

Аннотация

В рамках данного дипломного проекта разработано программное обеспечение контроллера пульта вахтера автоматизированной проходной системы безопасности предприятия (АЛГО-421ПВ).

В настоящей пояснительной записке (ПЗ) приводится описание структуры разработанной программной системы, структуры некоторых программных блоков и базы данных, а также описание некоторых алгоритмов.

Данная ПЗ включает в себя руководство по записи программного продукта на носитель, руководство по программному тестированию компонентов контроллера, руководство пользователя пульта вахтера.

Приводится расчет затрат на разработку программного продукта.

Производится анализ опасных и вредных факторов, возникающих в процессе эксплуатации пульта вахтера автоматизированной проходной, а также анализ негативного воздействия на окружающую среду в процессе производства контроллера.

Используемые сокращения

В настоящей пояснительной записке (ПЗ) при изложении материала используются следующие сокращения:

- КПВ – контроллер пульта вахтера АЛГО-421ПВ;
- УК – управляющий компьютер;
- ТК – один из терминальных контроллеров, или контроллеров турникетов;
- СБП – система безопасности предприятия;
- БД – база данных;
- SE210 – микросхема взаимодействия с клавиатурой и светодиодной матрицей панели управления пульта вахтера;
- ПВ – пульт вахтера.

При ссылке на материалы графической части проекта в настоящей ПЗ были использованы номера 1...11. Соответствие между этими номерами листов и их десятичными номерами, а также названиями указано ниже.

№ в ПЗ	Децимальный номер	Название
1	ДП.16100-01 90 01	Система безопасности предприятия. Схема структурная
2	ДП.16100-01 90 02	Контроллер АЛГО-421ПВ. Схема структурная
3	ДП.16100-01 98 01	Протокол АЛГО-421.УК-ПВ
4	ДП.16100-01 98 02	Протокол АЛГО-421.ПВ-ТК
5	ДП.16100-01 91 01	Программная система управления контроллером пульта вахтера. Структуры данных
6	ДП.16100-01 90 04	Программная система. Схема структурная
7	ДП.16100-01 90 03	Основной блок. Схема структурная
8	ДП.16100-01 90 05	Блок принятия решений. Схема структурная
9	ДП.16100-01 81 01	Программная система. Схема алгоритма работы
10	ДП.16100-01 81 02	Функция поиска записи по коду пропуска. Схема алгоритма
11	ДП.16100-01 90 06	Рабочее место настройщика контроллера пульта вахтера. Схема структурная

Содержание

Введение	10
1. Исследовательская часть	11
1.1. Система безопасности предприятия	11
1.2. Структурные компоненты СБП с автоматизированными проходными	14
1.2.1. Пульт вахтера	14
1.2.1.1. КПВ, его роль и выполняемые функции	15
1.2.1.2. КПВ. Структура, аппаратные компоненты	16
1.2.1.3. Панель управления пульта вахтера	18
1.2.1.4. Пульт вахтера. Организация ввода и вывода	19
1.2.2. Управляющий компьютер. Обмен между УК и КПВ	20
1.2.2.1. Функция «Поллинг проверки активности»	21
1.2.2.2. Функция «Поллинг проверки событий»	22
1.2.2.3. Функции «Установить параметры» и «Получить параметры»	22
1.2.2.4. Функция «Операции над базой данных КПВ»	23
1.2.3. Проходы. Контроллеры турникетов (ТК). Обмен между КПВ и ТК	24
1.2.4. Элементы проходов: турникеты	25
1.3. Постановка задачи	27
1.4. Выбор средств разработки	28
2. Конструкторская часть	30
2.1. Вводные замечания	30
2.2. Разработка обобщенного алгоритма функционирования программной системы	33
2.3. Структура программной системы	37

2.3.1. Взаимосвязь между блоками. Информационные и управляющие потоки между блоками программной системы	39
2.3.2. Структуры данных	47
2.3.3. Основной блок программной системы (main)	53
2.3.3.1. Основные положения	53
2.3.3.2. Структура блока. Взаимосвязь между функциями	53
2.3.4. Блок принятия решений (decide)	57
2.3.4.1. Основные положения	57
2.3.4.2. Структура блока. Взаимосвязь между функциями	57
2.3.4.3. Описание алгоритма поиска и функции поиска по коду пропуска в БД контроллера FindCode ..	61
2.3.4.4. Описание функции идентификации пользователя IdentifyUser	64
3. Технологическая часть	66
3.1. Структурная схема рабочего места настройщика контроллера АЛГО-421ПВ	67
3.2. Руководство по записи программного продукта на носитель	69
3.2.1. Требования к носителю, техническим и программным средствам записи	69
3.2.2. Пользовательская маркировка микросхем	70
3.2.3. Подготовка к записи	70
3.2.4. Операции записи и сравнения	71
3.3. Руководство по программному тестированию компонентов контроллера АЛГО-421ПВ	75
3.3.1. Предварительные установки контроллера	75

3.3.2. Тестирование компонентов контроллера: подготовительный этап	75
3.3.3. Тестирование компонентов контроллера: основной этап	76
3.3.4. Тестирование компонентов контроллера: заключительный этап	81
3.4. Руководство пользователя пульта вахтера	82
3.4.1. Общие положения	82
3.4.2. Описание режимов работы проходов, функциональ- ных клавиш и индикаторов проходов панели управления ПВ	83
3.4.3. Описание функциональных клавиш и индикаторов особого назначения панели управления ПВ	84
4. Организационно-экономическая часть	86
4.1. Расчет затрат на разработку программного продукта	86
5. Обеспечение безопасности жизнедеятельности и охрана окружающей среды	91
5.1. Обеспечение безопасности труда	91
5.1.1. Классификация опасных и вредных производственных факторов, воздействующих на человека в процессе эксплуатации пульта вахтера	92
5.1.1.1. Неудовлетворительные параметры микроклимата	92
5.1.1.2. Повышенный уровень шума на рабочем месте .	94
5.1.1.3. Опасный уровень напряжения в электрической цепи, замыкание которой может произойти через человека	96
5.1.1.4. Пожарная опасность	97

5.1.1.5.	Неудовлетворительное освещение	99
5.1.1.6.	Психофизиологические факторы	99
5.1.2.	Классификация опасных и вредных производственных факторов, воздействующих на человека в процессе производства контроллера	101
5.1.2.1.	Выделение газов, паров и пылей	102
5.1.2.2.	Расчет бортового отсоса	103
5.2.	Охрана окружающей среды	105
5.2.1.	Факторы негативного воздействия на окружающую среду в процессе производства контроллера	105
5.2.2.	Расчет воздушного пылевого фильтра	107
Заключение		110
Список литературы		111
Приложения		112
П1.	Электрическая принципиальная схема контроллера АЛГО-421ПВ	113
П2.	Функциональное назначение основных выводов микроконтроллера	116
П3.	Шина I ² C – описание.....	117
П4.	Микросхема SE210 как элемент контроллера пульта вахтера	121
П5.	Система команд SE210	122
П6.	Алгоритмы расчета контрольной суммы короткого и расширенного пакетов протокола АЛГО-421.УК-ПВ	123
П7.	Обобщенные временные диаграммы реализации обмена по протоколу АЛГО-421.УК-ПВ	125
П8.	Текст программы на языке С для Motorola HC12	126
П8.1.	Текст основного блока программной системы (main)	127
П8.2.	Текст блока принятия решений (decide)	131

Введение

Разработанная в ходе работы над дипломным проектом **программная система** предназначена для управления контроллером пульта вахтера автоматизированной проходной и в совокупности с аппаратным обеспечением является одним из компонентов *систем безопасности предприятий*, о которых пойдет речь в п. 1.1. Автоматизированные проходные как разновидность (либо составная часть) систем безопасности предприятий (СБП) служат для обеспечения стабильной работы предприятия, главным образом для защиты их территории от несанкционированного доступа и в целях учета и контроля использования рабочего времени.

