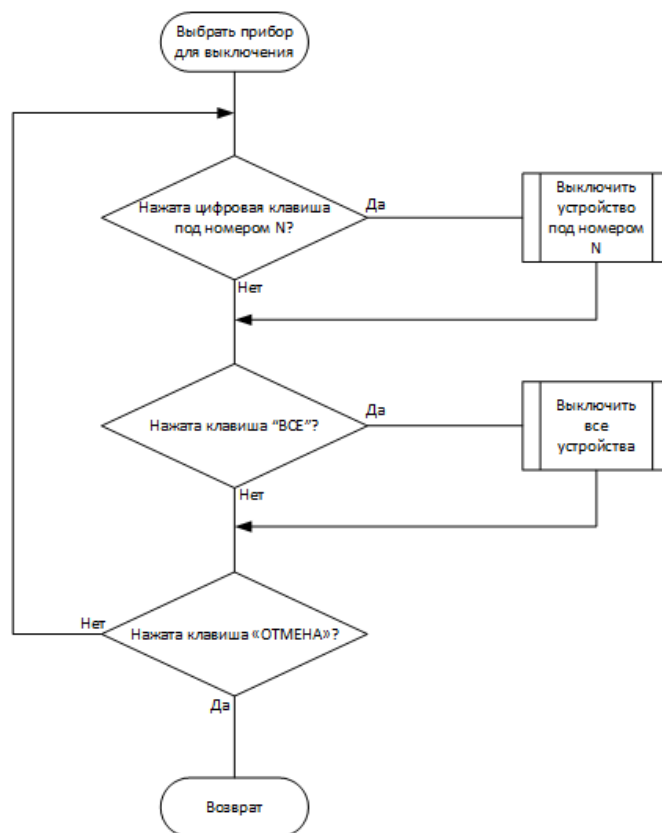


Рисунок 19 – Схема-алгоритм выбора прибора для выключения

2 Технологическая часть

2.1 Характеристика использованных систем разработки

Для проектирования и отладки разрабатываемой МК-системы в качестве средства разработки использованы следующие среды:

- 1) AVR Studio 4 – для отладки программного кода на ассемблере;
- 2) Proteus 8 Professional – для симуляции работы устройства.

AVR Studio 4 представляет из себя удобную и относительно простую среду для разработки программного обеспечения под микроконтроллеры фирмы AVR на языках C и Assembler.

ISIS Proteus позволяет строить и симулировать спроектированные схемы и прошивать микропроцессорные компоненты созданными программами в AVR Studio и запускать полученную виртуальную модель в режиме симуляции реального времени.

ISIS Proteus предоставляет взаимодействовать с элементами ввода-вывода, такими как светодиодные и LCD-дисплеи, а также с исполнительными механизмами (кнопки и различные другие переключатели).

Данный пакет программ специально предназначен для разработки программного обеспечения для микроконтроллеров AVR. Он дает возможность достаточно тщательно отлаживать разработанные программы, предоставляя визуальные инструменты и отображая состояния всех регистров используемого микроконтроллера.

2.2 Оценка количества задействованной памяти микроконтроллера ATmega8515

Среда AVR Studio позволяет определять процент используемой памяти микроконтроллера. Количественно задействованной памятью представлено на рисунке 20.

ATmega8515 memory use summary [bytes]:

Segment	Begin	End	Code	Data	Used	Size	Use%
[.cseg]	0x000000	0x00067e	1602	0	1602	8192	19.6%
[.dseg]	0x000060	0x000062	0	2	2	512	0.4%
[.eseg]	0x000000	0x000000	0	0	0	512	0.0%

● Assembly complete, 0 errors. 0 warnings

Рисунок 20 – Количество занимаемой памяти программой

2.3 Симуляция в Proteus 8

Отладка с помощью пакета программ ISIS Proteus дает возможность получить наглядный результат моделирования разработанной МК-системы.

