# РЕФЕРАТ

Пояснительная записка состоит из 63 страниц, включает в себя 29 рисунков и 5 таблиц. При работе были использованы 5 источников, из которых 2 ссылок на Web-страницы Internet.

Ключевые слова: Микроконтроллер, Индикатор, Клавиатура, Регистр, Алгоритмизация, Кондиционер, Датчик, Программирование.

Цель работы: разработка системы управления кондиционером, осуществляющим ввод желаемой температуры в помещении, вывод текущей температуры помещения, сравнение этих двух температур и выбор режима работы на обогрев или на охлаждение соответственно.

Содержание работы: в работе выполнено построение структурной схемы, выбор элементной базы, оптимальной для реализации поставленных задач по дешевизне и диапазону характеристик, сформирована функциональная схема, которая в полной мере показывает работу системы.

# СОДЕРЖАНИЕ

# РЕФЕРАТ

# СОДЕРЖАНИЕ

# СОКРАЩЕНИЯ

# ВВЕДЕНИЕ

# ОСНОВНАЯ ЧАСТЬ

# 1 Структурная схема

# 2 Выбор элементной базы

# 2.1 Пульт управления

# 2.2 Работа с шиной 1-Wire

# 2.3 Выбор датчика температуры

# 2.4 Внутренняя архитектура микросхемы DS18B20

# 2.5 Система команд транспортного уровня микросхемы DSI8B20

# 2.6 Выбор вентилятора

# 2.7 Выбор компрессора

# 2.8 Выбор микроконтроллера

# 3 Функциональная схема

# 4 Алгоритм работы кондиционера

# 5 Разработка программы

# ЗАКЛЮЧЕНИЕ

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

# ПРИЛОЖЕНИЕ

# Приложение 1

# СОКРАЩЕНИЯ

СКВ – система кондиционирования воздуха

МК – микроконтроллер

ЖКИ – жидкокристаллический индикатор

ПЗУ – постоянное запоминающее устройство

ОЗУ – оперативное запоминающее устройство

# ВВЕДЕНИЕ

Кондиционирование воздуха (лат. condicio условие, требование) - это создание и автоматическое поддержание (регулирование) в закрытых помещениях всех или отдельных параметров (температуры, влажности, чистоты, скорости движения) воздуха на определенном уровне с целью обеспечения оптимальных метеорологических условий, наиболее благоприятных для самочувствия людей или ведения технологического процесса.

Основные функции кондиционера - это охлаждение и обогрев воздуха, уже находящегося внутри помещения. Это означает, что кондиционер в общем случае не производит притока свежего воздуха с улицы или вытяжки воздуха из помещения. Для задач вытяжки и притока служит вентиляционное оборудование.

Нагрев - осуществляется электрическим калорифером. Поддержание температуры осуществляется ступенчатым регулированием мощности электрического нагрева.

Охлаждение - процесс охлаждения воздуха происходит в секции охлаждения, которая представляет собой радиатор, в котором происходит кипение фреона, т.е. по сути это испаритель кондиционера поступающего из компрессорно-конденсаторного блока. Степень охлаждения воздуха регулируется дискретным включением компрессора воздушно-конденсаторного блока, таким образом точно поддерживать температуру приточного воздуха невозможно.

1 Признаки классификации систем кондиционирования

Прежде чем перейти к классификации систем кондиционирования, следует отметить, что общепринятой классификации СКВ до сих пор не существует, и связано это с многовариантностью принципиальных схем, технических и функциональных характеристик, зависящих не только от технических возможностей самих систем, но и от объекта применения (кондиционируемых помещений). Современные системы кондиционирования могут быть классифицированы по следующим признакам [1]:

а) по основному назначению (объекту применения):

- комфортные

- технологические

б) по принципу расположения кондиционера по отношению к обслуживаемому помещению:

- центральные

- местные

в) по наличию собственного (входящего в состав кондиционера) источника тепла и холода:

- автономные

- неавтономные

г) по количеству обслуживаемых помещений (локальных зон):

- однозональные

- многозональные

д) по принципу действия:

- приточные

- рециркуляционные

- комбинированные

е) по способу регулирования выходных параметров кондиционированного воздуха:

- с качественным (однотрубным )

- с количественным (двухтрубным ) регулированием

ж) по степени обеспечения метеорологических условий в обслуживаемом помещении - первого, второго и третьего классов

з) по давлению, развиваемому вентиляторами кондиционеров - низкого, среднего и высокого давления.

и) Кроме приведенных выше классификаций существуют разнообразные системы кондиционирования, обслуживающие специальные технологические процессы, включая системы с изменяющимися по времени (по определенной программе) метеорологическими параметрами.

Комфортные СКВ

Комфортные системы кондиционирования воздуха предназначены для создания и автоматического поддержания температуры, относительной влажности, чистоты и скорости движения воздуха, отвечающих оптимальным санитарно-гигиеническим требованиям для жилых, общественных и административно-бытовых зданий или помещений.

Технологические СКВ

Технологические системы кондиционирования воздуха предназначены для обеспечения параметров воздуха, в максимальной степени отвечающих требованиям производства. Технологическое кондиционирование в помещениях, где находятся люди, осуществляется с учетом санитарно-гигиенических требований к состоянию воздушной среды

Центральные СКВ

Снабжаются извне холодом (доставляемым холодной водой или хладагентом), теплом (доставляемым горячей водой, паром или электричеством) и электрической энергией для привода электродвигателей вентиляторов, насосов и пр.

Центральные системы кондиционирования воздуха расположены вне обслуживаемых помещений и кондиционируют одно большое помещение, несколько зон такого помещения или много отдельных помещений. Иногда несколько центральных кондиционеров обслуживают одно помещение больших размеров (производственный цех, театральный зал, закрытый стадион или каток).

Центральные СКВ оборудуются центральными неавтономными кондиционерами, которые изготавливаются по базовым (типовым) схемам компоновки оборудования и их модификациям.

Центральные системы кондиционирования воздуха обладают следующими преимуществами:

- Возможностью эффективного поддержания заданной температуры и относительной влажности воздуха в помещениях.

- Сосредоточением оборудования, требующего систематического обслуживания и ремонта, как правило, в одном месте (подсобном помещении, техническом этаже и т.п.)

- Возможностями обеспечения эффективного шумо- и виброгашения. С помощью центральных СКВ при надлежащей акустической обработке воздуховодов, устройстве глушителей шума и гасителей вибрации можно достичь наиболее низких уровней шума в помещениях и обслуживать такие помещения, как радио- и телестудии и т.п.

Несмотря на ряд достоинств центральных СКВ, надо отметить, что крупные габариты и проведение сложных монтажно-строительных работ по установке кондиционеров, прокладке воздуховодов и трубопроводов часто приводят к невозможности применения этих систем в существующих реконструируемых зданиях

Местные СКВ

Местные системы кондиционирования воздуха разрабатывают на базе автономных и неавтономных кондиционеров, которые устанавливают непосредственно в обслуживаемых помещениях. Достоинством местных СКВ является простота установки и монтажа. Такая система может применяться в большом ряде случаев:

- В существующих жилых и административных зданиях для поддержания теплового микроклимата в отдельных офисных помещениях или в жилых комнатах

- Во вновь строящихся зданиях для отдельных комнат, режим потребления холода в которых резко отличается от такого режима в большинстве других помещений, например, в серверных и других насыщенных тепловыделяющей техникой комнатах административных зданий. Подача свежего воздуха и удаление вытяжного воздуха при этом выполняется, как правило, центральными системами приточно-вытяжной вентиляции

- Во вновь строящихся зданиях, если поддержание оптимальных тепловых условий требуется в небольшом числе помещений, например, в ограниченном числе номеров люкс небольшой гостиницы

- В больших помещениях как существующих, так и вновь строящихся зданий: кафе и ресторанах, магазинах, проектных залах, аудиториях и т.д.

Автономные СКВ

Автономные системы кондиционирования воздуха снабжаются извне только электрической энергией, например, кондиционеры сплит-систем, шкафные кондиционеры и т.п. Такие кондиционеры имеют встроенные компрессионные холодильные машины, работающие, как правило, на фреоне-22. Автономные системы охлаждают и осушают воздух, для чего вентилятор продувает рециркуляционный воздух через поверхностные воздухоохладители, которыми являются испарители холодильных машин, а в переходное или зимнее время они могут производить подогрев воздуха с помощью электрических подогревателей или путем реверсирования работы холодильной машины по циклу так называемого "теплового насоса".

Наиболее простым вариантом, представляющим децентрализованное обеспечение в помещениях температурных условий, можно считать применение кондиционеров сплит-систем.

Неавтономные СКВ

Неавтономные системы кондиционирования воздуха подразделяются на:

- Воздушные, при использовании которых в обслуживаемое помещение подается только воздух. (мини-центральные кондиционеры, центральные кондиционеры)

- Водовоздушные, при использовании которых в обслуживаемые помещения подводятся воздух и вода, несущие тепло или холод, либо и то и другое вместе (системы чиллеров-фанкойлов, центральные кондиционеры с местными доводчиками и т.п.)

Однозональные центральные СКВ

Однозональные центральные системы кондиционирования воздуха применяются для обслуживания больших помещений с относительно равномерным распределением тепла и влаговыделений, например, больших залов кинотеатров, аудиторий и т.п. Такие СКВ, как правило, комплектуются устройствами для утилизации тепла (теплоутилизаторами) или смесительными камерами для использования в обслуживаемых помещениях рециркуляции воздуха.

Многозональные центральные СКВ

Многозональные центральные системы кондиционирования воздуха применяют для обслуживания больших помещений, в которых оборудование размещено неравномерно, а также для обслуживания ряда сравнительно небольших помещений. Такие системы более экономичны, чем отдельные системы для каждой зоны или каждого помещения. Однако с их помощью не может быть достигнута такая же степень точности поддержания одного или двух заданных параметров (влажности и температуры), как автономными СКВ (кондиционерами сплит-систем и т.п.)

Прямоточные СКВ

Прямоточные системы кондиционирования воздуха полностью работают на наружном воздухе, который обрабатывается в кондиционере, а затем подается в помещение.

Рециркуляционные СКВ

Рециркуляционные системы кондиционирования воздуха, работают без притока или с частичной подачей (до 40%) свежего наружного воздуха или на рециркуляционном воздухе (от 60 до 100%), который забирается из помещения и после его обработки в кондиционере вновь подается в это же помещение. Классификация кондиционирования воздуха по принципу действия на прямоточные и рециркуляционные обуславливается, главным образом, требованиями к комфортности, условиями технологического процесса производства либо технико-экономическими соображениями.

СКВ с качественным регулированием

Центральные системы кондиционирования воздуха с качественным регулированием метеорологических параметров представляют собой широкий ряд наиболее распространенных, так называемых одноканальных систем, в которых весь обработанный воздух при заданных кондициях выходит из кондиционера по одному каналу и поступает далее в одно или несколько помещений. При этом регулирующий сигнал от терморегулятора, установленного в обслуживаемом помещении, поступает непосредственно на центральный кондиционер.

СКВ с количественным регулированием

Системы кондиционирования воздуха с количественным регулированием подают в одно или несколько помещений холодный или подогретый воздух по двум параллельным каналам. Температура в каждом помещении регулируется комнатным терморегулятором, воздействующим на местные смесители (воздушные клапаны), которые изменяют соотношение расходов холодного и подогретого воздуха в подаваемой смеси.

Двухканальные системы используются очень редко из-за сложности регулирования, хотя и обладают некоторыми преимуществами, в частности, отсутствием в обслуживаемых помещениях теплообменников, трубопроводов тепло-холодоносителя, возможностью совместной работы с системой отопления, что особенно важно для существующих зданий, системы отопления которых при устройстве двухканальных систем могут быть сохранены. Недостатком таких систем являются повышенные затраты на тепловую изоляцию параллельных воздуховодов, подводимых к каждому обслуживаемому помещению.

Двухканальные системы так же, как и одноканальные, могут быть прямоточными и рециркуляционными.

Степень обеспечения метеорологических условий

Кондиционирование воздуха, согласно СниП 2.04.05-91, по степени обеспечения метеорологических условий, подразделяется на три класса:

- Первый класс - обеспечивает требуемые для технологического процесса параметры в соответствии с нормативными документами.

- Второй класс - обеспечивает оптимальные санитарно-гигиенические нормы или требуемые технологические нормы.

- Третий класс - обеспечивает допустимые нормы, если они не могут быть обеспечены вентиляцией в теплый период года без применения искусственного охлаждения воздуха.

Создаваемое вентиляторами давление

По давлению, создаваемому вентиляторами центральных кондиционеров, системы кондиционирования воздуха подразделяются на системы

- Низкого давления (до 100 кг/кв.м.),

- Среднего давления (от 100 до 300 кг/кв.м.)

- Высокого давления (выше 300 кг/кв.м.).

2 Структура мирового и российского рынков кондиционеров

На рисунке1 приведены данные продаж 1999 г. по странам Европы (оконные, мобильные кондиционеры и настенные сплит-системы) [1]. Как и следовало ожидать, лидерами потребления климатических систем были страны Южной Европы — Испания, Италия и Греция. Непосредственно за ними следовала Россия (около 160 тыс. единиц). Однако по продажам на душу населения (рисунок 2) наша страна опережает только относительно прохладную Германию.