Разработанная в ходе выполнения проекта **документация** имеет следующую структуру.

Настоящая пояснительная записка содержит следующие основные части: **исследовательскую** (в ней анализируются системы безопасности с автоматизированными проходными и их компоненты, производится постановка задачи и выбор средств разработки); **конструкторскую** (приводится структура программной системы и некоторых ее блоков, рассматриваются некоторые алгоритмы); **технологическую** (рассматриваются вопросы настройки контроллера и эксплуатации пульта вахтера, записи программного обеспечения на носитель); **организационно-экономическую** (производится расчет затрат на разработку ПО); **часть по обеспечению безопасности жизнедеятельности и охране окружающей среды** (рассматриваются вопросы охраны труда и окружающей среды в процессе производства контроллера и эксплуатации пульта вахтера).

Графическая часть проекта также содержит чертежи и схемы, которые соответствуют исследовательской (листы 1 - 4), конструкторской (листы 5 - 10) и технологической (лист 11) частям пояснительной записки (список их десятичных номеров и названий приведен на странице 5).

1. Исследовательская часть

В рамках данной части дипломного проекта предстоит выяснить возможность создания программной системы управления контроллером пульта вахтера автоматизированной проходной системы безопасности предприятия (СБП), поставить задачу и выбрать средства и пути ее решения.

Программное обеспечение по теме настоящего проекта разрабатывается для одной из подсистем (компонент) системы безопасности предприятий, о которых пойдет речь в следующем пункте.

1.1. Система безопасности предприятия

Целью создания систем безопасности предприятий (СБП) является обеспечение решения ряда задач, актуальных практически для любого предприятия. К таким задачам можно отнести:

- задачу защиты предприятия и его имущества от несанкционированного доступа на территорию предприятия посторонних лиц;
- задачу табельного учета рабочего времени на основе норм и нормативов, а также создания условий, затрудняющих либо делающих невозможным нарушение трудового распорядка;
- задачу охраны труда и окружающей среды, а также имущества предприятия от чрезвычайных происшествий и стихийных бедствий, например, оповещения о резких изменениях параметров микроклимата помещения (повышенная температура, дым и т.д.) в связи с возникающим пожаром.

Потребность в автоматизации СБП и, в частности, проходных становится актуальной с ростом числа работников предприятия, с усилением требований к безопасности (на предприятиях особого режима – военных,

энергетических и прочих объектах стратегического назначения) и ужесточения условий, в которых приходится работать обслуживающему персоналу (на территории химических комбинатов, атомных станций).

Для повышения эффективности решения перечисленных выше задач в последние годы различными фирмами (например, в России – фирмы «Алгонт», «ААМ Systems», «Gulfstream»; за рубежом – «CheckPoint», «Apollo», «PelCo» и др.) разрабатываются комплексные автоматизированные системы безопасности.

Так, фирмой «Алгонт» (г. Калуга) были разработаны и разрабатываются аппаратные и программные компоненты автоматизированных СБП, такие как:

- контроллеры высокого уровня (Master, АЛГО-421М), и терминальные контроллеры (Slave, АЛГО-421Т). Контроллеры высокого уровня физически представляют собой печатную плату, содержащую, как правило, микроконтроллер, микросхемы постоянной и оперативной памяти, таймер, шинные формирователи и микросхемы внешнего интерфейса;
- подсистемы охранно-пожарной сигнализации и видеонаблюдения. Реализованы в виде датчиков и устройств индикации, подключаемых к платам расширения (АЛГО-422) Master- и Slave-контроллеров;
- автоматизированные проходные. Ими оборудуются как внешние, так и внутренние контрольно-пропускные пункты (КПП) предприятий. Компонентами этих проходных являются *турникеты* (либо специальные шлюзовые кабины) и *пульт вахтера*, а также считыватели, клавиатуры и бункеры сбора разовых пропусков;

- подсистема учета въезда и выезда автотранспорта. Реализуется на тех же контроллерах, что и система доступа. В качестве же исполнительных механизмов (объектов управления) здесь применяются автоматизированные шлагбаумы, ворота и противотаранные устройства;
- управляющие (головные) компьютеры с установленном на них ядром системы АССаД32 (автоматизированная система санкционированного доступа для Windows NT). Система АССаД32 включает в себя машину баз данных, модуль табельного учета рабочего времени, подсистему подготовки пропусков и организует глобальное взаимодействие указанных выше подсистем на программном уровне.

Как видим, системы безопасности предприятий могут иметь различные конфигурации (с автоматизированными проходными, с подсистемой учета въезда автотранспорта, с подсистемой подготовки пропусков, с подсистемой пожарной сигнализации, с подсистемой видеонаблюдения, с подсистемой контроля доступа в комнаты и помещение через двери и т.д.) либо представлять собой их сочетания. Рассмотрим далее основные структурные компоненты СБП с автоматизированными проходными более подробно.

1.2. Структурные компоненты СБП с автоматизированными проходными

В графической части дипломного проекта (лист 1) приведена структурная схема данного вида СБП, включающая следующие компоненты: *пульт вахтера* (панель управления и контроллер АЛГО-421ПВ); *управляющий компьютер* и центральная база данных; *ТК* (терминальные контроллеры); *проходы* автоматизированной проходной (турникеты, устройства для ввода кодов). Опишем отдельные ее компоненты.

1.2.1. Пульт вахтера

На пульт вахтера возлагают выполнение, *вообще говоря*, следующих функций:

- сбор и регистрация информации о проходах, включая время прохода и идентификатор проходивших людей (номер карточки, пропуска либо другого документа);
- сверка номера карточки (или пропуска) и времени прохода с разрешенными значениями и принятие решения о разрешении или запрещении прохода;
- сигнализация об особых случаях, таких как попытка несанкционированно преодолеть проход, нарушение целостности элементов проходной и т.д.

Изначально эти функции выполнял вахтер - человек, находящийся в специальной кабине со смотровым стеклом и оборудованной в лучшем случае кнопками пульта (например, включение освещения, сирены и т.д.).

Однако с увеличением численности работающих, с ужесточением требований к соблюдению трудового распорядка и охраны имущества

предприятия, а также в связи с нежелательностью работы вахтера в условиях вредных производств возникла потребность в автоматизации пульта вахтера с тем, чтобы часть или весь комплекс функций вахтера переложить на устройство - контроллер пульта вахтера (КПВ).

1.2.1.1. Контроллер пульта вахтера, его роль и выполняемые функции

Контроллер пульта вахтера призван работать в условиях механизированного и автоматизированного окружения (см. п. 1.1, компоненты автоматизированной системы безопасности).

В связи с этим формулировка функций, которые он должен выполнять, будет отличаться от перечисленных выше. Так, контроллер пульта вахтера должен выполнять следующие функции:

- реакция на попытку прохода, заключающаяся в получении информации о проходе от контроллеров турникетов и пересылке ее на управляющий компьютер;
- получение от управляющего компьютера команд разрешения либо запрещения прохода через турникет;
- передача информации о разрешении либо запрещении прохода на соответствующий турникету контроллер более низкого уровня;
- проверка исправности собственных компонентов и турникетов;
- обработка сигналов ручного управления и индикация состояний проходов.