Для симуляции работы МК и датчиков построена упрощенная схема в Proteus 8, представленная на рисунке 21. Эта схема состоит из матрицы кнопок, LCD-дисплея, драйвера MAX232, виртуального терминала и блока реле.

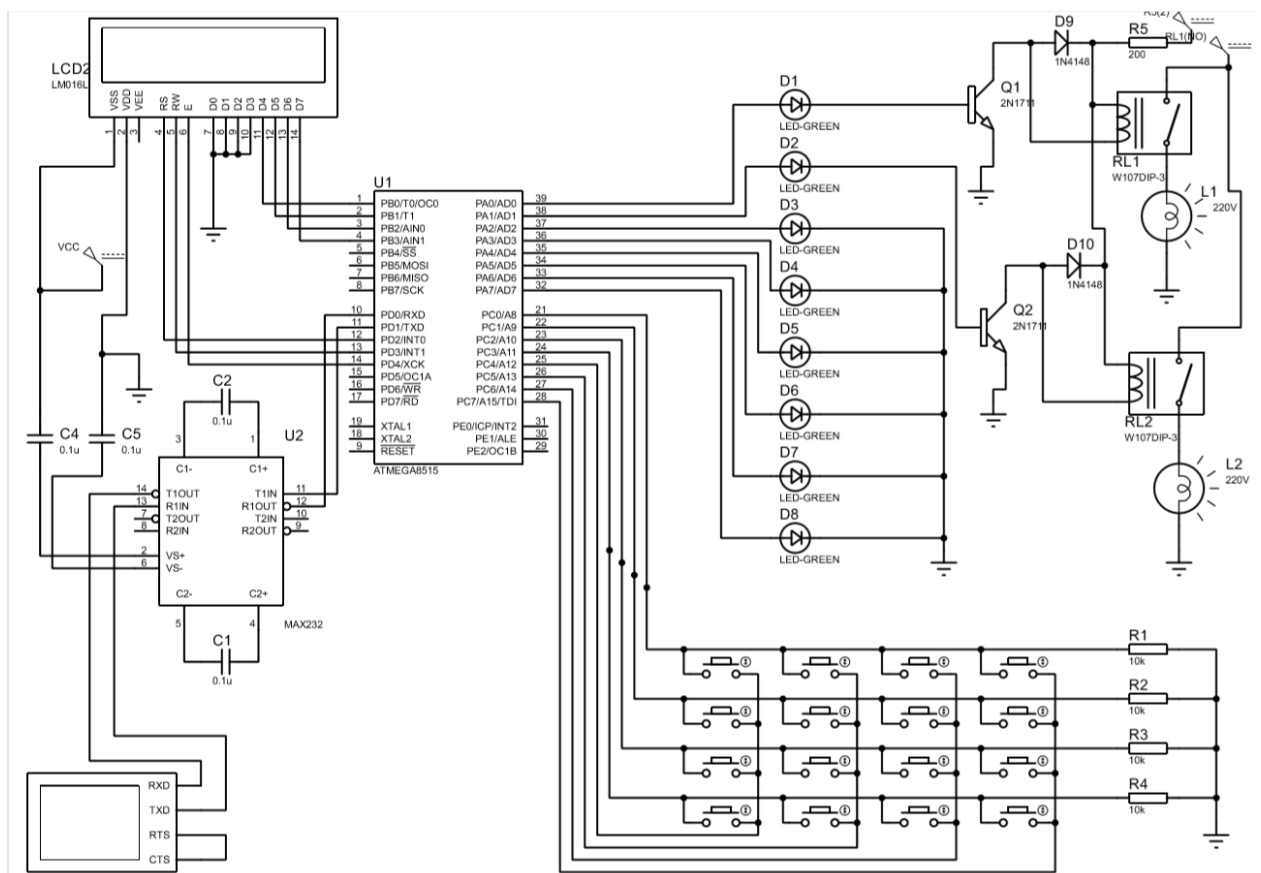


Рисунок 21 – Упрощенная схема разрабатываемой МК-системы

Для того, чтобы не перегружать схему однотипными компонентами, в блоке реле показаны только 2 компонента с реле, остальные заменены простыми светодиодами, олицетворяющие собой работу приборов.