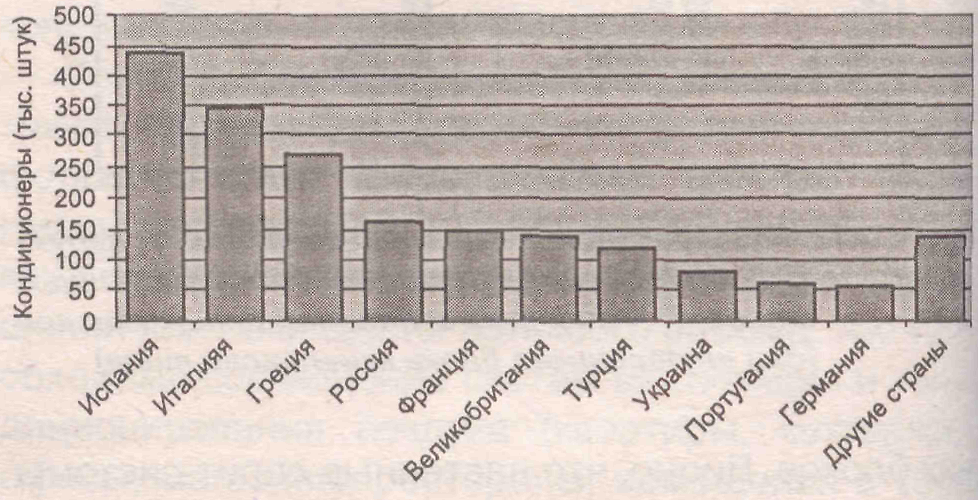


Рисунок 1 - Объемы продаж кондиционеров в странах Европы (в штучном исчислении, 1999 г.)

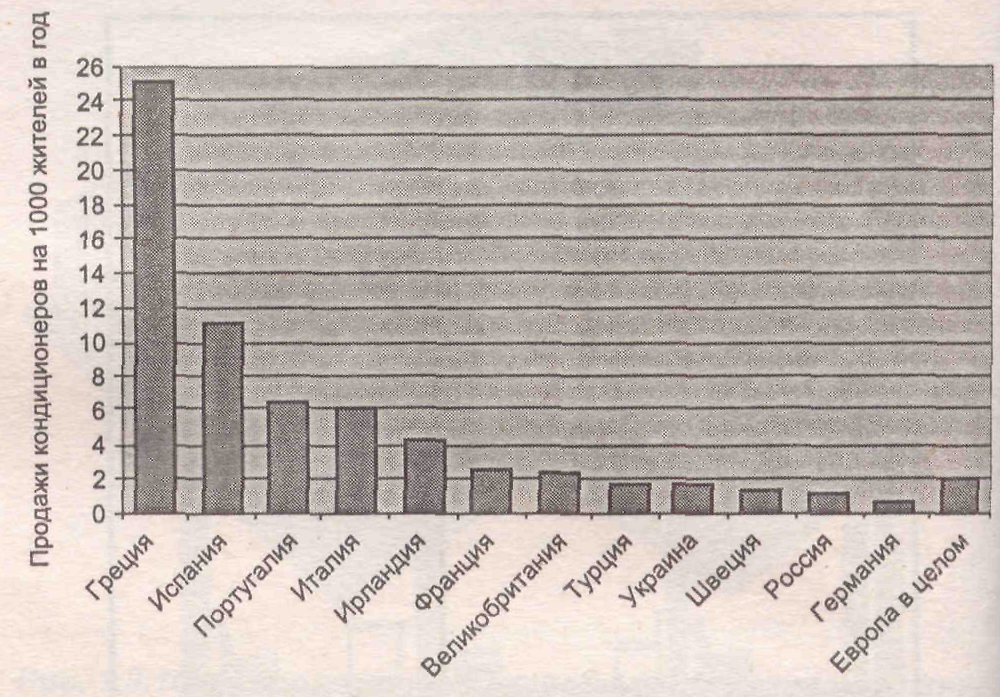


Рисунок 2 - Продажи кондиционеров в единицах на 1000 жителей.

Динамика роста российского рынка кондиционеров приведена на рисунке 3.

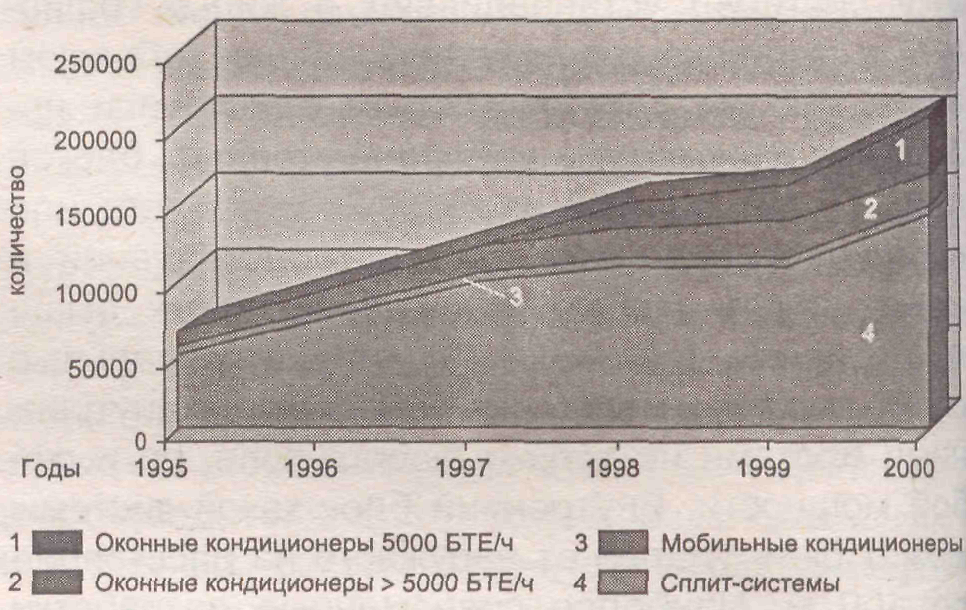


Рисунок 3 - Продажи кондиционеров различных типов на российском рынке в 1995 — 2000 гг

2001 г. стал рекордным для российского рынка, который впервые перешагнул отметку в 300 тыс. кондиционеров и теперь занимает третье место в Европе. На Россию теперь приходится 11... 12% европейского рынка, а по темпам продаж за последние годы мы существенно обгоняем соседей по континенту.

На рисунках 4 и 5 приведены доли продаж различных марок бытовых и полупромышленных климатических систем классов RAC и РАС на российском рынке в 2001 г. в количественном и стоимостном выражении соответственно [1]. Как видно из сравнения графиков, ценовые и количественные доли большинства марок не совпадают.

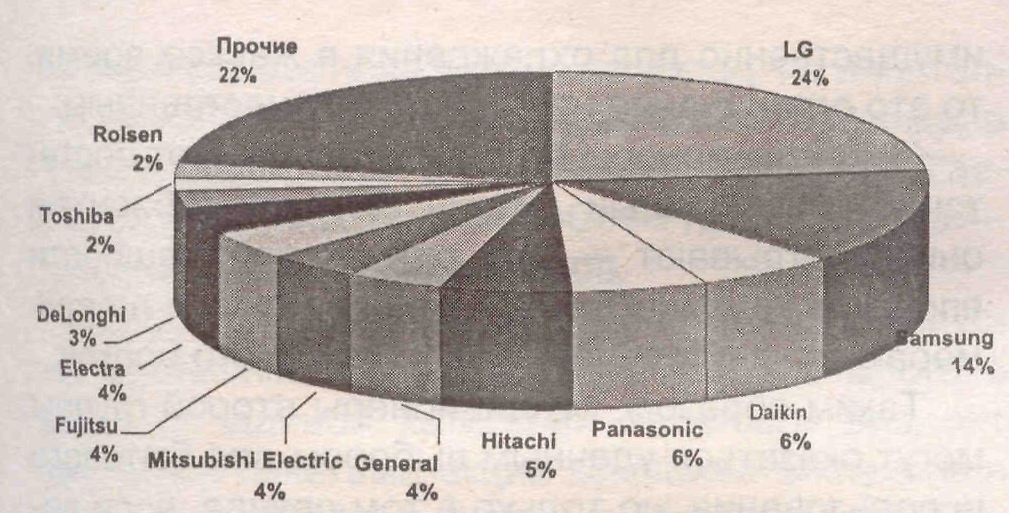


Рисунок 4 - Доли продаж различных марок климатических систем на российском рынке в 2001 г. в количественном выражении

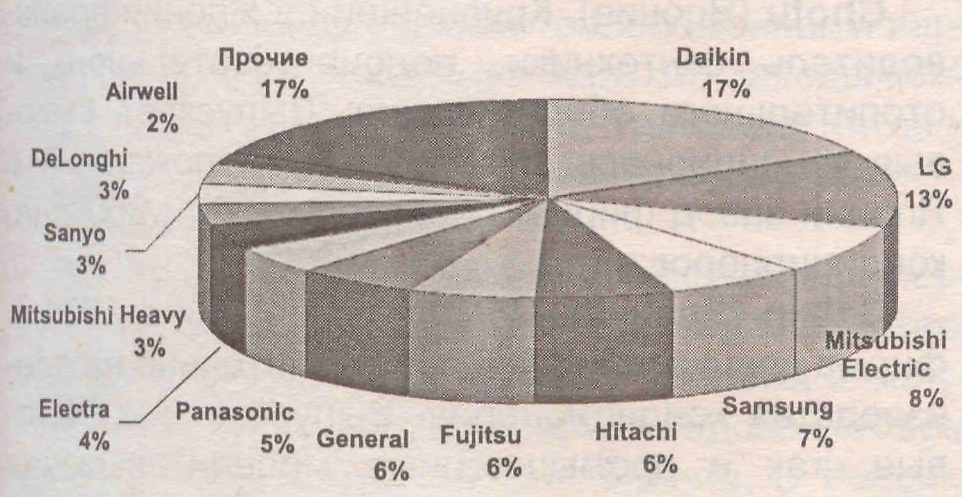


Рисунок 5 - Доли продаж различных марок климатических систем на российском рынке в 2001 г. в ценовом выражении

С точки зрения качества и надежности выпускаемой техники, все бренды можно условно разделить на две большие группы: в первую входят японские фирмы (Chofu, Daikin, Fujitsu General, Hitachi, Mitsubishi Heavy, Mitsubishi Electric, Panasonic, Sanyo, Sharp, Toshiba), во вторую — корейские (LG, Samsung) и израильские (Electra, Tadi-ran). Производители из Европы (AirWell, Argo, DeLonghi) и США (Carrier, McQuay, White-Westin-ghouse, York) по стоимости, надежности и другим характеристикам занимают промежуточное положение между этими группами.

# ОСНОВНАЯ ЧАСТЬ

## 1 Структурная схема

Структурная схема данной системы управления кондиционером представлена на Рисунке 6.



Рисунок 6 – Структурная схема

## 2 Выбор элементной базы

## 2.1 Пульт управления

Пульт управления кондиционером содержит два ЖКИ индикатора текущей температуры помещения и задаваемой пользователем (поддерживаемой в помещении) температуры и две кнопки изменения задаваемой пользователем температуры в сторону увеличения и уменьшения (шаг изменения температуры – 1°С). При включении кондиционера задаваемая пользователем температура будет по умолчанию равна 24°С, как наиболее комфортная температура при небольшой физической нагрузке (ходьба в помещении, легкий труд) в летнее время.

Для отображения температур выберем ЖКИ модуль типа МТ-10Е7-7 российской фирмы МЭЛТ.

Основные характеристики модуля МТ-10Е7-7:

тип индикатора – цифровой семисегментный;

количество строк – 1;

количество разрядов – 10;

напряжение питания – минимальное +3В, максимальное +5В;

ток потребления – 30мкА;

способ регулировки контрастности – ручной (внешний резистор);

количество выводов – 12;

габаритные размеры – 66х31,5х9,5 мм.

Модуль МТ-10Е7-7 – это недорогой, достаточно распространенный однострочный индикатор, содержащий десять семисегментных разрядов для вывода цифровой информации. Выводы модуля выполнены в виде контактных площадок на печатной плате с отверстиями для пайки проводов. Назначение выводов модуля показано в таблице 1. Подпрограммы вывода симвомов и строк на индикаторы показаны в приложении 1.

Таблица 1 – Назначение выводов модуля МТ – 10Т7-7

|  |  |  |  |
| --- | --- | --- | --- |
| Номер вывода | Название цепи | Назначение | |
| 1 | A0 | Вход выбора «адрес/данные» | |
| 2 | WR2 | Инверсный вход синхронизации записи | |
| 3 | WR1 | Прямой вход синхронизации записи | |
| 4 | DB3 | Разряд 3 | Шина данных/адреса |
| 5 | DB2 | Разряд 2 |
| 6 | DB1 | Разряд 1 |
| 7 | DB0 | Разряд 0 |
| 8 | GND | Общий провод | |
| 9 | V0 | Вход управления контрастностью | |
| 0 | +E | Питание модуля | |
| 11 | +L | Не используется | |
| 12 | -L | Не используется | |

## 

## 2.2 Работа с шиной 1-Wire

Шина 1-Wire [2] построена по технологии Master / Slave. То есть, на шине должно быть хотя бы одно ведущее устройство (Master). Все остальные устройства должны быть ведомыми (Slave). Ведущее устройство инициирует все процессы передачи информации в пределах шины. Master может прочитать данные из любого Slave устройства или записать их туда. Передача информации от одного Slave к другому напрямую невозможна. При разработке протокола 1-Wire большое внимание было уделено надежности работы сети. Изначально было поставлено условие – работа должна происходить в условиях плохих контактов.

Рассмотрим принципиальную электрическую схему, реализующую 1-Wire интерфейс. Схема соединения ведущего и ведомого устройств посредством однопроводной шины приведена на рисунке 7. На этом рисунке также показаны особенности схемной реализации выходных каскадов ведущего и ведомого устройств. В схеме 1-Wire интерфейса используются выходные каскады с открытым коллектором (стоком) и общей нагрузкой RH для всех элементов сети. В спецификации для 1-Wire интерфейса специально оговаривается, что резистор RH должен находиться в непосредственной близости от ведущего устройства.