В отличие от старой модели контроллера пульта вахтера (КПВ-М2), к новому контроллеру АЛГО-421ПВ (а значит, и к разрабатываемому для него программному обеспечению) предъявляется требование работы не только в **онлайновом** (т.е. постоянной связи с управляющим компьютером), но и в **оффлайновом** (автономном) режиме, что приводит к возникновению дополнительных функций:

- возможность накапливания событий о проходах в базе событий, располагающейся в памяти данных контроллера;
- возможность принятия решения о проходах на основе анализа кода пропуска, пин-кода (особый код, вводимый с клавиатуры возле прохода), времени и направления прохода; лишь в том случае, если код пропуска не найден в базе данных КПВ, решение о проходе принимается управляющим компьютером.

1.2.1.2. Контроллер пульта вахтера. Структура, аппаратные компоненты

Размер внутренней (регистровой) памяти микроконтроллера невелик (например, для микроконтроллера Motorola MC68HC812A4 он составляет всего 1 Кбайт, из которого под регистры особого назначения отводится около половины адресуемого пространства), поэтому возникает необходимость использования внешней оперативной памяти, конструктивно микросхемы которой располагаются на плате контроллера.

При проектировании старой версии КПВ (КПВ-М2) в качестве устройства управления был взят микроконтроллер Intel P80C32 (8-разрядная шина данных и 16-разрядная шина адреса).

Однако для построения нового пульта вахтера требуется микроконтроллер с большей разрядностью и быстродействием. Большая разрядность шины адреса необходима для того, чтобы разместить таблицы базы данных (адресовать пространство внешней оперативной памяти размером в 1МБайт). Для этого используется 21 линия адресной шины (самый старший и самый младший ее биты используются для формирования сигналов выборки микросхемы памяти). Каждая микросхема ОП имеет объем 512К, за один раз считывается 1 байт информации.

Итак, перечислим основные элементы контроллера АЛГО-421ПВ:

- микроконтроллер Motorola MC68HC812A4CPV8;
- таймер DS130SEN (Часы реального времени);
- микросхемы ПЗУ AT29C512/15 (2 шт.) – здесь будет размещаться разрабатываемая программа управления;
- микросхемы ОЗУ M5M5408AFP-70LL-I (2 шт.);
- DIP- переключатели конфигурации SA1 и микросхема MC74AC245DW считывания конфигурации контроллера;
- формирователи данных MC74AC245DW (2 шт.), по 8 разрядов каждый;
- генераторы тактовой частоты BQ1 (синхронизация работы микроконтроллера) и BQ2 (синхронизация работы часов реального времени);
- микросхема (микроконтроллер) интерфейса светодиодной матрицы и клавиатуры пульта вахтера CE210;
- шинные усилители и формирователи;
- микросхема MAX1480BEP1 интерфейса связи по физическому протоколу RS-485 с управляющим компьютером;
- микросхема интерфейса связи по физическому протоколу RS-485 с контроллерами турникетов MAX483EPA;
- другие элементы.

Принципиальная электрическая схема контроллера приведена в приложении П1.

Назначение основных выводов управляющего элемента контроллера (микроконтроллер Motorola MC68HC812A4) описано в приложении П2.

С архитектурой микроконтроллера Motorola можно ознакомиться в [2], а с его системой команд – в [1].

1.2.1.3. Панель управления пульта вахтера

Панель управления пульта вахтера (см. графическую часть, лист 1) должна обладать матрицей светодиодов (общее количество светодиодов $4 \cdot 10 + 5$) и клавиатурой ($2 \cdot 10 + 1$ клавиш). Дело в том, что данный пульт вахтера рассчитан на 10 проходов (хотя реальное количество может быть и меньшим), причем каждому из проходов должны соответствовать 4 светодиода и две клавиши.

На рисунке 1.1 кружком обозначены светодиоды, а небольшим прямоугольником – клавиши, относящиеся к каждому из проходов. Индикация автоматического режима имеет место в случае, если проход заблокирован или освобожден клавишами ручного задержания «Блокировка» и «Задержание». Клавиша «Блокировка прохода» означает, что проход заблокирован до тех пор, пока не будет еще раз нажата данная кнопка. Клавиша «Освобождение прохода» означает, что проход будет **безусловно** открыт до тех пор, пока не будет нажата еще раз данная клавиша.

Индикация аварийного состояния включается при возникновении сбоев в работе прохода, это состояние отслеживается контроллером турникета и передается на АЛГО-421ПВ. Индикация аварийного состояния должна дублироваться звуковым сигналом.

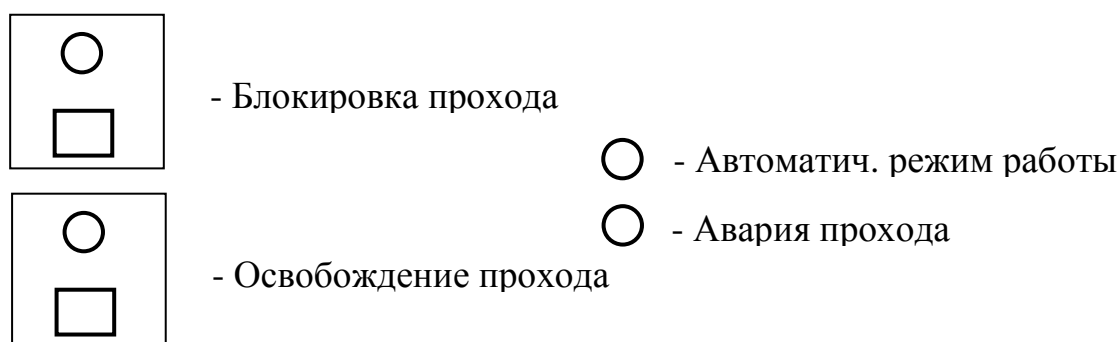


Рис. 1.1. Клавиши и индикация пульта вахтера (на 1 проход)

Помимо перечисленных выше, существует также одна клавиша **ручного сброса задержания** «Сброс задержания» (позиция б на структурной

схеме) и соответствующий ей один **светодиод 5** (включен, когда имеет место задержание по команде «Блокировать»). Имеется также один светодиод 7 «Несанкционированный проход», когда имеет место попытка прохода без пропуска, а также светодиод 10 «Нет связи с УК», сигнализирующий об отсутствии связи с управляющим компьютером. Наконец, имеется светодиод, сигнализирующий, что пульт включен (8), данный светодиод не поддается программному управлению.

1.2.1.4. Пульт вахтера. Организация ввода и вывода

Для организации ввода и вывода используются (см. графическую часть, лист 2) блок формирования сигналов управления для микросхемы CE210, собственно блок управления вводом/выводом сигналов ручного управления CE210, блок формирования сигналов клавиатуры и индикации, а также динамик, конструктивно расположенный на плате контроллера.

Обмен между микроконтроллером Motorola и микросхемой CE210 основан на эмуляции асинхронной шины последовательного обмена I²C (см. приложение П3; [3], с. 157-164). Линии SC и SD служат для передачи синхросигналов и данных соответственно в сторону CE210, а SCL и SDA – в сторону M68HC812A4.

Обобщенная временная диаграмма работы шины I²C и основные временные параметры, общий формат передаваемых пакетов, а также их описание приведены в приложении П3.

Описание структуры организации микросхемы CE210, а также описание системы команд приведены в [3], с.185-202, а также в приложениях П4 – П5.

1.2.2. Управляющий компьютер (УК). Обмен между УК и КПВ

Данный компонент является основным практически для всех СБП, он связывает воедино все остальные компоненты СБП. Управляющий компьютер зачастую выполняет и роль сервера, то есть содержит центральную базу данных и организует запросы к ней.