На рисунке 22 представлен скриншот работающей модели.

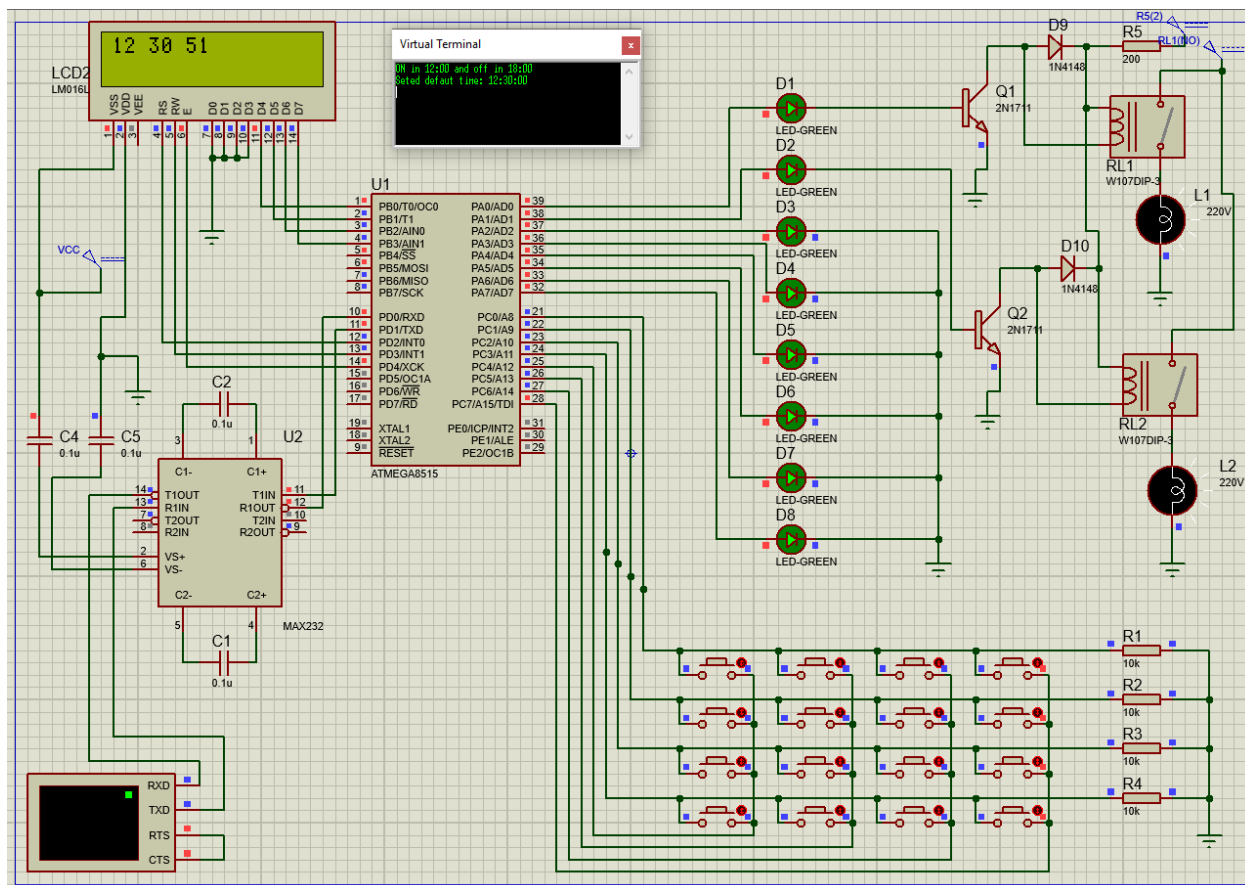


Рисунок 22 – Симуляция в Proteus

Матричная клавиатура представляет из себя систему из 16 кнопок, соединенных с портами порта С определенным образом. Она служит пультом оператора для взаимодействия пользователя с устройством напрямую, без необходимости передавать информацию через сервер расписания.