Биполярный транзистор в выходном каскаде ведущего устройства показан условно. С не меньшим (а скорее большим) успехом можно применять микросхемы, у которых выходные каскады построены по КМОП технологии. Ведомые устройства обычно построены на КМОП транзисторах. В режиме ожидания все выходные транзисторы закрыты. На шине присутствует напряжение логической единицы. Информация по шине передается при помощи отрицательных импульсов. Любое устройство, подключенное к шине, как ведущее, так и ведомое, может создавать отрицательные импульсы и тем самым передавать информацию. Однако ведомое устройство начинает процесс передачи только под управлением ведущего.

В составе ведомого устройства имеется источник тока, показанный на рисунке 7 двумя пересекающимися окружностями. Этот источник создает внутреннюю утечку на входе 1-Wire интерфейса. Смысл этой утечки – создать нулевой уровень сигнала на внутренних элементах ведомого устройства при его отключении от шины 1-Wire .Когда соединение будет восстановлено, внутренняя логика ведомой микросхемы обнаруживает перепад напряжения с нуля на единицу. Сразу после получения такого сигнала ведомая микросхема должна выдать на шину 1-Wire сигнал присутствия.

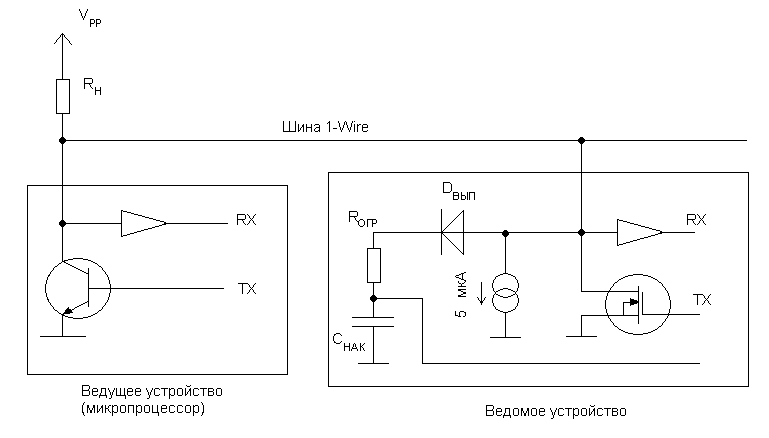


Рисунок 7 – Электрическая схема 1-Wire интерфейса

Каждое из подключенных устройств (ведущее и ведомое) должно содержать специальное управляющее устройство, реализующее протокол шины 1-Wire. Именно протокол определяет все правила передачи информации.

Протокол 1-Wire имеет несколько разных уровней. Самый низкий уровень описывает, каким образом передаются отдельные биты. При этом предусмотрен двусторонний обмен информацией. Все операции на шине производятся исключительно под управлением Master устройства. Оно может выполнять операции двух видов: записывать информацию в Slave устройство и считывать информацию из него. Информация передается побайтно, в последовательном виде, бит за битом, начиная с младшего бита. В обоих случаях Master устройство вырабатывает на шине тактовые импульсы. Для этого оно периодически «подсаживает» шину при помощи своего 1-Wire интерфейса. Полезная информация передается путем изменения длительности этих импульсов.

При записи информации длительностью импульсов управляет Master устройство. В режиме чтения Master устройство начинает формирование импульса, но Slave устройство может продлевать длительность любого импульса, «подсаживая» в свою очередь сигнал на линии в нужный момент. На рисунке 8 изображены две временные диаграммы. Верхняя диаграмма иллюстрирует режим записи двух разных битов информации, а нижняя – режим чтения. Участки диаграммы, где линия «отпущена» и уровень сигнала на линии определяется резистором RH , изображены на диаграмме при помощи тонких линий. Участки, где один из элементов сети «подсаживает» линию, изображены при помощи толстых линий.

Рассмотрим, как происходит запись бита. В исходном состоянии все Slave устройства, подключенные к шине, находятся в режиме ожидания. Линия «отпущена». То есть выходные транзисторы всех элементов шины закрыты, и напряжение на шине определяется резистором нагрузки RH . Для того, чтобы записать данные в одно из Slave устройств, Master начинает формировать отрицательные синхроимпульсы (верхняя диаграмма на рисунке 8). На каждый передаваемый бит формируется один импульс. Импульсы передаются путем «подсаживания» линии до нуля. Для передачи каждого бита выделяется промежуток времени стандартной длительности. Этот промежуток получил название «слот» (Slot). Как видно из рисунка 8, величина слота для передачи одного бита информации (Tx) должна лежать в пределах от 60 до 120 мкс.

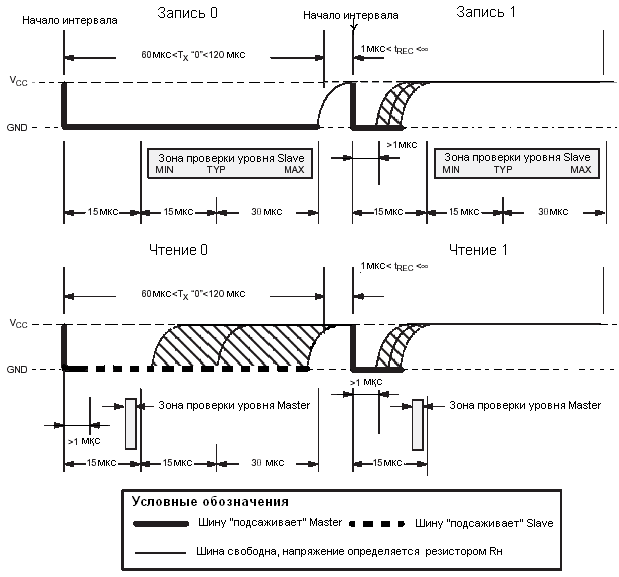


Рисунок 8 – Временная диаграмма процесса записи и чтения одного бита

Если значение передаваемого бита равно 0, то Master вырабатывает «длинный» импульс. Его длина равна длительности слота. Для передачи «единичного» бита Master вырабатывает «короткий» импульс, который, по сути, является чистым синхроимпульсом (1 мкс). Оставшаяся часть сигнала должна быть заполнена единичным сигналом. Между двумя слотами должен быть небольшой промежуток, во время которого сигнал на шине тоже равен единице. Slave устройство в этом режиме лишь принимает сигнал. Для этого оно постоянно находится в режиме ожидания. Обнаружив начало синхроимпульса, Slave устройство начинает процесс приема информации. Передний фронт этого импульса служит Slave устройству началом отсчета. Выдержав паузу, равную длительности синхроимпульса, Slave устройство считывает уровень сигнала на линии. Если в этот момент времени уровень сигнала на линии равен нулю, значит и передаваемый бит равен нулю. Если же сигнал будет равен единице, то бит равен единице. Протокол шины 1-Wire жестко определяет длительность слота. Интервал между слотами в свою очередь имеет ограничение только на минимальное значение – 1мкс. Максимальное значение этого интервала неограниченно. С помощью этого интервала может легко регулироваться скорость передачи данных от своего максимального значения (16,3 Кбит/с) практически до нуля.

Отличие процесса чтения бита (нижняя диаграмма на рисунке 8) от записи состоит в том, что Master вырабатывает только синхроимпульсы (короткой длительности). Обнаружив синхроимпульс, Slave устройство должно удлинить или не удлинять этот синхроимпульс в пределах слота. Если очередной считываемый бит равен нулю, то синхроимпульс удлиняется Slave устройством, если же он равен единице, то удлинения не происходит. На рисунке 8 участки временной диаграммы, где линию «подсаживает» Master устройство, изображены толстой сплошной линией. Участки, которые «подсаживает» Slave устройство изображены прерывистой сплошной линией. Master устройство считывает эту информацию, контролируя уровень сигнала внутри слота сразу после синхроимпульса.

Для надежной работы однопроводного интерфейса необходимо, чтобы в процессе передачи информации всеми элементами сети строго соблюдались временные параметры. При этом более жесткие требования предъявляются к Master устройству, так как в его роли выступает микроконтроллер, способный с высокой точностью отрабатывать любые временные интервалы. При записи ведомое устройство (в нашем случае датчик температур), обнаружив на шине передний фронт синхроимпульса, должно сформировать задержку минимум в 15 мкс и затем произвести проверку сигнала на шине. Допустимый разброс времени задержки лежит в пределах от 15 до 60 мкс. Этот диапазон показан на рисунке 8 в виде области, обозначенной как «Зона проверки уровня Slave». В режиме чтения бита Master вырабатывает только синхроимпульсы, длительность которых равна 1 мкс. Если читаемый бит равен нулю, Slave устройство продлевает длительность синхроимпульса. Минимальная длительность продленного импульса составляет 15 мкс. Для этого временного интервала тоже допускается довольно значительный разброс: длительность удлиненного импульса может вырасти еще на 45 мкс. Если читаемый бит равен единице, удлинения синхроимпульса не происходит. Таким образом, для того, чтобы правильно оценить значение читаемого бита, Master устройство должно прочитать уровень сигнала на шине сразу после окончания синхроимпульса, но не позднее, чем через 15 мкс. Зона проверки для Master устройства в режиме чтения значительно уже аналогичной зоны для Slave устройства в режиме записи.

Ряд отдельных бит при чтении или записи формируется в байты. Байты передаются младшим битом вперед. Первые восемь битов – это первый байт. Следующие восемь – второй байт, и так далее. Начало всей этой цепочки определяется сигналом сброса. Импульс сброса – это сверхдлинный отрицательный импульс на шине 1-Wire, вырабатываемый ведущим устройством. Временная диаграмма, иллюстрирующая процесс формирования импульса сброса, приведена на рисунке 9. С импульсом сброса тесно связан еще один служебный сигнал – сигнал присутствия на шине. Сигнал присутствия вырабатывает каждое Slave устройство сразу после окончания действия импульса сброса. Master устройство должно проконтролировать наличие этого импульса. Если импульса нет, значит на линии нет ни одного Slave устройства.

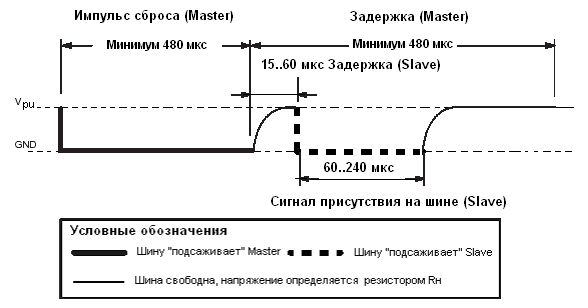


Рисунок 9 - Временная диаграмма процесса начального сброса

Кроме инициации импульсов присутствия импульс сброса переводит в исходное состояние всю систему. Любые незаконченные процессы на линии моментально завершаются, и отсчет битов начинается с начала. Длительность импульса сброса должна быть не менее 480 мкс. Процесс передачи информации по линии может начинаться не раньше, чем через 480 мкс после окончания действия импульса сброса (рисунок 9). В этом временном интервале и ожидается появление сигнала присутствия. Для этого после окончания импульса сброса Master «отпускает» линию и ждет сигнала от Slave устройства. Slave устройство, обнаружив импульс сброса, выдерживает паузу в 15..60 мкс, а затем «подсаживает» линию. Длительность импульса присутствия составляет 60..240 мкс. Ведущее устройство проверяет наличие нулевого уровня на линии в середине этого интервала. Если сигнал обнаружен, то это значит, что на линии имеется нормально работающее Slave устройство и Master может продолжать работу в сети. Если сигнал не обнаружится, микроконтроллер перейдет к обработке этой ситуации.

Любая операция в сети Master устройство - Slave устройство начинается с команды. Команда представляет собой байт информации. Каждая команда имеет свой собственный код. Выполнение команды начинается с импульса сброса. Затем Slave устройство вырабатывает, а Master устройство проверяет сигнал присутствия на линии. Если сигнал на месте, Master выдает на линию код команды в режиме записи. Получив этот код, Slave устройство переключается в режим выдачи информации.

При описании протокола 1-Wire принято выделять два уровня, связанных с логикой работы протокола – сетевой и транспортный уровни. Сразу после сигнала сброса шина переходит на сетевой уровень. Отработав команду сетевого уровня, шина переходит на транспортный. Система команд сетевого уровня протокола 1-Wire включает 5 команд: Чтение ПЗУ, Чтение ПЗУ (для микросхемы DS1990A команда имеет другой код), Совпадение ПЗУ, Пропуск ПЗУ, Поиск ПЗУ. Все эти команды, кроме команды Пропуск ПЗУ, не используются в случае, если в сети имеется лишь одно устройство. Поскольку для выполнения требований технического задания одного датчика температуры вполне достаточно, ограничимся описанием команды Пропуск ПЗУ (Skip ROM). Эта команда (код – 0ССН) позволяет перейти на транспортный уровень всем устройствам в сети, оставляя их активными. В нашем случае устройство всего одно.

После команды сетевого уровня все элементы сети переходят на транспортный уровень. Список команд этого уровня разнится для каждого устройства. Команды электронного датчика температуры DS18B20 будут рассмотрены в данной работе позднее. Следует отметить, что для выполнения очередной команды после выполнения команды транспортного уровня необходимо произвести импульс начального сброса.

Теперь приступим непосредственно к увлекательному описанию интегрального датчика температуры фирмы Dallas Semiconductor DS18B20.

## 2.3 Выбор датчика температуры

Фирма Dallas Semiconductor выпускает целый набор микросхем-измерителей температуры. Самый первый интегральный термодатчик с 1-Wire интерфейсом назывался DS1820. Эта оригинальная микросхема сразу после своего появления приобрела популярность у разработчиков электронной аппаратуры.