Как видно из структурной схемы (см. графическую часть, лист 1), на управляющем компьютере должно быть установлено следующее ПО: операционная система Windows NT 4.0 и выше, ядро СУБД Oracle 8.0, а также одна из версий ядра разработанной специалистами фирмы «Алгонт» программной системы АССаД32. Ядро АССаД32 выполняет функции получения и отправления данных в соответствии с протоколом обмена АЛГО-421 и его модификациями (например, АЛГО-421.УК-ПВ).

Как уже отмечалось выше, физическим протоколом обмена между контроллером и управляющим компьютером является протокол последовательного обмена RS-485 (скорость обмена по линии – 4800 бит/сек.). Логический же протокол обмена (АЛГО-421ПВ), по-видимому, будет базироваться на уже существующем протоколе «АЛГО-421», который использовался ранее при обмене данными между мастер-контроллерами типа «АЛГО-421х» и управляющим компьютером. Этот протокол на практике показал свою надежность, кроме того, использование его в качестве основы при разработке нового протокола было рекомендовано заказчиком.

В данном протоколе различают *короткие* (имеющие фиксированную длину) и *расширенные* (имеющие переменную длину, не превышающую 64-х байт) пакеты. Первые три байта расширенного пакета имеют ту же структуру, что и короткий пакет. **Форматы передаваемых пакетов** приведены в графической части дипломного проекта (лист 3).

Так как в линии связи могут возникать разного рода помехи, приводящие в ряде случаев к искажению передаваемой информации, пакеты должны содержать байты контрольной суммы (на приемной стороне

контрольная сумма подсчитывается и сравнивается с байтом контрольной суммы). Различают контрольную сумму короткого и расширенного пакетов. Алгоритмы подсчета контрольных сумм приведены в приложении П..., а обобщенные временные диаграммы обмена в соответствии с протоколом АЛГО-421ПВ – в приложении П....

Четырехбитное поле функции пакета в третьем байте (см. графическую часть, лист 3) может содержать одно из следующих значений:

Таблица 1.1. Обмен с УК. Функции пакета

Код функции	Тип функции
0	Поллинг проверки активности (Acknowledge Poll)
1	Поллинг проверки событий (Event Poll)
2	-
3	Установить параметры (Set Parameters)
4	Получить параметры (Get Parameters)
5	-
6	Операции над базой данных КПВ (Long)
Другой	-

Охарактеризуем приведенные ниже функции.

1.2.2.1. Функция «Поллинг проверки активности»

Используется в **коротком** пакете, посылаемом УК на КПВ для определения активности КПВ (т.е. его способности работать с УК). КПВ отвечает тем же коротким пакетом, что и был получен. Пакет не может быть передан на ТК.

1.2.2.2. Функция «Поллинг проверки событий»

Используется в **коротком** пакете, посылаемом УК на КПВ для определения, произошли ли какие-либо события на КПВ и ТК. Если событий не произошло, КПВ отвечает тем же коротким пакетом, что и был им получен. Если же имели место события, КПВ отвечает расширенным пакетом в соответствии с протоколом.

Пакет с данной функцией не может быть передан на ТК. КПВ через определенные промежутки времени опрашивает ТК на предмет наличия событий и сохраняют эти события в собственных очередях событий. По получении короткого пакета с данной функцией КПВ отвечает расширенным пакетом, содержащим признаки событий.

1.2.2.3. Функции «Установить параметры» и «Получить параметры»

Используются управляющим компьютером для того, чтобы внести изменения в базу данных КПВ; для того, чтобы установить те или иные параметры КПВ (ТК), а также получить состояние отслеживаемых параметров КПВ или ТК. Включают в себя как непосредственно адресуемые КПВ команды (прямые команды), так и команды передачи.

Например, чтобы установить время на часах реального времени КПВ, УК посылает **прямую** команду Set Clock на КПВ. Чтобы установить время на часах реального времени ТК, УК посылает **команду передачи** Set Clock на КПВ. Адрес ТК включен в расширенный пакет (пятый байт). После этого КПВ пересылает команду Set Clock на ТК.

Чтобы получить значения параметров (данные) от ТК, УК должен послать соответствующую **команду передачи** на КПВ. КПВ должен затем запросить данные у ТК. Пока КПВ занят этим, УК должен периодически посылать пакеты Response Poll на КПВ. Пока КПВ занят, он просто пересылает эхо-повтор Response Poll на УК. Но как только КПВ получает

запрошенные ранее данные от ТК, он (КПВ) должен ответить на очередной Response Poll расширенным пакетом. Полученные данные должны быть включены в расширенный пакет.

После того, как УК получил пакет с событием в ответ на Event Poll и определил, что контрольная сумма верна, он (УК) должен послать особый байт опознания события (\$06) на КПВ. Тот факт, что КПВ не получил такого байта, означает, что УК не получил пакет события. В этом случае КПВ должен послать **тот же самый** пакет события в ответ на следующий поллинг события.

1.2.2.4. Функция «Операции над базой данных КПВ»

Выполнение команд, относящихся к данной функции, занимает (по сравнению с другими командами) довольно много времени, так как производятся операции с обширными массивами данных, записываемых в память или хранящихся в памяти данных контроллера.

К данной функции относятся следующие команды:

- вставить запись БД пользователей;
- удалить запись БД;
- обнулить статус повторного прохода для всех пользователей;
- удалить всех пользователей, относящихся к заданному уровню доступа.

1.2.3. Проходы. Контроллеры турникетов, или терминальные контроллеры (ТК). Обмен между КПВ и ТК

Согласно техническому заданию, всего в проектируемой системе на каждой проходной может быть до десяти проходов. Каждый пульт вахтера соответствует одной проходной. Каждый контроллер турникета может обслуживать до двух проходов (один проход – на вход и один – на выход).

Физическим протоколом обмена между контроллерами турникетов, или терминальными контроллерами (ТК), и контроллером пульта вахтера (КПВ) является протокол последовательного обмена RS-485. Логический протокол обмена (АЛГО-421.ПВ-ТК) был разработан и рекомендован к применению специалистами фирмы «Алгонт». Он устанавливает требования к формату передаваемых пакетов данных, а также к возможным командам.

Сразу заметим, что в целях исключения столкновений в данном протоколе реализована одна из программно реализуемых разновидностей метода маркерного доступа – метод *поллинга*.

Формат пакета, также а коды и параметры команд приведены в графической части дипломного проекта, лист 4.

Поясним некоторые команды. Команда поллинга \$F0 посылается на каждый ТК через определенный интервал времени (например, 5 сек.), наибольшее время ожидания ответа на поллинг – порядка 70 мсек. Биты пересылаемого во время поллинга байта состояния сообщают нужному ТК о том, что оператором были нажаты те или иные кнопки панели пульта вахтера.

На команду поллинга ТК должен ответить одной из команд \$FC. Команда \$FC пересылается в случае, если не возникло новых событий, связанных с проходами. В байте данных команды \$FC могут присутствовать признаки присутствия того или иного прохода, признак необходимости

генерации звука динамиком пульта вахтера, признак того, что имеет место задержание (выполнена команда «Блокировать»); признак попытки несанкционированного прохода (без пропуска), а также признаки сбоя (аварии) по каждому из проходов.

Команда \$F1 («Запрос на вход») посылается от ТК, когда считан код пропуска и (возможно) пин-код и пользователь желает пройти через турникет.