Драйвер MAX232 представляет собой связующий элемент, позволяющий микроконтроллеру принимать с удаленного сервера информационные сигналы значительного большего уровня напряжения, чем 5В. Кроме того, при отправке данных на удаленный сервер, данный драйвер наоборот усиливает сигнал, чтобы он был успешно доставлен и распознан ПЭВМ.

Виртуальный терминал позволяет симулировать общение между ПЭВМ и разработанной МК-системой с помощью модуля USART микроконтроллера ATmega8515.

Схема из 8 светящихся диодов показывает какие устройства включены или выключены посредством вывода их текущего состояния на порт А микроконтроллера. Данная схема позволяет понять во время отладки работы МК-системы какое из устройств в каком состоянии находится в результате действий оператора или в результате установки расписания с удаленного сервера расписаний.

2.4 Способы программирования памяти микроконтроллера ATmega8515

Для программирования МК ATmega8515 могут быть использованы два способа:

- программирование повышенным напряжением в параллельном формате с использованием дополнительного источника питания +12В;
- внутрисхемное программирование ISP с использованием последовательного периферийного интерфейса SPI.

Рассмотрим внутрисхемное программирование ISP по интерфейсу SPI.

Это самый популярный способ прошивать современные контроллеры. Внутрисхемным данный метод называется потому, что микроконтроллер в этот момент находится в схеме целевого устройства. Для нужд программатора в этом случае выделяется несколько выводов контроллера (обычно 3..5 в зависимости от контроллера). К этим выводам подключается прошивающий шнур программатора и происходит заливка прошивки. После чего шнур отключается, и контроллер начинает работу.

Для подключения по SPI, передачи прошивки по нему в МК AVR и работы программатора нужно четыре линии и питание (достаточно только земли, чтобы уравнивать потенциалы земель программатора и устройства):

1) SCK — тактовые импульсы интерфейса SPI, синхронизирующие все операции обмена данными.

2) MOSI (Master-Output/Slave-Input) — линия данных от ведущего устройства к контроллеру.

3) MISO (Master-Input/Slave-Output) — линия данных от контроллера к ведущему.

4) RESET — сигналом на RESET программатор вводит контроллер в режим программирования, потому что для разрешения прошивки по SPI нужно подать логический «0» на этот вывод.

5) GND — земля.

Доступны две разные схемы контактов разъёма ISP: 6-контактная и 10-контактная. Для программирования микроконтроллера был выбран Atmel 6-Pin ISP разъём.

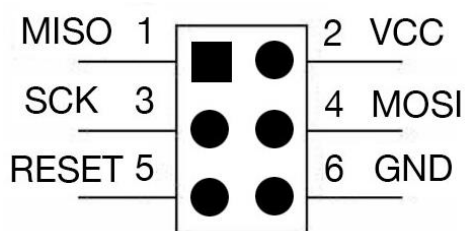


Рисунок 23 – Разъём Atmel 6-pin ISP

2.4.2 Алгоритм последовательного программирования через SPI

Взаимодействие устройств по интерфейсу SPI требует установки одного из устройств в режим ведущего, а остальных — в режим ведомого. При этом ведущее устройство отвечает за выбор ведомого и инициализацию передачи.

SPI является синхронным интерфейсом: все операции синхронизированы фронтами тактового сигнала (SCK), который вырабатывается ведущим устройством.

Программирование микроконтроллера по SPI осуществляется путём отправки 4-байтовых команд на вывод MOSI МК, в который один или два байта

определяют тип операции, остальные – адрес, записываемый байт, установочные биты и биты защиты, пустой байт. При выполнении операции чтения считываемый байт снимается через вывод MISO. Так же можно запрограммировать память данных EEPROM.