DSI8B20 - более совершенная микросхема. Высокая дискретность достигается увеличением количества разрядов результирующего кода. Причем в микросхеме имеется возможность изменения количества .разрядов выходного регистра. По умолчанию выходной регистр имеет 9 разрядов. Изменяя содержимое регистра конфигурации, микроконтроллер может увеличить количество разрядов до 12. Точность измерения температуры в диапазоне —10...+85\*С составляет -±0,5°С. На выходе микросхемы DS18B20 мы получаем прямой код. значение которого равно величине измеряемой температуры. В 9-разрядном режиме значение измеряемой температуры выдается -С дискретностью в 0,5'С. В двенадцатиразрядном режиме количество отсчетов повышается в восемь раз. Максимальное время преобразования для микросхемы DS18B20 также зависит от выбранного количества разрядов. Для 12-разрядного режима работы оно равно 750 мс.

## 2.4 Внутренняя архитектура микросхемы DS18B20

Микросхема DS18B20 выпускается в двух модификациях [2], Они отличаются исключительно конструкцией корпусу, На рисунке 10 приведен внешний вид обеих модификаций микросхемы. Основной вариант микросхемы выполнен в миниатюрном пластмассовом корпусе типа ТО-92. Второй вариант заключен в планарный восьмивыводной, миниатюрный корпус типа SOIC. Для того, чтобы различать эти два варианта исполнения, второй вариант получил обозначение DSI8B20Z. Микросхема имеет всего три задействованных вывода: DO-вход/выход данных 1-Wire интерфейса; VDD — вывод внешнего питания: GND — общий провод. Расположение выводов показано на рисунке 10.

Внутренняя структура микросхемы DS18B20 приведена на рисунке 11. Сигнал с шины DQ и напряжение с внешнего вывода питания (VDD ) прежде всего поступают на схему паразитного питания. Однако в схеме паразитного питании имеется еще один дополнительный элемент, о котором не говорилось ранее. Это датчик наличия питания. Датчик представляет собой пороговый элемент, на который поступает напряжение питания от внешнего источника. Датчик вырабатывает логически сигнал, поступающий в схему управления. В результате микросхема получает возможность автоматически определять режим своего питания. Микроконтроллер, работающий в качестве Master устройства на той же шине, имеет возможность запросить у всех подключенных к ней датчиков информацию о режиме питания и соответствующим образом скорректировать алгоритм своей работы.

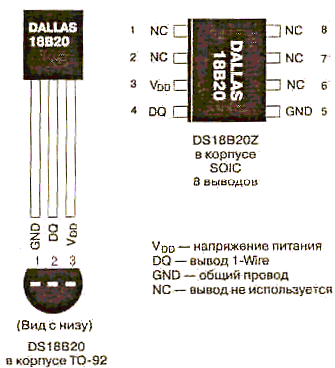


Рисунок 10 - Внешний вид микросхем DS18B20 в двух разных исполнениях

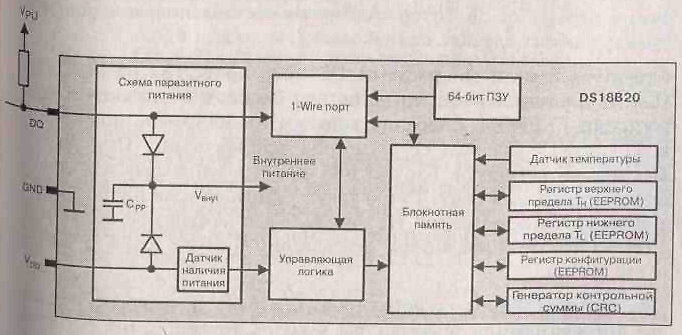


Рисунок 11 - Внутренняя структура микросхемы DS18B20

Сигнал DQ, обеспечив напряжением схему паразитного питания поступает на 1-Wire порт, который служит аппаратной частью одно проводного интерфейса. Данные, полученные при помощиэтого интерфейса, поступают в блокнотную память. Блокнотная память предназначена для временного хранения информации от датчика температуры и трех специальных регистров: регистра верхнего предела (Тн), регистра нижнего предела (TL) и регистра конфигурации. Все три специальных регистра представляют собой три ячейки флэш-памяти (EEPROM).

С блокнотной памятью также связан генератор контрольной суммы. Этот генератор автоматически вычисляет контрольную сумму всех регистров блокнотной памяти. При считывании информации из блокнотной памяти контрольная сумма также читается и служит для проверки правильности прочитанной информации. Применение блокнотной памяти позволяет повысить надежность передачи информации. Информация никогда не записывается непосредственно в ячейки флэш-памяти (регистры Тн и TL и регистр конфигурации). Предварительно она помещается в блокнотную память. Затем микроконтроллер читает ее оттуда и проверяет контрольную сумму. Если результат проверки положительный, микроконтроллер подает по шине специальную команду «Копирование блокнотной памяти в EEPROM».

Посредством 1-Wire интерфейса можно также прочитать содержимое 64-битного ПЗУ, в котором хранится ID код микросхемы. Последние восемь битов ID кода представляют собой контрольную сумму первых ее 56 битов.

Структура памяти микросхемы DS18B20 приведена на рисунке 12. Память состоит из восьми регистров блокнотной памяти и трех регистров EEPROM. Операции записи и чтения блокнотной памяти выполняются для всех ее регистров одновременно. При; записи все восемь регистров блокнотной памяти записываются одним блоком из восьми байт. Точно также одним блоком происходит считывание информации. На рисунке 12 для каждого регистра обозначена его позиция внутри передаваемого блока (байт 0, байт 1 и так далее).

Два самых младших регистра (байт 0 и байт 1) содержат результат преобразования температуры в код. Следующие три регистра служат для промежуточного хранения информации для регистров флэш-памяти. В регистр Тн записывается верхний предел температуры. В регистр TL — нижний. Эти регистры используют для проверки факта выхода величины измеренной температуры за границы установленного диапазона. Микроконтроллер способен быстро отыскать в сети Micro LAN все термодатчики, у которых не соблюдается это условие. Если не нужен механизм ограничения температуры, то регистры Тн и TL можно использовать как дополнительные ячейки энергонезависимой памяти и хранить в них любые данные. Например, туда можно записать код места положения конкретного датчика. Регистр конфигурации служит для переключения количества разрядов измерителя температуры.



Рисунок 12 - Структура памяти микросхемы DS18B20

Все три описанные выше регистра (байт 2, байт 3, байт 4) имеют механизм автоматического восстановления. При включении питания в них автоматически копируется информация из соответствующих регистров EEPROM. В регистр температуры после включения питания помешается код 0550Н (старший байт 05Н, младший байт 50Н), что соответствует температуре 85°С.

Оставшиеся три регистра блокнотной памяти (байт 5, байт 6 и байт 7) в микросхеме DSI8S20 не используются. Они зарезервированы для будущих ее модификаций. При чтении все три неиспользуемых регистра возвращают код OFFH (единицы во всех разрядах). Последний, восьмой регистр блокнотной памяти — это регистр генератора контрольной суммы.

Формат регистра температуры приведен на рисунке 13.

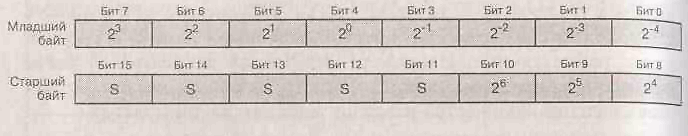


Рисунок 13 - Структура регистра температуры

После окончания процесса преобразования эти регистры содержат прямое значение величины измеренной температуры в двоичном виде. Регистр температуры — это два регистра блокнотной памяти. На рисунке 13 показан вес каждого разряда регистра. Биты с 11-го по 15-й (обозначенные буквой S) содержат одно и то же значение. Оно равно знаку записанного числи (0 — плюс, 1 — минус). Положительные значения температуры записываются в прямом коде, а отрицательные — в дополнительном (для того, чтобы перевести двоичное число в дополнительный код, нужно инвертировать его, а затем прибавить единицу).

Теперь рассмотрим формат регистров Тн и TL. На рисунке 14 он представлен в графическом виде. Как видно из рисунка, эти регистры имеют всего по восемь разрядов. Причем старший разряд - это знак числа. Поэтому верхний и нижний пределы температуры могут устанавливаться лишь с шагом в 1 градус. Для записи положительных и отрицательных чисел в регистрах Тн и TL также используются прямой и дополнительный коды.

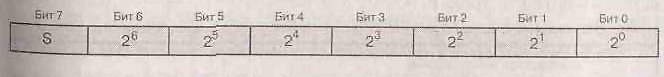


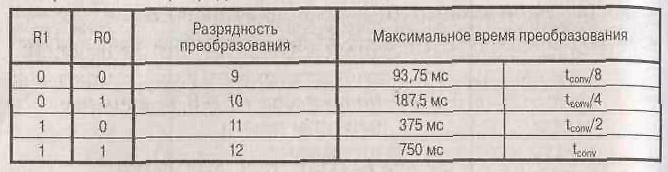
Рисунок 14 - Формат регистров Тн и TL.

Формат регистра конфигурации приведен на рисунке 15. Для изменения конфигурации используются только два разряда этого Регистра — бит 5 и бит 6. Значения остальных битов показаны на рисунке. В таблице 2 представлены все четыре режима, которые можно установить при помощи регистра конфигурации. Номер режима определяется разрядами RO и R1. При отключении лишних разрядов уменьшается точность измерения температуры, но одноименно уменьшается и время, необходимое для преобразования температуры в код. В таблице 2 для каждого из режимов работы приведено максимальное значение времени преобразования.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Бит 7 | Бит 6 | Бит 5 | Бит 4 | Бит 3 | Бит 2 | Бит 1 | Бит 0 |
| 0 | R1 | R0 | 1 | 1 | 1 | 1 | 1 |

Рисунок 15 - Формат регистра конфигурации

Таблица 2 - Выбор количества разрядов



На рисунке 16 показана схема включения микросхем DSI8B20 врежиме внешнего питания. Внешнее питание подается через вывод VDD. Если термодатчик находится на значительном удалении от микроконтроллера, то применен такой схемы включения не очень желательно, так как для питания датчика придется прокладывать еще один (третий) провод.

Второй вариант включения микросхем DSI8B20 изображен на рисунке 17. В такой схеме реализованы два режима питания. Для переключения режимов используется управляемый электронный ключ KI. Ключ управляется от микроконтроллера, для чего используется отдельная линия ввода/вывода. Такая схема позволяет переключать режимы питания программным путем. Основной режим работы для схемы, изображенной на рисунке 17 - это режим паразитного питания. В этом режиме ключ KI закрыт и напряжении на шине определяется резистором нагрузки R1, что позволяет передавать информацию по шине, используя 1-Wire протокол.

В нужный момент ключ KI открывается и на шину поступает полноценное питание от источника VPU. Питание поступает только на время выполнения одной из энергоемких команд. Пока ключ К1 открыт, информационный обмен по шине невозможен. Микроконтроллер выдерживает шину в таком состоянии необходимое время, а затем закрывает ключ К1. Шина возвращается в обычный режим работы и снова обретает возможность передачи данных. Для того, чтобы микросхема DSI8B20 правильно работала в режиме паразитного питания, нужно соединить между собой выводы VDD и GND и подключить оба этих вывода к общему проводу, как показано на рисунке.

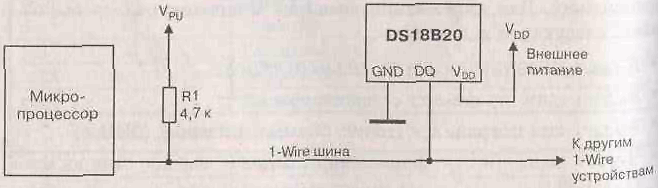


Рисунок 16 - Схема включения термодатчика в режиме внешнего питания

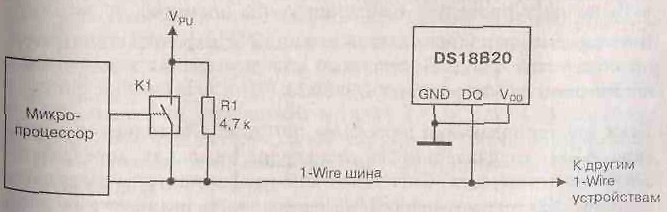


Рисунок 17 - Схема включения термодатчика в режиме паразитного питания

На рисунке 18показана схема включая вспомогательной цепи и цепи питания. Роль однопроводной шины выполняет линия Р1.1 микроконтроллера. Резистор R2 — это нагрузочный резистор шины. Рекомендованное значение номинала этого резистора 4,7 кОм. Электронный ключ для переключения режима питания собран на элементах VT1, R3, R4 и R5. Микроконтроллер управляет ключом при помощи линии Р1.0. Резистор R5 служит для ограничения тока базы транзистора VT1. Резистор R4 введен для надежного закрывания транзистора. Резистор R3 — страховочный. Он служит для ограничения тока при коротком замыкании в цепи датчика.

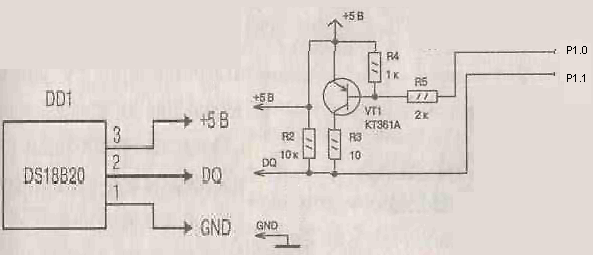


Рисунок 18 - Схема включения вспомогательной цепи и цепи питания

## 2.5 Система команд транспортного уровня микросхемы DSI8B20

Система команд транспортного уровня микросхемы DSI8B20 представлена в таблице 3:

Таблица 3 – Система команд транспортного уровня микросхемы DS18B20

|  |  |
| --- | --- |
| Код команды | Описание |
| 4EH | Запись блокнотной памяти |
| 0BEH | Чтение блокнотной памяти |
| 0B4H | Чтение режима питания |
| 0B8H | Чтение из EEPROM в блокнотную память |
| 48H | Копирование блокнотной памяти в EEPROM |
| 44H | Запуск процесса преобразования |

В данном проекте используется внешний режим питания микросхемы, поэтому команда «Чтение режима питания» (0B4H) не используется.