Команды \$F3...\$F5 посылаются на ТК после принятия решения о проходе, причем команда «Пропустить» имеет место, если совпадают время прохода и все введенные коды с допустимыми; команда «Не пропускать» имеет место, если код пропуска не найден в базе данных пользователей КПВ и, кроме того, отсутствует в центральной БД управляющего компьютера. Команда «Блокировать» имеет место, если код пропуска найден в БД КПВ, но остальные параметры пользователя (пин-код, время либо направление прохода) не соответствуют требуемым.

Команда «Проход состоялся» (\$F6) передается от ТК в случае, если пользователь успешно прошел через турникет.

1.2.4. Элементы проходов: турникеты

Фирмой «Алгонт» выпускаются следующие разновидности турникетов:

- нормально открытый турникет АЛГО-111С (см. рис. 1.2а). Основное достоинство нормально открытых турникетов заключается в том, что отсутствует какое-либо срабатывание механических частей в момент совершения прохода. Как правило, такие турникеты лучше и с точки зрения вандалозащищенности. На крышку турникетов устанавливаются считыватели пластиковых карт, двухцветные индикаторы проходов и, при необходимости, клавиатура для ввода пин-кода;

- нормально закрытый турникет АЛГО-122Т (см. рис. 1.2б). Имеет два датчика прохода. По последовательности срабатывания датчиков электроника, управляющая турникетом, может фиксировать не только сам факт прохода, но и его направление;
- нормально закрытый турникет АЛГО-121Т (см. рис. 1.2в). Как и АЛГО-121Т, хорошо вписывается в интерьер офисных помещений. По последовательности срабатывания датчиков электроника, управляющая турникетом, может фиксировать не только сам факт прохода, но и его направление.



Рис.1.2а



Рис.1.2б



Рис.1.2в

1.3. Постановка задачи

Итак, все потоки информации в рассматриваемой автоматизированной системе безопасности можно разделить на 3 большие группы (см. рис. 1.3):

- двунаправленная связь с **управляющим компьютером** (с управляющего компьютера загружаются справочные таблицы БД, а также передаются некоторые команды, определяемые используемым протоколом передачи; на управляющий компьютер передается фактическая информация о проходах, а также ответные команды);
- двунаправленная связь с **контроллерами турникетов**;
- связь с **клавиатурой и индикаторами** пульта вахтера.

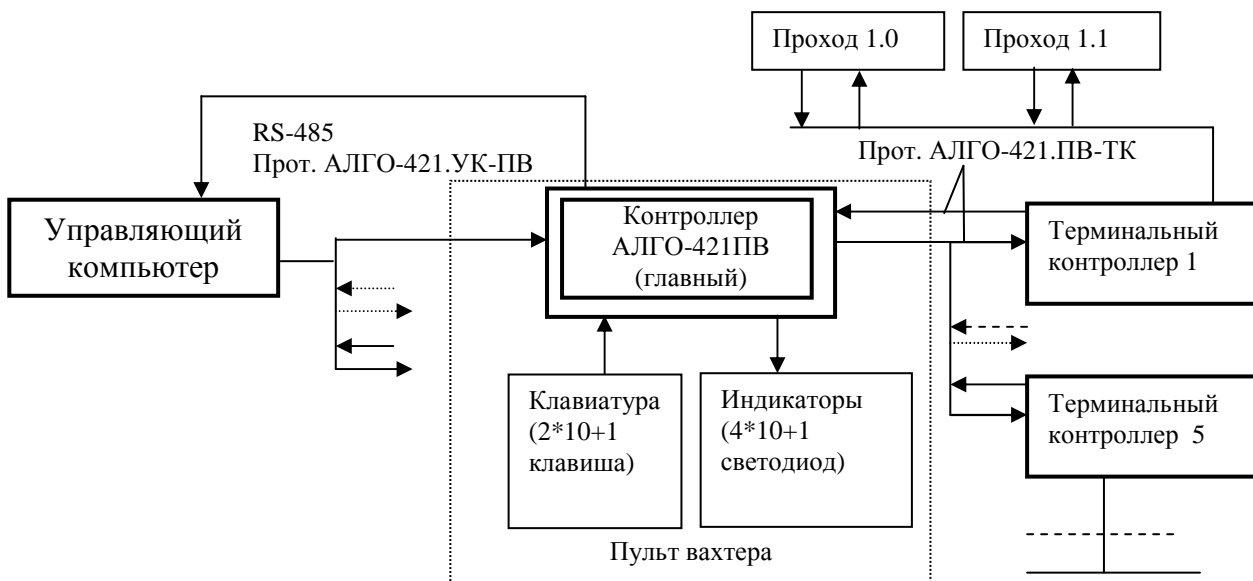


Рис. 1.3. Контроллер пульта вахтера АЛГО-421ПВ в системе безопасности предприятия

Теперь сформулируем задачу, которую предстоит решить разработчику программного обеспечения в ходе работы над дипломным проектом.

Задача состоит в том, чтобы разработать программное обеспечение контроллера пульта вахтера, которое физически будет располагаться в его Flash-ПЗУ и обеспечивать функциональное взаимодействие пульта вахтера с другими элементами автоматизированной системы безопасности (см. рис. 1.3); при этом программная система должна удовлетворять требованиям к ее функциональным характеристикам, надежности, информационной и программной совместимости, режимам работы, приведенным в техническом задании на настоящий дипломный проект. Кроме выполнения основных функций, программная система должна обеспечивать инициализацию и тестирование (в особом – тестовом - режиме) компонентов контроллера.

Замечание. В дальнейшем будем использовать следующие сокращения: УК – управляющий компьютер; ТК – один из терминальных контроллеров; ТК1, ТК2... – терминальный контроллер соответствующего порядкового номера; КПВ – контроллер пульта вахтера.

1.4. Выбор средств разработки

Для разработки программного обеспечения был выбран пакет прикладных программ «HIWARE TOOLS M68C12» фирмы © HIWARE, 1997 (см. [4], [5], [6]) по следующим причинам:

- этот комплект включает в себя отдельные программы настройки, редактирования (запуск какого-либо редактора Windows), отладчика, эмулятора, кросс-компилятора, линковщика, а также программы, превращающей промежуточный объектный код в собственно программу прошивки и др. Кросс-компилятор переводит операторы языка высокого уровня (C, C++) не в объектный код процессора персонального компьютера, а в объектный код микроконтроллера Motorola;

- данный пакет обеспечивает кросс-компиляцию с языка C/C++, одновременно являющегося и языком высокого уровня, и языком системного программирования (так как он включает в себя поразрядные операции). Кросс-компилятор с этого языка обеспечивает оптимизацию объектного кода, минимальную его избыточность (см. [4]);
- по сравнению с аналогичным компилятором, разработанным фирмой Cosmic Software [7], кросс-компилятор HIWARE поддерживает страничную (bank- модель) адресацию памяти путем объявления сегментов в расширенной памяти с помощью директивы far;
- заказчиком был рекомендован данный пакет в целях совместимости программы с некоторыми ранее разработанными типовыми модулями, разработанными с использованием языка и средств пакета фирмы HIWARE.

2. Конструкторская часть

В данной части дипломного проекта дано описание структуры и функционального назначения программной системы и некоторых ее блоков, раскрыты взаимосвязи между блоками программной системы, а также особенности реализации некоторых алгоритмов.

2.1. Вводные замечания

Всякая разрабатываемая система управления (либо собственно субъект управления СУ) может быть представлена в первом приближении как «черный ящик», реализующий некоторое отображение вектора входных воздействий X на выходной вектор Y , $F: X \rightarrow Y$, или $Y = F[X]$ (см. рис. 2.1).