Во время последовательной записи в ATmega8515 данные тактируется нарастающим фронтом SCK. Во время чтения данные тактируются спадающим фронтом SCK. Временная диаграмма представлена на рисунке 17.

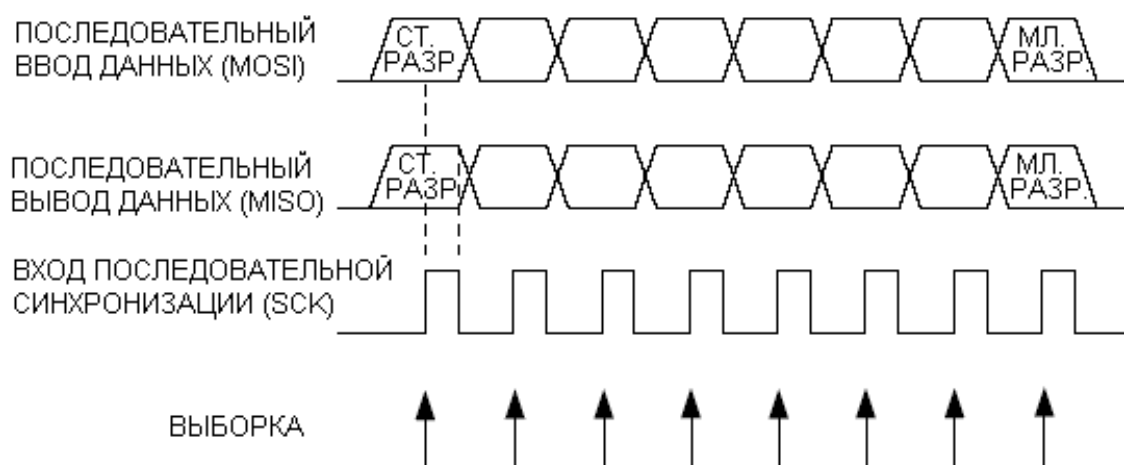


Рисунок 24 – Последовательное программирование оп интерфейсу SPI

1) Последовательность подачи питания: подать напряжение питания между VCC и GND, когда на входах RESET и SCK присутствует лог. 0. В некоторых системах программатор не может гарантировать, что SCK = 0 при подаче питания. В этом случае необходимо сформировать положительный импульс на RESET длительностью не менее двух тактов ЦПУ после того, как для SCK установлено значение «0».

2) Задержка не менее 20 мс и разрешение последовательного программирования путём записи команды разрешения последовательного программирования через вход MOSI.

3) Инструкции последовательного программирования не выполняются, если связь не синхронизирована. Когда связь синхронизирована, будет возвращаться значение второго байта (\$53) от МК при отправке третьего байта команды разрешения программирования. Независимо от того корректно или нет принятое значение, все четыре байта инструкции должны быть переданы.

Если принятое значение не равно \$53, то формируется положительный импульс на входе RESET и вводится новая команда разрешения программирования.

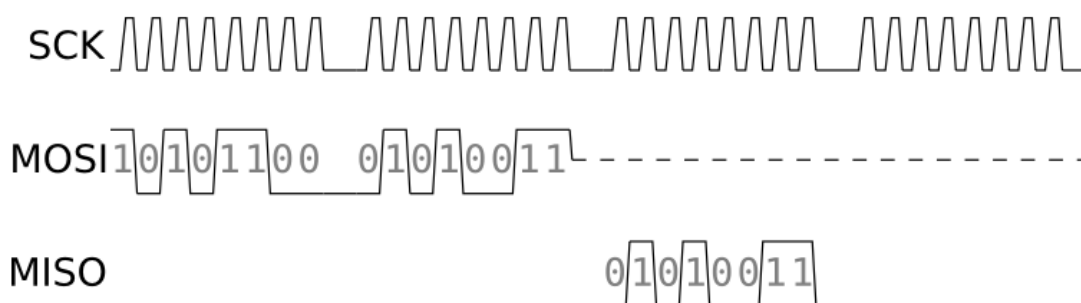


Рисунок 25 – Команда «Program Enable»