Команда «Запись блокнотной памяти» (Write Scratchpad). При выполнении этой команды микроконтроллер выдает на шину следующие сигналы:

-сигнал начального сброса;

-команду сетевого уровня «Пропуск ПЗУ»(0ССН);

-код операции Запись блокнотной памяти» (4EH);

-восемь байт для записи во все восемь регистров этой памяти.

Микросхема DS18B20 принимает все эти данные и записывает в регистры памяти.

Команда «Чтение блокнотной памяти» (Read Scratchpad). При выполнении этой команды микроконтроллер выдает на шину следующие сигналы:

-сигнал начального сброса;

-команду сетевого уровня «Пропуск ПЗУ»(0ССН);

-код операции «Чтение блокнотной памяти» (0ВEH);

Затем он считывает восемь байт данных из блокнотной памяти.

Команда «Чтение из EEPROM в блокнотную память» (Recall E2).

Команда служит для переноса информации из EEPROM в блокнотную память. Для выполнения этой команды микроконтроллер производит следующие действия:

-выдает на шину сигнал начального сброса;

-выдает команду сетевого уровня «Пропуск ПЗУ»(0ССН);

-выдает код операции «из EEPROM в блокнотную память» (0B8H).

Сразу после получения этой команды содержимое EEPROM копируется в блокнотную память. Эта команда выполняется автоматически каждый раз после включения питания.

Команда «Копирование блокнотной памяти в EEPROM» (Copy Scratchpad). При выполнении этой команды микроконтроллер выдает на шину следующие сигналы:

-сигнал начального сброса;

-команду сетевого уровня «Пропуск ПЗУ»(0ССН);

-код операции «Копирование блокнотной памяти в EEPROM» (48H);

-выполняет процедуру ожидания конца операции.

В результате выполнения этой операции содержимое блокнотной памяти копируется в EEPROM.

Команда «Запуск процесса преобразования» (Convert T). При выполнении этой команды микроконтроллер выдает на шину следующие сигналы:

-сигнал начального сброса;

-команду сетевого уровня «Пропуск ПЗУ»(0ССН);

-код операции «Запуск процесса преобразования» (44H);

-выполняет процедуру ожидания конца операции.

В результате выполнения этой команды измеренная температура преобразуется в код. Полученный код помещается в соответствующий регистр микросхемы DS18B20.

Длительность процедуры ожидания определяется сигналом готовности. Сигнал готовности формируется следующим образом. Как только микросхема термодатчика начинает выполнять одну из команд «Копирование блокнотной памяти в EEPROM» или «Запуск процесса преобразования», она «подсаживает» 1-Wire шину. Микроконтроллер проверяет уровень сигнала на шине. Обнаружив нулевой сигнал, он переходит в режим ожидания. Режим ожидания продолжается до тех пор, пока микросхема термодатчика не «отпустит» шину.

## 2.6 Выбор вентилятора

В кондиционерах применяются вентиляторы с крыльчаткой тангенциального типа [1], поток воздуха в которых поступает в крыльчатку с одной стороны, а выходит с другой, изменив направление своего движения.

Поперечное сечение такого вентилятора показано на рисунке 19. Срыв потока с кромок лопаток крыльчатки приводит к образованию ядра завихрения, служащего источником шума и гидравлических потерь устройства. Для обеспечения максимального акустического комфорта при работе кондиционера и максимальной дальнобойности воздушной струи фирмы-производители уделяют большое внимание отработке конфигурации направляющего аппарата.

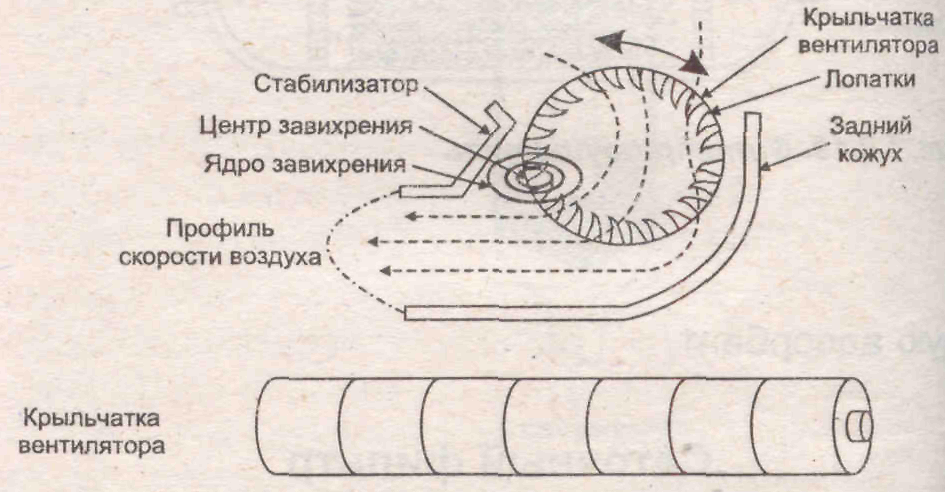


Рисунок 19 - Вентилятор кондиционера

В ряде моделей кондиционеров Daikin и Toshiba крыльчатка вентилятора имеет переменный шаг лопастей, что исключает возможность возникновения резонансных частот и связанного с ними шума.

Конструкторы кондиционеров стараются увеличить диаметр крыльчатки вентилятора, чтобы при том же расходе воздуха снизить его скорость. Чтобы избежать возникновения пульсаций воздушного потока на резонансных частотах, лопатки вентилятора располагают под разными углами к оси вращения. Характеристики выбранного вентилятора приведены в таблице 4.

Таблица 4 - Характеристики вентилятора

|  |  |  |
| --- | --- | --- |
| Тип | | Центробежный |
| Диаметр / длина крыльчатки, мм | | 70/598 |
| Модель двигателя | | YDK10-2A |
| Количество полюсов | | 2 |
| Максимальная скорость вращения, об/мин | | 1950 |
| Номинальная выходная мощность, Вт | | 10 |
| Сопротивление обмоток, Ом (при 20°С).  Цвет изоляции выводов:  «Белый – серый»  «Белый – розовый» | | 410±10%  301±10% |
| Устройство безопасности | Тип | Внутренний термопредохранитель |
| Температура срабатыания, °С | 145±8 |
| Управляющий конденсатор | Емкость, мкФ | 1,0 |
| Мощность, ВА | 450 |

## 

## 2.7 Выбор компрессора

В бытовых и полупромышленных кондиционерах в настоящее время используются три основных типа компрессоров — ротационный, спиральный и поршневой, причем на долю ротационных компрессоров приходится около 90%. Так из 23 млн. компрессоров, проданных по всему миру в 1995 г. для применения в климатических системах, более 20 млн. были ротационного типа. В климатических системах большой мощности (от 160 до 3500 кВт) применяются винтовые компрессоры. Ротационные компрессоры (рисунок 20) осуществляют всасывание и сжатие газа с помощью вращающегося на валу ротора. За счет вращательного движения рабочих органов в компрессорах этого типа (также как в спиральных и винтовых) существенно снижены пульсации давления и пусковые токи.

Ротационные компрессоры [3] производятся в двух вариантах: со стационарными и вращающимися пластинами. Рабочий цикл компрессора со стационарной пластиной показан на рисунке 21. Ротор эксцентрично закреплен на валу компрессора. При вращении вала эксцентрик обкатывается по внутренней поверхности цилиндра, сжимая перед собой очередную порцию хладагента. Пластина разделяет области высокого и низкого давления.

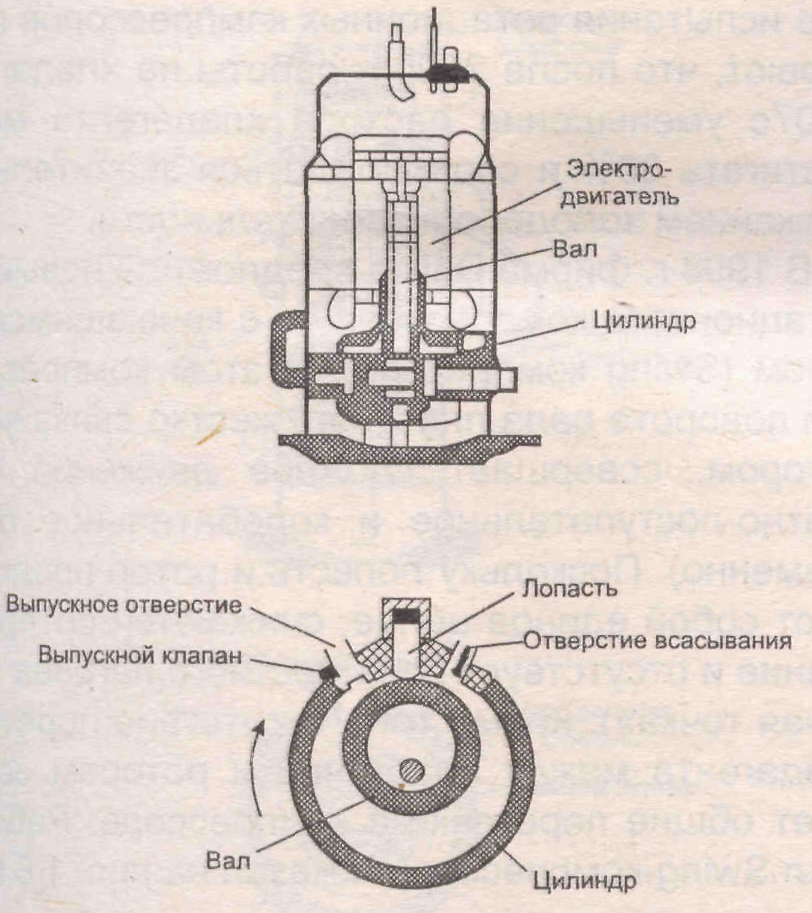


Рисунок 20 - Устройство ротационного компрессора

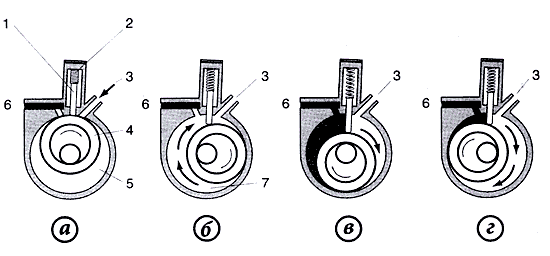


Рисунок 21 - Рабочий цикл ротационного компрессора со стационарными пластинами

а — рабочий объем цилиндра заполнен хладагентом,

б — начало сжатия хладагента (слева от ротора) и всасывание новой его порции (справа),

в — продолжение сжатия и всасывания,

г — завершение сжатия и заполнение рабочего объема цилиндра новой порцией хладагента.

1 — пластина,

2 — пружина,

3 — отверстие всасывания,

4 — ротор,

5 — рабочий объем цилиндра,

6 — выпускной клапан.

Ряд фирм-производителей Panasonic, Sanyo и др.) применяют в своих ротационные компрессоры с двумя роторами (рисунок 22) [1]. На валу компрессора эксцентрично вращаются два ротора, каждый из которых осуществляет сжатие хладагента в своем цилиндре. Эксцентрики расположены на валу противофазно (рисунок 23), благодаря чему уменьшается суммарная вибрация при их совместной работе. Цилиндры двухроторного компрессора соединены между собой перепускной трубкой (байпасом) с управляющим клапаном, что позволяет эффективно регулировать производительность при использовании компрессоров данного типа.

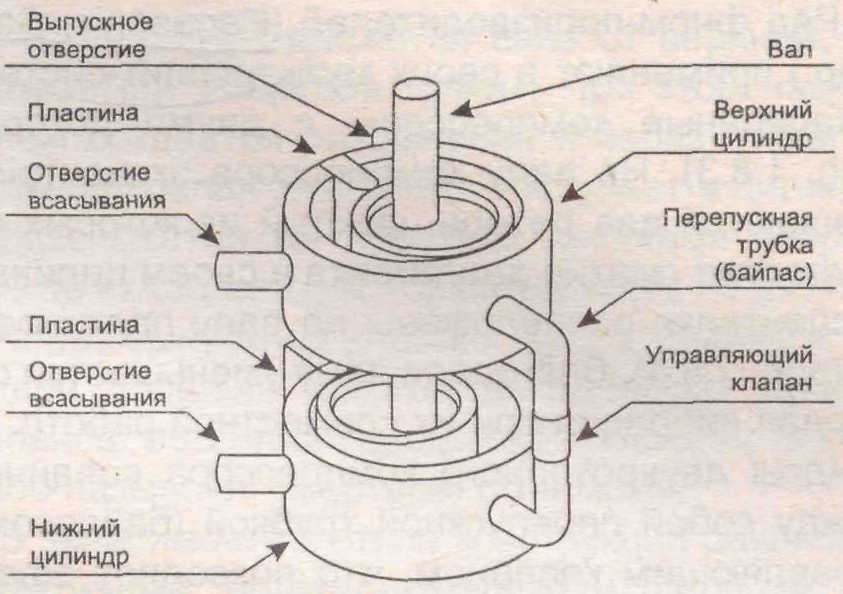


Рисунок 22 - Ротационный компрессор с двумя роторами

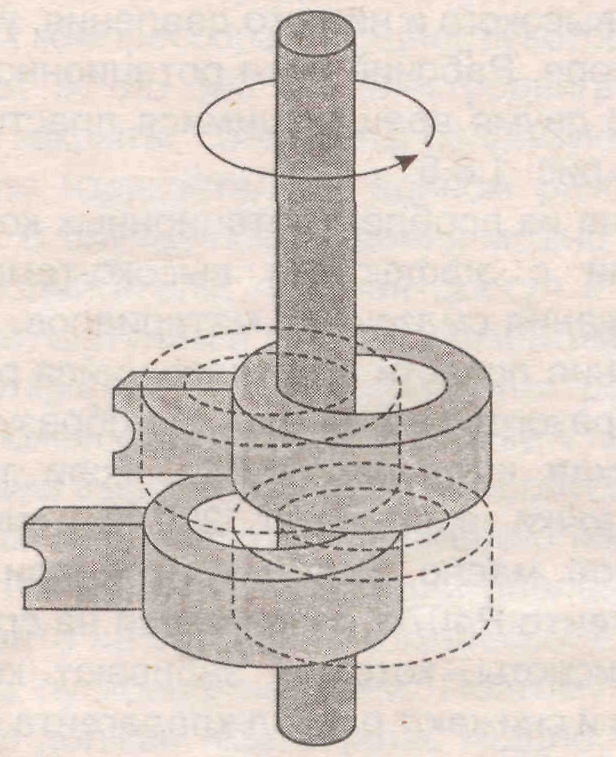


Рисунок 23 - Противофазное расположение роторов на валу двухроторного компрессора

В компрессорах с вращающимися пластинами эти пластины (две или более), разделяющие области высокого и низкого давления, установлены на роторе.