Рис. 2.1. Субъект управления как «черный ящик»

В качестве компонент вектора входных воздействий $\{x_i\}$ обычно выступают осведомительные сигналы, являющиеся функциями состояния объекта управления (т.е. управляемого объекта), а также прочие (внешние) сигналы. В качестве компонент выходного вектора $\{y_i\}$ обычно выступают управляющие воздействия субъекта управления (управляющий элемент системы) на объект управления (управляемый элемент системы).

Будем рассматривать как субъект управления **только разрабатываемую программную систему, аппаратно реализуемую на микроконтроллере Motorola MC68HC812A4** (а не управляющий компьютер и его процессы, которые, хотя и являются субъектами управления

более высокого уровня, в данном контексте могут и должны рассматриваться как источники внешних воздействий). В этом случае *объектом непосредственного управления* будут являться некоторые другие аппаратные компоненты контроллера АЛГО-421ПВ (микроконтроллер индикации и ввода клавиатуры, таймер), а также *опосредованного управления* - контроллеры турникетов (опосредованного – т.к. передача сигналов и информации опосредуется через некоторые элементы контроллера).

При этом компоненты x_i представляются:

- сигналами со стороны объектов непосредственного управления (сигналы прерывания от таймера и микроконтроллера SE210 клавиатуры, сброса от супервизора MAX817);
- информационными потоками со стороны объектов непосредственного управления (информация о текущих дате и времени от часов реального времени, информационные пакеты от элемента SE210 клавиатуры и индикации);
- информационными потоками со стороны объектов опосредованного управления (информационные пакеты со стороны терминальных контроллеров, передаваемые в соответствии с логическим протоколом АЛГО-421.ПВ-ТК);
- информационными потоками извне (байты командных пакетов и байты подтверждения приема события со стороны управляющего компьютера, передаваемые в соответствии с логическим протоколом АЛГО-421.УК-ПВ).

Компоненты y_i , в свою очередь, представляются:

- информационными потоками в сторону объектов опосредованного управления (командные пакеты, отправляемые на терминальные контроллеры);

- сигналами воздействия на объекты непосредственного управления (управляющий обменом с микросхемой CE210 сигнал SC, сигнал выбора часов PB CSW, сигнал пересброса супервизора MAX817);
- информационными потоками на объекты непосредственного управления (командные пакеты, направляемые на микросхему CE210; информационные байты, пересылаемые на часы реального времени (ЧРВ) в процессе установки даты и времени);
- информационными потоками на внешние объекты (информационные пакеты в сторону управляющего компьютера).

Прежде, чем описать структуру проектируемой программной системы, рассмотрим обобщенный алгоритм ее функционирования в заданном окружении (рассматривается обычный, не тестовый режим функционирования), который приведен на листе 9 графической части дипломного проекта.

2.2. Разработка обобщенного алгоритма функционирования программной системы

При описании данного алгоритма производятся ссылки на номера соответствующих блоков в схеме алгоритма на листе 9 графической части дипломного проекта, указываемые в скобках.

Вначале должна быть произведена инициализация внутренних регистров микроконтроллера, а также компонентов контроллера, загрузка базы данных пользователей и построение индексного массива для ускорения операций поиска (2).

Далее могут возникнуть *следующие основные ситуации*:

- поступление прерывания от внутреннего таймера (4);
- поступление сигнала ручного управления от микросхемы CE210 (3);
- получен и подлежит анализу пакет от управляющего компьютера (7);
- получен и подлежит анализу пакет от одного из ТК (8);
- настало время поллинга (опроса) очередного ТК.

Все эти ситуации должны постоянно или периодически отслеживаться в ходе выполнения программы. Рассмотрим реакции программной системы на них.

Поступило прерывание от внутреннего таймера. Необходимо его обработать, то есть выполнить действия, связанные с регулярно выполняемыми действиями, например, модификацию счетчиков времени (5), которые будут использованы в программе, а также запустить процесс (в дальнейшем мы будем называть его фоновым) анализа остальных основных ситуаций (кроме ситуации поступления сигнала от микросхемы CE210, вызывающего свой обработчик прерывания) (6).

Поступил сигнал ручного управления. Вахтер нажал на одну из клавиш панели управления. В этом случае необходимо побитно считать (14), а затем проанализировать (15) код нажатой клавиши, получив из него номер прохода, номер ТК и тип нажатой клавиши. В том случае, если соответствующий данной клавише ТК присутствует в системе (16), анализ продолжается. Если была нажата клавиша «Блокировка прохода», то необходимо изменить индикацию авторежима прохода (новая индикация авторежима является инверсией нового состояния индикатора блокировки, или совпадает с прежним состоянием индикатора), после чего инвертировать индикацию и хранимый в памяти признак блокировки, а также обнулить индикацию и признак блокировки. Признаки нажатия на ту или иную клавишу необходимо хранить в памяти с той целью, чтобы затем передавать на ТК во время опроса. В случае освобождения поступают аналогично, с той лишь разницей, что слова «Блокировка» и «Освобождение» следует поменять местами.

В случае, если была нажата клавиша «Сброс задержания» (17), необходимо сбросить признак задержания, хранимый в памяти.

На схеме алгоритма все описанные выше случаи упрощенно представлены в виде блоков 18 («Изменение индикации») и 19 («Изменение признака»).

Получен и подлежит анализу пакет от управляющего компьютера. В этом случае необходимо проанализировать функцию полученного пакета (см. п. 1.2.2) (20) и выполнить действия в соответствии с протоколом обмена: в случае функции «Проверка активности» - ответить коротким пакетом (27); в случае функции «Проверка событий ТК» - ответить коротким пакетом, если новых событий не произошло (22), и отправить последнее непосланное событие ТК из памяти контроллера на компьютер, если со времени предыдущего опроса произошло новое событие (28). На посланное событие в

соответствии с протоколом обмена ожидается байт подтверждения (ожидание с таймаутом).

В случае функции «Установить параметры» либо устанавливаются параметры КПВ и посылается ответ коротким пакетом (если выполнено условие блока 23) (29), либо команда пересылается на ТК в соответствующем формате (например, команда «Пропустить», если код найден в центральной БД) (24).

В случае функции «Получить параметры» получают требуемые параметры контроллера (25) и пересылаются на УК в соответствии с полученной командой (30).

В случае функции «Модификация БД КПВ» анализируется код команды полученного пакета (26), выполняются операции над БД КПВ (см. п. 1.2.2.4) (31) и посылается ответ на УК коротким пакетом.

Получен и подлежит анализу пакет от ТК. Получен пакет от ТК может быть только в ответ на поллинг (см. п. 1.2.3). Анализируется код команды (36). Если команда – «Проход состоялся», то в память записывается новое событие «Проход состоялся» (35) и посылается подтверждение на ТК (37).

В случае получения обычного ответа на поллинг поочередно анализируются биты полученного байта состояния ТК (38). В ответ на это соответствующим образом изменяется индикация панели пульта вахтера (39) либо начинается / продолжается выдача звукового сигнала, если необходимо (см. п. 1.2.3) (41). В случае команды «Запрос на вход» анализируется наличие и правильность введенного пользователем кода пропуска, наличие и правильность ввода пин-кода (если нужно), наличие повторного прохода и времени прохода (на схеме алгоритма данные действия обозначены блоком 32 «Анализ события»). Во всех случаях в память КПВ заносится новое событие (35), а если код найден в памяти КПВ, то еще и посылается на ТК команда «Пропустить» либо «Блокировать» (34).

Настало время поллинга (опроса) очередного ТК. На обобщенной схеме алгоритма процесс поллинга помечен блоком 9.