4) Flash память программируется постранично, размер страницы составляет 64 байта. В страницу памяти загружается по одному байту за раз, предоставляя 5 младших бит адреса и данные вместе с командой загрузки страницы памяти программ. Чтобы обеспечить корректную загрузку страницы, сначала необходимо записать младший байт, а затем старший байт данных по данному адресу. Страница памяти программ сохраняется путём загрузки инструкции «запись страницы памяти программ» с 7 младшими битами адреса страницы. Доступ к интерфейсу последовательного программирования до завершения операции записи во Flash может привести к неправильному программированию.

5) Массив памяти EEPROM программируется побайтно, при этом в инструкции записи указывается адрес и данные. Перед записью новых данных первоначально автоматически стирается адресуемая ячейка EEPROM.

6) Любую ячейку памяти можно проверить, используя инструкции чтения, которые возвращают содержимое ячейки по указанному адресу путём последовательной передачи на выходе MISO.

7) По завершении программирования вход RESET должен быть переведён в высокое состояние для возобновления нормальной работы.

8) Последовательность выключения (при необходимости): установка RESET = "1", отключить питание VCC.

Таблица 5 – Формат байтов команд для программирования

Инструкция	Формат инструкции				Функция
	Байт 1	Байт 2	Байт 3	Байт 4	
Разрешение программирования	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx	Разрешение последовательного программирования после подачи лог. 0 на RESET.
Стирание кристалла	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Стирание EEPROM и Flash памяти.
Чтение памяти программ	0010 H000	0000 aaaa	bbbb bbbb	oooo oooo	Чтение старшего (H=1) или младшего (H=0) байта данных о из памяти программ по адресу a:b.
Загрузка страницы памяти программ	0100 H000	0000 xxxx	xxxb bbbb	iiii iiii	Запись старшего (H=1) или младшего (H=0) байта данных i в страницу памяти программ по адресу b. Мл. байт данных должен быть загружен перед старшим байтом по тому же адресу.
Запись страницы памяти программ	0100 1100	0000 aaaa	bbbx xxxx	xxxx xxxx	Запись страницы памяти программ по адресу a:b.
Чтение EEPROM	1010 0000	00xx xxxa	bbbb bbbb	oooo oooo	Чтение данных o из EEPROM по адресу a:b.
Запись EEPROM	1100 0000	00xx xxxa	bbbb bbbb	iiii iiii	Запись данных i в EEPROM по адресу a:b.
Чтение бит защиты	0101 1000	0000 0000	xxxx xxxx	xxoo oooo	Чтение бит защиты. "0" - запрограммирован, "1" - не запрограммирован.
Запись бит защиты	1010 1100	111x xxxx	xxxx xxxx	11ii iiii	Запись бит защиты. Запись "0" приводит к программированию бита защиты.
Чтение сигнатурного байта	0011 0000	00xx xxxx	xxxx xxbb	oooo oooo	Чтение сигнатурного байта o по адресу b.
Запись конфигурационных бит	1010 1100	1010 0000	xxxx xxxx	liii iiii	Указывайте "0" для программирования, "1" для стирания.
Запись старших конфигурационных бит	1010 1100	1010 1000	xxxx xxxx	liii iiii	Указывайте "0" для программирования, "1" для стирания.
Чтение конфигурационных бит	0101 0000	0000 0000	xxxx xxxx	oooo oooo	Чтение конфигурационных бит. "0" - запрограммирован, "1" - не запрограммирован.
Чтение старших конфигурационных бит	0101 1000	0000 1000	xxxx xxxx	oooo oooo	Чтение старших конфигурационных бит. "0" = запрограммирован, "1" = не запрограммирован.
Чтение калибровочного байта	0011 1000	00xx xxxx	0000 00bb	oooo oooo	Чтение калибровочного байта o по адресу b.

Примечание:

- a - адрес старших разрядов;
- b - адрес младших разрядов;
- H - 0 - мл. байт, 1 - ст. байт;
- o - вывод данных;
- i - ввод данных;
- x - произвольное значение.