Рабочий цикл ротационного компрессора с двумя вращающимися пластинами показан на рисунке 24.

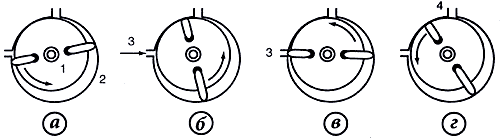


Рисунок 24 - Рабочий цикл ротационного компрессора с вращающимися пластинами

а — рабочий объем цилиндра заполнен хладагентом,

б — начало сжатия хладагента и всасывание новой его порции,

в — продолжение сжатия и всасывания,

г — завершение сжатия и заполнение рабочего объема цилиндра новой порцией хладагента.

1 — ротор,

2 — цилиндр,

3 — отверстие всасывания,

4 — выпускное отверстие

Одна из проблем ротационных компрессоров связана с эффектами высоко-температурного разложения смазочных материалов. В результате трения лопасти о вращающийся ротор происходит разогрев ее кромки, где образуется так называемая «горячая точка». Если температура этой точки превышает 200 °С, синтетическое эфирное масло, используемое при работе на хладагенте R407c, разлагается на спирт и жировые кислоты, которые забивают капиллярные трубки и снижают расход хладагента. Лабораторные испытания ротационных компрессоров показывают, что после 2000 ч работы на хладагенте R407c уменьшение расхода хладагента может достигать 30% и сопровождаться значительным снижением холодопроизводительности.

В 1998 г. фирма Daikin предложила новый вид ротационного компрессора — с качающимся ротором (Swing компрессор). В этом компрессоре при повороте вала пластина, жестко связанная с ротором, совершает сложное движение (возвратно-поступательное и колебательное одновременно). Поскольку лопасть и ротор представляют собой единое целое, снижаются потери на трение и отсутствует зона местного нагрева («горячая точка»). Кроме того, отсутствие перетечек хладагента между пластиной и ротором сокращает общие перетечки в компрессоре. Рабочий цикл Swing-компрессора показан на рисунке 25.



Рисунок 25 - Рабочий цикл Swing-компрессора.

фаза 1 — рабочий объем цилиндра заполнен хладагентом

фаза 2 — начало сжатия хладагента и всасывание новой его порции

фаза 3 — сжатие и всасывание продолжается

фаза 4 — завершение сжатия и заполнение рабочего объема цилиндра новой порцией хладагента

В климатических системах малой и средней мощности (от 5 до 40 Вт) используются также спиральные компрессоры [1] (компрессоры Scroll). Компрессор (рисунок 26) состоит из двух стальных спиралей, расширяющихся от центра к периферии цилиндра и вставленных одна в другую. Одна из спиралей закреплена неподвижно, вокруг нее вращается подвижная спираль. Профиль спиралей образован эквивалентной кривой. Подвижная спираль установлена на эксцентрике и при вращении ее внешняя поверхность как бы катится по внутренней поверхности неподвижной спирали. Благодаря этому точка контакта спиралей постепенно перемещается от периферии к центру, сжимая перед собой пары хладагента и вытесняя их в центральное отверстие в верхнее крышке цилиндра. Так как точек контакта несколько (они расположены на каждом витке подвижной спирали), то происходит более плавное сжатие паров, уменьшается нагрузка на электродвигатель, особенно в момент пуска.

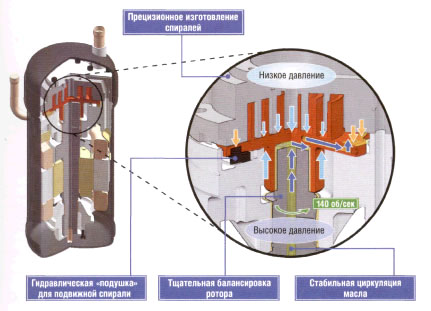


Рисунок 26 - Спиральный компрессор (компрессор Scroll).

В технологическом плане компрессор Scroll более сложен, поскольку необходимо обеспечить герметичность по торцам спиралей и очень точное прилегание профилей спиралей. Поэтому компрессоры данного типа пока нашли ограниченное применение.

В поршневом компрессоре (рисунок 27) сжатие газа происходит при возвратно-поступательном движении поршня в цилиндре/

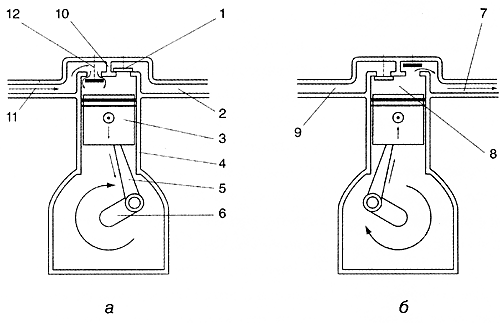


Рисунок 27 - Поршневой компрессор

а — фаза всасывания хладагента,

б — фаза сжатия и выпуска хладагента высокого давления.

1 — выпускной клапан, 2 — линия нагнетания хладагента,

3 — поршень, 4 — цилиндр,

5 — шатун,

6 — коленчатый вал,

7 — головка клапанов,

8 — линия всасывания хладагента,

9 —впускной клапан

В фазе всасывания (а) поршень движется вниз от верхней, так называемой «мертвой точки». При этом над поршнем создается разрежение и через открытый впускной клапан хладагент поступает в цилиндр. В фазе сжатия (б) поршень движется вверх и сжимает хладагент, который выходит из цилиндра через выпускной клапан. При движении в цилиндре поршень никогда не касается головки клапанов, оставляя свободное пространство, которое называют «мертвым объемом».

В зависимости от типа конструкции различают герметичные, полугерметичные и открытые поршневые компрессоры. В герметичном компрессоре электродвигатель и компрессор находятся в едином герметичном корпусе. Такие компрессоры, мощностью 1,7...35 кВт применяются в холодильных машинах малой и средней мощности. В полугерметичных компрессорах, мощность которых варьируется от 30 до 300 Вт, электродвигатель и компрессор закрыты, соединены напрямую и расположены по горизонтали в едином разборном контейнере. В случае повреждения можно извлекать электродвигатель, получая доступ к клапанам, поршню, шатунам и другим элементам конструкции. В открытых компрессорах электродвигатель расположен снаружи (вал с соответствующими сальниками выведен за пределы корпуса).

Основным недостатком поршневого компрессора является наличие пульсаций давления паров хладагента на выходе из компрессора, а также большие пусковые нагрузки. Поэтому электродвигатель должен иметь запас мощности для пуска компрессора и иметь акустическую защиту для снижения уровня шума.

Количество запусков компрессора является наиболее критичным для его срока службы. Именно на режиме запуска происходит наибольшее количество отказов, поэтому приходится ограничивать время между повторными пусками компрессора (как правило, не менее 6 мин), и время между остановом компрессора и его повторным пуском (2...4 мин). Характеристики выбранного компрессора приведены в таблице 5.

Таблица 5 – характеристики компрессора.

|  |  |  |  |
| --- | --- | --- | --- |
| Модель | | | PH180X1C-4DT2 |
| Номинальная выходная мощность компрессора, Вт | | | 800 |
| Объем смазочного масла (SUNISO 4GSD), см3 | | | 400 |
| Ток при заторможенном роторе компрессора, А  -220 В  -240 В | | | 23,6  25,5 |
| Сопротивление обмотки компрессора, Ом (при 20°С) | | | C-R: 3,13 |
| C-S: 4,46 |
| Предохранительное устройство компрессора | Тип | | Внутреннее |
| Модель | | UPQE0591-T51 |
| Рабочая температура,°С | Открыто | 150±5 |
| Закрыто | 90±10 |
| Ток управления, А (внутренняя температура 25°С) | | 25 (переключение через 3-10 с) |
| Управляющий конденсатор | | | 25 |
| 450 |

## 2.8 Выбор микроконтроллера

В настоящее время среди всех 8-разрядных микроконтроллеров - семейство MCS-51 является несомненным чемпионом по количеству разновидностей и количеству компаний, выпускающих его модификации. Оно получило свое название от первого представителя этого семейства - микроконтроллера 8051, выпущенного в 1980 году на базе технологии HMOS. Удачный набор периферийных устройств, возможность гибкого выбора внешней или внутренней программной памяти и приемлемая цена обеспечили этому микроконтроллеру успех на рынке.

Важную роль в достижении такой высокой популярности семейства 8051 сыграла открытая политика фирмы Intel, родоначальницы архитектуры, направленная на широкое распространение лицензий на ядро 8051 среди большого количества ведущих полупроводниковых компаний мира.

В результате на сегодняшний день существует более 200 модификаций микроконтроллеров семейства 8051, выпускаемых почти 20-ю компаниями. Эти модификации включают в себя кристаллы с широчайшим спектром периферии: от простых 20-выводных устройств с одним таймером и 1К программной памяти до сложнейших 100-выводных кристаллов с 10-разрядными АЦП, массивами таймеров-счетчиков, аппаратными 16-разрядными умножителями и 64К программной памяти на кристалле. Каждый год появляются все новые варианты представителей этого семейства. Основными направлениями развития являются: увеличение быстродействия (повышение тактовой частоты и переработка архитектуры), снижение напряжения питания и потребления, увеличение объема ОЗУ и FLASH памяти на кристалле с возможностью внутрисхемного программирования, введение в состав периферии микроконтроллера сложных устройств типа системы управления приводами, CAN и USB интерфейсов и т.п.

Все микроконтроллеры из семейства MCS-51 имеют общую систему команд [4]. Наличие дополнительного оборудования влияет только на количество регистров специального назначения.

Основными производителями клонов 51-го семейства в мире являются фирмы Philips, Siemens, Intel, Atmel, Dallas, Temic, Oki, AMD, MHS, Gold Star, Winbond, Silicon Systems и ряд других.

В рамках СССР производство микроконтроллера 8051 осуществлялось в Киеве, Воронеже (1816ВЕ31/51, 1830ВЕ31/51), Минске (1834ВЕ31) .

Для данной задачи был выбран микроконтроллер AT89C51 фирмы Atmel [5]. Несмотря на то, что фирма Atmel уже давно делает упор на новое поколение микроконтроллеров (серии AVR), микроконтроллер AT89C51 тоже довольно широко применяется. И не случайно, так как эта микросхема имеет еще достаточно большой потенциал. Параметры микросхемы позволяют создавать широкий спектр современных электронных устройств, находящих свое применение в самых различных областях микропроцессорной техники. Главным преимуществом выбора именно этой микросхемы является ее широкая доступность и приемлемая цена.

## 3 Функциональная схема

Функциональная схема кондиционера изображена на рисунке 28.



Рисунок 28 - Функциональная схема кондиционера

## 4 Алгоритм работы кондиционера

Алгоритм работы кондиционера представлен на рисунке 29.



Рисунок 29 - Алгоритм работы кондиционера

## 5 Разработка программы

Программная реализация алгоритма работы кондиционера на языке ассемблер для микроконтроллера AT89C51 представлена в приложении 1.

В основной программе сначала определяются константы: коды банков памяти, счетчик задержки, буфер для приема данных из термодатчика, буфер для хранения заданной температуры помещения Туст, счетчик рабочих циклов кондиционера для определения его режима работы. Дальше присваивается начальное значение Туст и вызывается подпрограмма опроса клавиатуры с последующим определением нажатых клавиш. При последующих опросах клавиатуры, если клавиша нажата, то происходит переход к метке in1. Затем следует изменение Туст (увеличение или уменьшение на 1) и вызов ряда процедур для вывода Туст на ЖКИ №1. После вывода Туст запускается преобразование температуры помещения в термодатчике (Тпм) и чтение ее (с сохранением в буфере bufLAN). Далее Тпм выводится на ЖКИ №2, после чего происходит обработка Тпм (а именно запись Тпм в один байт) и сравнение Тпм<=Туст. Обнуляется счетчик циклов кондиционера и происходит переход к нужному режиму работы кондиционера. В конце основной программы происходит переход на один из режимов работы кондиционера или на метку in1 (где происходит анализ нажатых клавиш) по результатам выполнения подпрограмм reoh (режим охлаждения) и reob (режим обогрева).

Подпрограмма inkl ввдда с клавиатуры опрашивает две клавишы S1 и S2 и выводит в аккумулятор результат (0 бит аккумулятора – отвечает за кнопку S2, 1 бит аккумулятора – за кнопку S1).

Далее в приложении 1 представлен ряд подпрограмм для инициализации и вывода символов и строк на оба ЖКИ дисплея. Таблица символов ts необходима для удобства работы с кодами символов ЖКИ дисплеев (в ней все коды представлены по порядку). Подпрограмма вывода сообщения об ошибке будет использоваться при работе с 1-Wire шиной.

Подпрограмма вывода Туст на ЖКИ №1 записывает из bufLAN2 значение Туст в аккумулятор и отделяет единицы температуры от десятков, после чего выводит значение Туст на ЖКИ №1 со знаком «градус Цельсия».