В соответствии с алгоритмом поллинга процесс поллинга может быть продолжен лишь в том случае, если был получен ответ от опрошенного ранее ТК. После этого в заданной последовательности по кругу анализируются остальные ТК на предмет их присутствия в системе (об этом говорит так называемая карта терминальных контроллеров) и приоритета поллинга. В начальный момент времени все ТК имеют равные приоритеты (количества их неответов на поллинг равны нулю). В том случае, если какой-либо из ТК не ответил на поллинг, его приоритет падает: он уже опрашивается каждый пятый раз, если не ответил еще раз – каждый десятый и т. д. Как только данный ТК отвечает на очередной поллинг, его приоритет поллинга восстанавливается до первоначального значения. В процессе поллинга в соответствии с протоколом (п. 1.2.3) на ТК посылается байт, каждый бит которого информирует, была ли нажата та или иная клавиша панели пульта вахтера.

Помимо описанных выше основных ситуаций, на схеме алгоритма отдельно рассмотрены случаи поступления одиночного байта от ТК и УК, по которым вызываются соответствующие обработчики прерываний:

- в случае поступления байта от ТК (10) поступивший байт записывается в буфер обмена с ТК (11);
- в случае поступления байта пакета от УК (12) поступивший байт записывается в буфер обмена с УК (13). Байт подтверждения получения события в буфер не записывается.

В обоих случаях программно зарезервированы буферы, длина которых равна максимальной длине соответствующих пакетов (от ТК, от УК).

2.3. Структура программной системы

Раскроем теперь структуру программной системы, которая будет определять описанное выше отображение, то есть удовлетворять предъявляемым к ней требованиям управления в системе безопасности, и реализовывать описанный выше обобщенный алгоритм работы.

В самом общем виде данная структура (хотя и с определенной долей условности) может быть представлена в виде **следующих структурно-функциональных блоков** (см. графическую часть, лист 6):

- Основной блок (запуска, инициализации и тестирования контроллера);
- Блок управления памятью данных (контроллера);
- Блок управления обменом с терминальными контроллерами (ТК);
- Блок управления обменом с управляющим компьютером (УК);
- Блок взаимодействия с элементом ввода и индикации сигналов ручного управления пульта вахтера (далее будем называть “блок ввода и индикации”);
- Блок принятия решений;
- Блок выполнения действий по истечении определенного времени, в дальнейшем – блок управления таймером и ЧРВ.

В таблице 2.1 показано соответствие между блоками внутренней структуры программной системы и модулями языка С.

Таблица 2.1. Соответствие между блоками программы и модулями языка С

Название блока	Название программного модуля	Название заголовочного файла
Основной блок	main.c	main.h
Блок управления памятью данных	memory.c	memory.h
Блок управления обменом с ТК	xchg_tc.c	xchg_tc.h
Блок управления обменом с УК	xchg_cmp.c	xchg_cmp.h
Блок ввода и индикации	kbd_ind.c	kbd_ind.h
Блок принятия решений	decide.c	decide.h
Блок управления таймером и ЧРВ	timer.c	timer.h
Объявления нестандартных типов данных	-	dtypes.h

В заголовочном файле **dtypes.h** описаны следующие нестандартные типы данных:

- Булев тип (переменные такого типа могут принимать два значения: 0 (ложь) или 1 (истина)):

```
typedef struct boolean_ {
    unsigned int b: 1; //всего 1 бит
} boolean;
```

- Трехзначный тип (переменные такого типа могут принимать только три значения: -1, 0 и 1):

```
typedef struct three_ {
    signed int v: 2; //всего два бита
} three;
```

Раскроем теперь взаимосвязь между отдельными блоками программной системы (см. графическую часть ДП, лист 6, прямоугольники соответствуют отдельным блокам, а линии – информационным и управляющим связям между блоками; после названий блоков в скобках указаны номера блоков для дальнейших ссылок).

2.3.1. Взаимосвязь между блоками. Информационные и управляющие потоки между блоками программной системы

Как отмечалось выше, программная система может быть представлена в виде семиблочной структуры. При этом информация и сигналы поступают в программную систему не через все блоки. Так, элементы «Блок принятия решений (3)» и «Блок управления памятью данных (1)» ни получают, ни отправляют информации и управляющих воздействий во внешнюю по отношению к программной системе среду. Таким образом, их можно считать внутренними. Остальные блоки так или иначе обмениваются сигналами и (или) информацией с окружением программной системы.

Проследим теперь, каким образом в программной системе с **данной структурой** реализуется упомянутое в самом начале п. 2.1 отображение $F[X]$ относительно каждого из блоков.

Для единообразия будем рассматривать ситуации, когда

а) входные сигналы (входная информация) блока приводят к изменению его внутреннего состояния

либо

б) входные сигналы (входная информация) блока приводят к изменению одного либо нескольких выходных сигналов (выходной информации) блока

либо

в) выходные сигналы (выходная информация) блока влекут появление определенных сигналов (информации) на входах данного блока.

2.3.1.1. Взаимосвязь блока 2 с другими блоками

Пересброс контроллера либо включение питания влекут за собой вызов функции `main()` основного блока. Так начинается работа контроллера под управлением проектируемой программной системы. Старт (включение питания) либо пересброс приводят к запуску процесса программной инициализации контроллера. При этом *на выходах* блока появляется информация, связанная с:

- вызовом функции распределения памяти (входная управляющая информация для блока 1);
- установкой будильника 1 (будильник – программный счетчик времени ожидания) и его последующей проверкой в целях обеспечения задержки перед программной инициализацией микросхемы SE210. Используются вызовы функций блока 5;
- вызовом функции инициализации микросхемы SE210 связи контроллера пульта вахтера с панелью управления пульта вахтера (см. графическую часть ДП, лист 2) – входная управляющая информация для блока 7;
- инициализацией часов реального времени, для чего используется два вызова функции записи в ЧРВ (входная информация для блока 5);
- тестированием компонентов контроллера (ОЗУ, ПЗУ, SE210), для чего вызываются функции чтения байта из ПЗУ, записи байта в ОЗУ и чтения байта из ОЗУ блока управления данных (1). При тестировании ОЗУ (ПЗУ) в ответ на запросы чтения от блока 1 (как значение функции) передается содержимое байта ячейки ОЗУ или ПЗУ, адрес которой был ранее указан в качестве одного из аргументов при вызове функции. Заметим, что тестирование производится в специальном режиме тестирования, вывод о

нахождении в котором можно сделать путем считывания конфигурации с dip- переключателей контроллера;

- вызовом функции запроса даты (блока 5), сопровождающимся получением месяца и дня (для определения, является ли текущий день праздничным);
- вызовом функции анализа пакета от УК (входная информация для блока 4) – в фоновом процессе;
- вызовом функции анализа пакета от ТК (входная информация для блока б) – в фоновом процессе;
- вызовом функции поллинга ТК (входная информация для блока б) – в фоновом процессе;
- вызовом функции создания массива кодов пропусков по получении последнего пакета с командой «Загрузить БД пользователей» - в фоновом процессе.

2.3.1.2. Взаимосвязь блока 7 с другими блоками

Входной сигнал прерывания по нажатию клавиши влечет вызов обработчика данного прерывания `InitReadI2C()`, относящегося к блоку ввода и индикации. В свою очередь этот же блок, реагируя на сигнал, вырабатывает в соответствии с временной диаграммой работы шины I²C (см. приложение ПЗ) в нужные моменты времени *выходные сигналы и данные*: старт-условие (s), биты байтов пересылаемых пакетов (B), синхроимпульсы (отмечают каждый из девяти тактов приема/отправления байта пакета) Sc, стоп-условие (p). На каждый отправленный байт, кроме последнего, ожидается *входная информация – подтверждение приема Ask* (см. приложение ПЗ). После инициирования операции чтения принимается *входной байт кода клавиши*. Результатом обработки байта кода нажатой клавиши в целом являются следующие данные и сигналы:

- *выходная информация* - измененные биты байта состояния поллинга соответствующего ТК;
- выдача *выходных* {s, B, Sc, p} в нужные моменты времени, ожидание и получение *входной* управляющей информации *Ask* для формирования пакетов команд выдачи звука или включения соответствующей клавише индикации.