2.4.3 Опрос данных Flash памяти

Когда страница программируется во Flash, при чтении по адресам в пределах данной страницы возвращается \$FF. Микроконтроллер готов к записи новой страницы, если запрограммированное значение считано корректно.

Это используется для определения момента, когда может быть загружена следующая страница. Важно, что запись выполняется всей страницы одновременно, и любой адрес в пределах страницы может использоваться для опроса.

Опрос данных Flash не будет работать для значения \$FF, поэтому при записи этого значения необходимо подождать не менее t_{DD_FLASH} , прежде чем запрограммировать следующую страницу. Поскольку микроконтроллер с очищенной памятью содержит \$FF по всем адресам, можно пропустить повторную запись значения \$FF.

2.4.4 Опрос данных EEPROM

Когда новый байт был записан и в последующем программируется в EEPROM, чтение значения по этому адресу вернёт \$FF. Устройство готово к новому байту, если запрограммированное значение считываться корректно.

Это используется для определения момента, когда может быть записан следующий байт.

Данное не распространяется для значения \$FF, но программист должен обратить внимание на следующее: поскольку устройство с очищенной памятью содержит \$FF по всем адресам, программирование ячейки значением \$FF может быть пропущено.

Пропуск недопустим, если EEPROM перепрограммировано без предварительного стирания всей памяти. В этом случае опрос данных не может использоваться для значения \$FF, и программист должен предусмотреть задержку не менее t_{WD_EEPROM} , прежде чем запрограммировать следующий байт.

Таблица 6 – Минимальная задержка при записи

Обозначение	Минимальная задержка
t_{WD_FUSE}	4.5 ms
t_{WD_FLASH}	4.5 ms
t_{WD_EEPROM}	9.0 ms
t_{WD_ERASE}	9.0 ms

Заключение

В результате выполнения курсового проекта было получено функциональное, структурно и принципиальное описание разработанного устройства.

Разработаны алгоритмы функционирования микроконтроллера ATmega8515. Написан код программы на язык ассемблер без использования сторонних библиотек, функций и исходных кодов.

Разработанная МК-система представляет из себя устройство управления, осуществляющее управление 8 приборами жилого помещения согласно расписанию, получаемому с сервера, который представляет из себя ПЭВМ, передача которого осуществляется по USART по протоколу передачи RS-232.

Устройство обладает следующими важными при функционировании данной системы техническими характеристиками:

- 1) частота работы устройства составляет 8 МГц;
- 2) управляет до 8 приборами одновременно;
- 3) отправляет запросы по получению расписания на ПЭВМ;
- 4) отправляет запросы по получению текущего времени на ПЭВМ;
- 5) обладает пультом управления оператора на 16 кнопок;
- 6) устанавливает расписанию по умолчанию в случае отсутствия связи с сервером расписания;
- 7) способен хранить до 127 меток включения или выключения приборов при внутренней SRAM 512 Кбайт;
- 8) выводит текущее время МК-системы на LCD-дисплей;
- 9) работает от линии питания 12 В.

Список литературы

1. Хартов В.Я. Микроконтроллеры AVR. Практикум для начинающих. 2-е издание, Издательство МГТУ им. Баумана, 2012 г. – 278 с.
2. Хартов В.Я. Микропроцессорные системы: учебное пособие для студентов учреждения высшего профессионального образования, Академия, М., 2014 г. – 368 с.
3. Atmel ATmega8515 datasheet - doc2512, [Электронный ресурс] // ATmega8515 datasheet - doc2512: электронный документ ATmega8515(L) - Complete Datasheet URL: <http://ww1.microchip.com/downloads/en/DeviceDoc/doc2512.pdf> (дата обращения: 09.12.2019)
4. Производитель МК ATMEGA – компания Microchip [Электронный ресурс]. - URL: <https://www.microchip.com/> (дата обращения 09.12.2019)