Подпрограмма zader вызывается подпрограммами reoh и reob. Она с интервалом в 100мс опрашивает клавиатуру и выводит Тпм на ЖКИ №2. При нажатии клавиш подпрограмма записывает в аккумулятор значение 1 и завершает свою работу. По состоянию аккумулятора в подпрограмме reoh либо происходит выход из подпрограммы с целью анализа клавиатуры в основной программе (при значении 1 аккумулятора), либо продолжается ее выполнение (при значении 0 аккумулятора). На выполнение подпрограммы тратится примерно 3мин.

Подпрограмма sttmp предназначена для передачи по шине команды «Запуск процесса преобразования». Для этого, в качестве команды сетевого уровня, используется команда «Пропустить ПЗУ». Датчик переходит в режим преобразования. Подпрограмма использует банк номер 2. Сначала происходит сохранение в стеке регистра psw и переключение банков.

Затем вызов процедуры начального сброса. Снова контролируется код ошибки. Затем формируется команда сетевого уровня «Пропустить ПЗУ». Команда транспортного уровня («Запуск процесса преобразования») формируется с помощью кода 044Н. Получив эту команду, термодатчик переходит в режим преобразования температуры, а программа переходит в режим ожидания. Управление передается по метке stm4.

Процедура ожидания занимает всего две строки. Эти две строки реализуют цикл ожидания сигнала готовности. Сигнал готовности формирует микросхема термодатчика в момент, когда он закончит процесс преобразования температуры в код. Далее происходит чтение бита информации и его проверка. Если прочитанный бит равен нулю, то управление передается по метке stm4 и цикл чтения продолжается. Процесс чтения и проверки продолжается до тех пор, пока очередной раз не будет получена единица.

Подпрограмма содержат команды обработки ошибок. В случае возникновения ошибки управление передается к метке stm6. Здесь вызывается подпрограмма prErr, которая выводит сообщение об ошибке на экран. Затем управление передается по метке stmf на начало подпрограммы, и процесс передачи команды запуска на преобразование повторяется.

Подпрограмма чтения температуры называется rdtmp. Подпрограмма производит считывание температуры с датчика. Она возвращает значение измеренной температуры в виде двух байт, записанных в две старшие ячейки буфера bufLAN. Младший байт регистра температуры записывается по адресу bufLAN, а старший — по адресу bufLAN+1.

Начинается подпрограмма rdtmp с команды сохранения регистра psw. Далее включается банк номер два.

После этого подпрограмма может начинать выполнение своей главной задачи: чтение регистров блокнотной памяти термодатчика. Начинается этот процесс с вызова процедуры начального сброса, проверяется признак ошибки. Далее выдается команда «Пропустить ПЗУ».

При этом датчик перейдет на транспортный уровень. Далее, программа перейдет к формированию команды «Чтение блокнотной памяти». Сначала в линию выдается код операции (0ВЕН). Затем начинается процесс чтения регистров термодатчика. Читаются только два первых регистра. Именно они и содержат значение температуры. Протокол 1-Wire допускает такой режим работы. Первый байт значения температуры помещается в ячейку памяти с адресом bufLAN. Второй байт и помещается в ячейку с адресом bufLAN+1. На этом подпрограмма rdtmp завершает свою работу. Перед выходом из подпрограммы восстанавливается содержимое регистра psw.

Подпрограмма resLAN предназначена для формирования импульса начального сброса. Эта же подпрограмма производит проверку наличия сигнала присутствия на линии. Подпрограмма способна обнаруживать два вида ошибок: обрыв линии и короткое замыкание. В случае возникновения ошибки подпрограмма возвращает код ошибки через аккумулятор.

Начинается подпрограмма resLAN с формирования импульса начального сброса. Процесс формирования начинается с того, что программа «подсаживает» линию. Затем формируется задержка длительностью 700 мкс. После чего линия «отпускается». Для формирования задержки используется метод пустого цикла. Сначала параметру цикла присваивается начальное значение. В данном случае это значение равняется 100. Именно столько раз в цикле вызывается подпрограмма Dely. После формирования импульса начального сброса программа приступает к проверке сигнала присутствия на линии. Для этого она формирует задержку в 65 мкс, а затем проверяет значение сигнала на линии. Если нет обрыва на линии и сигнал присутствия сформирован нормально, то именно в этот момент времени уровень сигнала на линии должен быть равен нулю. Если уровень сигнала на линии р1.1 равен нулю, то это значит, что ошибки не обнаружено. Управление передается по метке rst3. Если уровень сигнала на линии равен единице, то в аккумулятор помещается код ошибки «обрыв на линии», а затем подпрограмма досрочно завершается.

Если все же произошел переход к метке rst3 (ошибка не обнаружена), подпрограмма формирует задержку в 500 мкс и снова проверяет уровень сигнала на линии. Задержка в 500 мкс необходима для правильного завершения процесса начального сброса. По истечении этого промежутка времени сигнал присутствия должен уже закончиться и уровень сигнала на линии должен быть равен единице. Если это не так, то это значит, что линия просто закорочена. В таком случае в аккумулятор записывается код ошибки 2 («короткое замыкание») и программа досрочно завершается. Если короткого замыкания не обнаружено, то управление передается по метке rst3 и подпрограмма завершается нормальным образом. В этом случае, перед завершением подпрограммы в аккумулятор помещается нулевой код. Ноль— это код отсутствия ошибки.

Подпрограмма mrslot (чтение слота) служит для чтения одного бита информации из ведомого устройства. Подпрограмма возвращает прочитанный бит через CY (признак переноса). Сначала подпрограмма формирует синхроимпульс. Для этого она «подсаживает» линию, формирует задержку в 1 мкс, а затем линию отпускает. После окончания синхроимпульса формируется задержка в 13 мкс и считывается уровень сигнала на линии. Прочитанный бит помещается в CY. Это и есть результат чтения слота. Однако на этом выполнение подпрограммы не завершается.

Прежде чем выйти из подпрограммы, необходимо закончить формирование длительности слота. Общая длительность слота должна лежать в пределах 60... 120 мкс. Для попадания времени выполнения нашей подпрограммы в этот диапазон формируется задержка в 60 мкс. И лишь после этого подпрограмма завершается.

Программа mwlow записывает в слот нулевое значение. Действие подпрограммы сводится к формированию в пределах слота отрицательного импульса длительностью 80 мкс.

Подпрограмма mwhi чуть сложнее. Она выполняет запись в слот единичного значения. Действие подпрограммы начинается с формирования синхроимпульса длительностью 5 мкс. Для формирования такой длительности производится однократное обращение к подпрограмме Delay. После завершения синхроимпульса формируется задержка 75 мкс. Эта задержка доводит длительность слота до стандартной величины.

Подпрограмма Dely служит для формирования временных интервалов. Как видно из текста подпрограммы, она представляет собой один оператор nор и завершается командой выхода из подпрограммы. Такая подпрограмма обеспечивает задержку, примерно равную 5 мкс. Подпрограмму Delay удобно использовать при формировании относительно больших задержек по времени, включая вызов этой подпрограммы в тело пустого цикла. Команда пор не выполняет абсолютно никаких действий. В то же время, ее выполнение занимает один машинный цикл микроконтроллера. При тактовой частоте, равной 12 МГц, длительность машинного цикла будет одна микросекунда (12/12 = 1 мкс). В микросхеме АТ89С2051 частота тактового генератора делится на 12. Для формирования небольших временных интервалов достаточно нескольких операторов пор, поставленных подряд друг за другом. Для более длительных интервалов применяется пустой цикл. Цикл просто выполняет несколько пустых операторов определенное число раз. В представленных здесь подпрограммах широко используются оба способа формирования временных интервалов. В качестве счетчика цикла используется специальная ячейка памяти, обозначенная в программе, как LoopCnt.

Подпрограмма wr8LAN выводит содержимое байта, полученного через аккумулятор, бит за битом в режиме записи. Именно поэтому описываемая подпрограмма представляет собой, по сути, цикл для передачи битов. Регистр r1 используется в качестве параметра цикла. Начальное значение r1 равно восьми (по числу битов в байте). Выполнение цикла начинается с команды извлечения очередного бита. Для этого содержимое аккумулятора сдвигается вправо при помощи команды rrс. Действие этой команды приводит к тому, что очередной бит оказывается в ячейке признака переноса CY.

Затем программа оценивает значение этого бита. Если он равен единице, вызывается подпрограмма записи единицы в слот mwhi. В противном случае вызывается подпрограмма mwlow для записи в слот нуля. После выполнения одной из этих программ управление передается по метке wb3. В этой строке находится команда организации цикла. Она передает управление на начало цикла, если это не последний бит, и завершает цикл, если бит последний.

Подпрограмма wr1LAN предназначена для передачи по шине одного бита в режиме записи. Эта подпрограмма очень похожа на wr8LAN. Отличие только в отсутствии цикла. Бит информации, предназначенный для передачи по шине 1-Wire, передается в подпрограмму через ячейку CY. В зависимости от значения этого бита выполняется либо подпрограмма mwhi, либо mwlow.

Подпрограмма rd8LAN предназначена для чтения одного байта по шине 1-Wire. Программа возвращает прочитанный байт в аккумулятор. Основная часть подпрограммы — это цикл ввода битов. В качестве параметра цикла используется регистр r1. Первая команда, выполняемая в теле цикла — вызов подпрограммы чтения слота. Она помещает прочитанный бит в ячейку CY. Затем бит помещается в аккумулятор методом сдвига. Сдвиг производится при помощи команды rrс. После каждого такого сдвига в аккумулятор «вдвигается» очередной прочитанный бит. После восьми циклов сдвига аккумулятор будет содержать полноценный прочитанный байт. Оператор djnz служит для организации всего этого цикла.

Подпрограмма rdlLAN предназначена для чтения одного бита из линии 1-Wire. Она вызывает подпрограмму чтения слота mrslot.

В ОЗУ микроконтроллера температура записана в двоичном виде. Перед нами стоит задача: перевести это значение из двоичной системы в десятичную. И лишь затем вывести его на экран.

Для двоично-десятичного преобразования удобно использовать команду целочисленного деления div. Для того, чтобы осуществить двоично-десятичное преобразование, нужно выполнить целочисленное деление исходного числа на 10. После такой операции аккумулятор будет содержать частное, а регистр b — остаток от деления. Частное будет соответствовать количеству десятков, а остаток — количеству единиц в десятичном представлении числа.

Затем можно просто вывести на экран два этих числа. Сначала содержимое аккумулятора, а затем содержимое регистра b. В результате, на экране мы получим изображение целой части значения температуры в десятичном виде. Такой простой метод преобразования будет правильно работать только в том случае, если целая часть числа не превышает значения 99. При большем значении температуры в старшем разряде полученного десятичного числа будут появляться буквы: частное от деления будет больше 10. Однако для измерения температуры помещения диапазона (0...60)°С вполне достаточно. Для реализации описанного алгоритма разработана подпрограмма prtmp. Подпрограмма prtmp читает два байта температуры из буфера bufLAN и выводит значение целой части температуры на экран в десятичном виде.

Подпрограмма reoh обеспечивает режим охлаждения работы кондиционера. Она дискретно (с интервалом в 3мин.) вклычает и выключает компрессор и вентилятор и анализирует разницу Туст и Тпм. Если Тпм меньше Туст на 2 градуса в течении двух циклов работы кондиционера в режиме охлаждения, то кондиционер переходит в режим обогрева (подпрограмма reoh с помощью аккумулятора переходит к метке in4 основной программы). Если нет, то кондиционер остается в режиме охлаждения (переходит к метке in5). Подпрограмма reoh так же вызывает подпрограмму zader ( если клавиша нажата, то переход с помощью аккумулятора к метке in1).

Подпрограмма reob обеспечивает режим обогрева работы кондиционера. Она не представлена в приложении 1, но работает аналогично подпрограмме reoh (включает и выключает не компрессор, а нагревательный элемент).

# ЗАКЛЮЧЕНИЕ

В проекте были разработаны структурная и функциональная схемы кондиционера с микропроцессорным управлением, в которых были учтены все необходимые элементы. Также был разработан алгоритм и большая часть программы.

Разработанная система управления позволяет поддерживать заданную температуру в помещении, удовлетворяет основным требованиям: обеспечивает экономичность, низкую стоимость, простоту в использовании и минимальные затраты на эксплуатацию.

В процессе выполнения работы был выбран термодатчик с 1-Wire интерфейсом, который оптимально подходит для решения задач курсового проекта. Была разобрана работа 1-Wire интерфейса на разных уровнях программной реализации. Выбраны типичные исполнительные устройства кондиционера, такие, как компрессор, вентилятор, нагревательный элемент по таким критериям, как низкая стоимость, простоту в использовании и минимальные затраты на эксплуатацию. Был выбран микроконтроллер семейства МК51 по тем же критериям.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. В.В.Коляда. Кондиционеры – М.: СОЛОН-Пресс, 2002 – с. 233.
2. А.В.Белов. Конструирование устройств на микроконтроллерах – СПб.: Наука и техника, 2005 – с. 255.
3. www.inrost.ru - научно-техническая библиотека сайта фирмы ИНРОСТ.
4. А.В.Логинов. Программирование микро-ЭВМ семейства МК51: Учеб. Пособие – СПб.: Балт. гос. тех. ун-т, 1996 – 72с.
5. www.atmel.ru - официальный сайт корпорации Atmel.

# ПРИЛОЖЕНИЕ

## Приложение 1 ;основная программа

bank0 EQU 00000000B ;коды банков памяти

bank1 EQU 00001000B

bank1 EQU 00010000B

bank2 EQU 00110000B

DSEG

ORG 30H

LoopCnt: DS 1 ;счетчик задержки

bufLAN: DS 8 ;буфер для приема данных из MicroLAN

bufLAN2: DS 8 ;буфер для Туст

bufLAN3: DS 1 ;счетчик рабочих циклов кондиционера

ORG 60H ;начало стека

stack: DS 20H ;глубина стека

main: mov bufLAN2, #18H ;начальное значение Туст=24 градуса Цельсия

call inkl

jz in3 ;если клавиши не нажаты переход

in1: cjne a, #11B, in2 ;если обе клавиши нажаты переход

inc bufLAN ;Туст увеличить на 1

jmp in3

in2: dec bufLAN2

in3: call eclr1 ;вывод Туст

mov a, #2

call ecur1

call prtmu ;Туст на ЖКИ №1

call eclr2

mov a, #2

call ecur2

call sttmp ;запуск преобразования Тпм

call rdtmp ;чтение Тпм

call prtmp ;вывод Тпм на ЖКИ №2

mov a, bufLAN+1 ;записьТпм в аккумулятор

swap a

anl a, #0F0H

mov R5, a

mov a, bufLAN

swap a

anl a, #0FH

add a, R5 ;в аккумуляторе байт Тпм

mov a, #LE(a, bufLAN2) ;сравнение Тпм<=Туст

mov bufLAN3, #0 ;обнуление счетчика циклов работы кондиционера

cjne a, #0FFFFH, in5 ;обработкаусловия Тпм<=Туст

in4: call reob

jmp in6

in5: call reoh

in6: mov b, a

mov a, EQ(b, #1)

cjne a, #0FFFFH, in7

jmp in1

in7: mov a, EQ(b, #4)

cjne a, #0FFFFH, in5

jmp in4

;Подпрограмма ввода с клавиатуры (0 бит аккумулятора – за S2, 1 бит – за S1)

inkl: setb P0.0 ;установка 1 на линиях клавиатуры

setb P0.1

mov a, P0 ;считывание порта p0

anl a, #00000011B ;обнуление лишних разрядов

xrl a, #00000011B ;инвертирование

ret

;подпрограмма сброса ЖКИ №1

eres1: mov P2, #0FH ;засылаем адрес регистра блокировки

setb P2.4

clr P2.4

mov P2, #01000001B ;записываем код снятия блокировки

setb P2.4

clr P2.4

mov P2, #0 ;устанавливаем адрес на первый разряд

setb P2.4

clr P2.4

ret

; подпрограмма сброса ЖКИ №2

eres2: mov P3, #0FH

setb P3.4

clr P3.4

mov p3, #01000001B

setb P3.4

clr P3.4

mov P3, #0

setb P3.4

clr P3.4

ret

;подпрограмма очистки ЖКИ №1

eclr1: call eres1 ;сброс экрана

mov P2, #01000000B ;установка управляющего кода

mov R0, #20 ;инициализация счетчика разрядов

c11: set P2.4 ;импульс записи

clr P2.4

djnz R0, c11 ;оператор цикла

ret

; подпрограмма очистки ЖКИ №2

eclr2: call eres2

mov P3, #01000000B

mov R0, #20

c21: set P3.4

clr P3.4

djnz R0, c21

ret

;подпрограмма установки курсора для ЖКИ №1 по аккумулятору

ecur1: anl a, #0FH ;отделяем биты адреса

mov P2, a ;устанавливаем адрес на первый разряд

setb P2.4

clr P2.4

ret

; подпрограмма установки курсора для ЖКИ №2 по аккумулятору

ecur2: anl a, #0FH

mov P3,a

setb P3.4

clr P3.4

ret

;таблица символов

ORG 0600H

ts: DB 0EEH ;0

DB 060H ;1

DB 02FH ;2

DB 06DH ;3

DB 0E1H ;4

DB 0CDH ;5

DB 0CFH ;6

DB 068H ;7

DB 0EFH ;8

DB 0EDH ;9

DB 0EBH ;A

DB 0C7H ;b

DB 08EH ;C

DB 067H ;d

DB 08FH ;E

DB 08BH ;F

DB 000H ;

DB 0A9H ;°

DB 001H ;-

DB 003H ;r

DB 047H ;o

;подпрограмма вывода символа на ЖКИ №1 через аккумулятор

prch1: push DPL ;сохранение регистров в стеке

push DPH

push b

mov DPTR, #ts ;начало таблицы символов

mov c, acc.7 ;сохраняем значение точки

clr acc.7 ;сбрасываем значение бита

movc a, @a+DPTR ;получаем выводимый код

mov acc.4, c ;восстанавливаем значение точки

mov b, a ;сохраняем его в b

anl a, #0FH ;отделяем младший полубайт

mov P2, a ;выводим его на индикатор

setb P2.6 ;вкл. Режим вывода данных

setb P2.4 ;импульс записи

clr P2.4

mov a, b ;восстанавливаем выводимый код

swap a

anl a, #0FH ;отделяем старший полубайт

mov p2, a ;выводим его на индикатор

setb p2.6 ;вкл. Режим вывода данных

setb P2.4 ;импульс записи

clr P2.4

pop b

pop DPH ;восстановление регистров

pop DPL

ret

;подпрограмма вывода символа на ЖКИ №2 через аккумулятор

prch2: push DPL ;сохранение регистров в стеке

push DPH

push b

mov DPTR, #ts ;начало таблицы символов

mov c, acc.7 ;сохраняем значение точки

clr acc.7 ;сбрасываем значение бита

movc a, @a+DPTR ;получаем выводимый код

mov acc.4, c ;восстанавливаем значение точки

mov b, a ;сохраняем его в b

anl a, #0FH ;отделяем младший полубайт

mov P3, a ;выводим его на индикатор

setb P3.6 ;вкл. Режим вывода данных

setb P3.4 ;импульс записи

clr P3.4

mov a, b ;восстанавливаем выводимый код

swap a

anl a, #0FH ;отделяем старший полубайт

mov P3, a ;выводим его на индикатор

setb P3.6 ;вкл. Режим вывода данных

setb P3.4 ;импульс записи

clr P3.4

pop b

pop DPH ;восстановление регистров

pop DPL

ret

;подпрограмма вывода сообщения на ЖКИ №2

prstr2: push psw ;сохранение регистра psw

mov psw, #bank3 ;включаем банк 3

ps1: mov a, #0

movc a, @a+DPTR ;получение очередного символа

cjne a, #0, ps2 ;если это не последний выводим

pop psw ;завершение подпрограммы

ret

ps2: call prch2 ;вывод символа на экран

inc DPTR ;увеличиваем значение указателя

jmp ps1

;подпрограмма вывода сообщения об ошибке

prErr: push acc

call eclr2 ;очистка экрана

mov a, #2 ;курсор в позицию 2

call ecur2

mov DPTR, #serr ;вывести надпись «Error»

call prst2

pop acc

call prch2 ;вывод номера ошибки

ret

;сообщение «Error»

ORG 0680H

Serr: DB 14, 19, 19, 20, 19, 0 ;сообщение «Error», 0 в конце – признак окончания слова

;подпрограмма вывода Туст на ЖКИ №1

prtmu: mov a, #16

call prch1

mov a, bufLAN2

mov b, #10

div ab

call prch1

mov a, b

call prch1

mov a, #17

call prch

mov a, #0CH

call prch1

ret

;подпрограмма с длительностью выполнения примерно 3 мин. опроса клавиатуры и вывода Тпм

zader: mov R2, #7

za1: mov R1, #255

za2: mov R0, #56;

za3: mov LoopCnt, #255

za4: call Delay

djnz LoopCnt, za4

djnz R0, za3

call inkl ;опрос клавиатуры

jnz za5 ;переход при нажатии

call eclr2 ;вывод Тпм

mov a, #2

call ecur2

call sttmp

call rdtmp

call prtmp

djnz R1, za2

djnz R2, za1

mov a, #0

ret

za5: mov a, #1

ret

;подпрограмма запуска процесса измерения температуры

sttmp: push psw

mov psw, #bank2 ;выбор банка 2 памяти

stm1: call resLAN ;сброс

cjne a, #0, stm6 ;если ошибка переход

mov a, #0CCH ;пропуск ПЗУ

call wr8LAN

mov a, #44H ;запуск преобразования

call wr8LAN

stm4: call rd1LAN ;запрос об окончании преобразования

jne stm4 ;если не готово / повторить

stm5: pop psw ;выход из подпрограммы

ret

stm6: call prErr ;если ошибка, печать

jmp stm1 ;перейти в начало

;подпрограмма чтения температуры

;выходные параметры – температура в bufLAN - младший байт, в bufLAN+1 - старший байт

rdtmp: push psw

mov psw, #bank2 ;выбор банка 2 памяти

call resLAN ;вызов процедуры сброса

cjne a, #0, rtm2 ;если ошибка

mov a, #0CCH ;пропуск ПЗУ

call wr8LAN

mov a, #0BEH ;выдать команду

call wr8LAN

call rd8LAN ;чтение младшего байта

mov bufLAN, a ;запись в буфер

call rd8LAN ;чтение старшего байта

mov bufLAN+1, a ;запись в буфер

pop psw ;завершение программы

ret

rtm2: call prErr ;если ошибка, вывести на экран

jmp rtm1 ;перейти в начало

;подпрограмма формирования сигнала начального сброса

resLAN:clr P1.1 ;закорачиваем линию

mov LoopCnt, #100 ;задержка 700 мкс

rst1: call Delay

djnz LoopCnt, rst1

setb P1.1 ;отпускаем линию

mov LoopCnt, #9 ;задержка 65 мкс

rst2: call Delay

djnz LoopCnt, rst2

jnb P1.1, rst3 ;проверка сигнала присутствия

mov a, #1 ;код ошибки «нет датчика»

ret

rst3: mov LoopCnt, #71 ;задержка 500 мкс

rst4: call Delay

djnz LoopCnt, rst4

jb P1.1, rst5 ;если на линии 1 переход

mov a, #2 ;код ошибки «КЗ в линии»

ret

rst5: mov a, #0 ;код нормального завершения

ret

;подпрограмма чтения слота (через признак переноса)

mrslot: clr P1.1 ;закоротить линию

nop

setb P1.1 ;отпустить линию

call Delay ;задержка 13 мкс

call Delay

nop

nop

nop

mov C, P1.1 ;прочитать и сохранить бит

mov LoopCnt, #8 ;задержка 60 мкс

msl1: call Delay

djnz LoopCnt, msl1

ret

;подпрограмма записи 0 в слот

mwlow: clr P1.1 ;закоротить линию

mov LoopCnt, #11 ;задержка 80 мкс

mwl1: call Delay

djnz LoopCnt, mwl1

setb P1.1 ;отпустить линию

ret

;подпрограмма записи 1 в слот

mwhi: clr P1.1 ;закоротить линию

call Delay ;задержка 5 мкс

setb P1.1 ;отпустить линию

mov LoopCnt, #11 ;задержка 75 мкс

mwh1: call Delay

djnz LoopCnt, mwh1

ret

;процедура задержки 5 мкс (при тактовой частоте 12МГц)

Delay: nop

ret

;программа записи байта через аккумулятор

wr8LAN:

mov R1, #8 ;инициализация счетчика разрядов

wb1: rrc A ;младший бит в C

jc wb2 ;если текущий бит = 0

call mwlow ;запись 0 в слот

jmp wb3

wb2: call mwhi ;запись 1 в слот

wb3: djnz R1, wb1 ;продолжать, если не последний

ret

;подпрограмма записи бита, который передается через С

wr1LAN:

jc wb4 ;если бит = 0

call mwlow ;запись 0 в слот

jmp wb5

wb4: call mwhi ;запись 1 в слот

wb5:

ret

;подпрограмма чтения байта

rd8LAN:

mov R1, #8 ;счетчик количества битов

rb1: call mrslot ;чтение очередного бита

rrc A ;сдвиг битов младшим разрядом вперед

djnz R1, rb1 ;продолжать, если не последний

ret

;подпрограмма чтения бита

rd1LAN:

call mrslot

ret

;вывод Тпм наЖКИ №2

prtmp: mov a, #16 ;вывод пробела

call prch2

mov a, bufLAN+1 ;считываем старший байт

swap a ;обмен тетрад

anl a, #0F0H ;получили старший полубайт

mov R5, a ;временно записываем в R5

mov a, bufLAN ;считываем младший байт

swap a ;обмен тетрад

anl a, #0FH ;получили младший полубайт

add a, R5 ;складываем полубайты

mov b, #10 ;деление на 10

div ab

call prch2 ;вывод десятков

mov a, b ;берем остаток от деления

call prch2 ;вывод единиц

mov a, #17 ;вывод символа «градус»

call prch2

mov a, #0CH ;вывод буквы «С»(Цельсия)

call prch2

ret

;подпрограмма режима охлаждения

reoh: clr P0.2 ;выкл. Нагревательный элемент

setb P0.4 ;вкл. компрессор

setb P0.3 ;вкл. вентелятор

call zader

jz re4 ;переход при нажатой клавише

ret

re4: clr P0.4 ;выкл. компрессор

clr P0.3 ;выкл. вентелятор

call zader

jz re3 ;переход при нажатой клавише

ret

re3: mov a, bufLAN+1

swap a

anl a, #0F0H

mov R5, a

mov a, bufLAN

swap a

anl a, #0FH

add a, R5 ;в аккумуляторе байт Тпм

mov b, bufLAN2

mov b, #b-2

mov a, #LE(a, b) ;сравнение Тпм<=Туст-2

cjne a, #0FFFFH, re1 ;переход, если не истина

inc bufLAN3 ;счетчик циклов режима увеличить на 1

mov a, #GT(bufLAN3, 2)

cjne a, #0FFFFH, re2 ;сравнение «счетчик циклов>=2»

mov bufLAN3, #0 ;обнуление счетчика

mov a, #4 ;в режим обогрева

ret

re1: mov bufLAN3, #0 ;обнуление счетчика

re2: mov a, #5 ;режим охлаждения

ret

;конец программы

end