Входной управляющий сигнал «вызов функции инициализации микросхемы CE210» блока 7 также приводит к многократному появлению выходных сигналов {s, B, Sc, p} и ожидания *Ask* в соответствии с временной диаграммой. Это связано с посылкой пяти четырехбайтных (включая байт адреса CE210) пакетов инициализации микросхемы CE210 (см. п. 1.2.1.4).

Входные управляющие сигналы типа «Запрос на индикацию» и «Запрос на выдачу звукового сигнала» блока 7 также влекут за собой выполнение команд, которое сводится к пересылке на CE210 пакетов различной длины (см. п. 1.2.1.4), т.е. генерации выходных сигналов {s, B, Sc, p}. Управляющая информация *Ask* ожидается после получения байтов от CE210.

Заметим также, что в целях предотвращения ситуаций, в которых возникает бесконечное ожидание (например, ожидание готовности CE210 продолжить операцию обмена, ожидание подтверждения), введены таймауты (ограничения по времени на ожидание). Для этого блок 7 посылает на блок 1 запросы установки и проверки будильников (счетчиков времени), проверяя состояние будильников. *Выходная информация:* запрос на установку будильников, номер будильника (на чертеже совмещены). *Выходная информация:* запрос на проверку будильника, номер будильника; соответствующая ей *входная* - состояние будильника (на чертеже совмещены).

2.3.1.3. Взаимосвязь блока 4 с другими блоками

Входная информация: байты пакета и подтверждения от управляющего компьютера. Передаются в соответствии с протоколом обмена «АЛГО-421.УК-ПВ» (см. п. 1.2.2). Эта входная информация приводит к изменению значений переменных состояния блока (так, байты получаемого пакета приводят к изменению содержимого буфера обмена с управляющим компьютером `comp_buf`, а получение подтверждения `$06` приводит к установлению соответствующей переменной состояния, используемой для определения, какое из событий, накопленных в памяти, послать следующим (если не было получено предыдущее событие, то оно отправляется повторно)).

Входная информация (управление): вызов функции анализа пакета от управляющего компьютера. Приводит в результате интерпретации функции, кода команды и данных (если имеются) пакета (он содержится в буфере) к появлению выходной информации – пакета, пересылаемого на управляющий компьютер, а также выходной информации (управления) «Вызов отправки пакета на ТК» (в случае, если полученная команда подлежит пересылке на ТК, - команда «Пропустить»).

Выходная информация: установка будильников для отслеживания таймаутов обмена с управляющим компьютером (ожидания символа `$06` – подтверждения получения события управляющим компьютером), а также обеспечения минимальных задержек по времени между получением и отправлением байта подтверждения и пакетов (см. приложение П7), обеспечения защиты от получения информации со стороны УК раньше положенных моментов времени; проверка будильников (в ответ на вызов получается от блока 5 входная информация: состояние будильника); обнуление будильника по получении подтверждения раньше таймаута (в сторону блока 5 передается управляющая информация и номер будильника). Одна из переменных блока `Received06` – признак того, что был получен байт

подтверждения раньше достижения таймаута. Один из будильников используется для контроля таймаута на получение очередного пакета от УК с записями БД пользователей в процессе инициализации контроллера (после достижения таймаута вызывается функция создания индексного массива CreateIndex из фонового процесса модуля 2).

2.3.1.4. Взаимосвязь блока 6 с другими блоками

Входная информация: байты пакета от ТК. Передаются в соответствии с протоколом обмена «АЛГО-421.ПВ-ТК» (см. п. 1.2.3). Эта входная информация приводит к изменению значений переменных состояния блока 6 (к изменению содержимого буфера обмена ТК tc_buf; если получены команды \$F1 или \$F6, требующие пересылки команд не по поллингу, то и к установлению флага блокирования поллинга до отправки команд ответа на \$F1 или \$F6 в сторону ТК).

Входная информация (управление): вызов посылки пакета (команда «Пропустить» или «Не пропускать»). Пересылка команды от компьютера в формате протокола АЛГО-421.ПВ-ТК (см. п. 1.2.3). Приводит к появлению выходной информации – пакета для ТК.

Входная информация (управление): вызов функции анализа пакета от ТК. Если получена команда «Запрос на вход (\$F1)» (это означает, что пользователь ввел код пропуска и (возможно) пин-код и ожидает принятия решения системой), блок 6 инициирует *запрос даты и времени* у блока 5 (получая в ответ текущие месяц Month, день месяца Day, день недели Day_of_Week, час Hour и минуту Minute), после чего инициирует *выходную управляющую информацию* – запрос на идентификацию пользователя, адресуемый блоку 3. В процессе идентификации блок 3 использует полученную ранее информацию о дате и времени. В ответ на этот запрос блок 3 посылает блоку 6 *входную информацию* - результаты идентификации пользователя (более подробно об этом см. ниже, при описании функций

блока 3, п. 2.3.4). В зависимости от этих результатов блок 6 определяет тип произошедшего события, помещает текущее событие в память. Кроме этого, блок 6 в зависимости от результатов идентификации определяет, следует ли переслать команду на ТК, и если да, то с каким кодом. В этом случае *на выходе* блока появляется информация – пакет для ТК.

Входная информация (управление): вызов функции поллинга ТК. В случае, если поллинг разрешен, приводит к анализу возможности поллинга для каждого из терминальных контроллеров в соответствии с алгоритмом поллинга (см. п. 1.2), и если находится ТК, подлежащий поллингу, на выходе блока появляется однобайтный пакет для ТК. В ходе поллинга на ТК переслается байт данных, отдельные биты которого изменяются в блоке 7 по получении кода клавиши панели управления пульта вахтера.

Выходная информация: установка будильника для отслеживания таймаута обмена с ТК (ожидания ответа на поллинг); проверка будильников (в ответ на вызов получается от блока 5 входная информация: состояние будильника); обнуление будильника по получении ответа на поллинг раньше достижения таймаута (в сторону блока 5 передается управляющая информация и номер будильника). Признаки получения ответа по каждому из ТК хранятся в переменных состояния блока – массиве ReceivedTCAnswer.

2.3.1.5. Взаимосвязь блока 5 с другими блоками

Входная информация: вызов функции установки времени (поступает от блока 4). *Результат:* многократный вызов функции записи в часы РВ для отправки на ЧРВ команд установки текущих года, месяца, дня месяца, дня недели, часа, минуты и секунды. Это приводит к появлению информации *на выходе* блока (обмен данными с ЧРВ).

Входная информация: сигнал прерывания от таймера (появляется каждые четыре миллисекунды). Это сопровождается вызовом обработчика прерывания `int_timer`, что приводит к модификации значений внутренних

переменных состояния блока (значений времен ожидания будильников), а также появлению *выходной* управляющей информации вызова фонового процесса (данная информация является входной для блока 2).

Входная информация: запись в ЧРВ. Приводит к появлению информации *на выходе* блока (обмен данными с ЧРВ).

2.3.1.6. Взаимосвязь блока 1 с другими блоками

Входная информация (управляющая): вызов функции распределения памяти. *Результат:* изменение переменных состояния (присвоение значений границам функциональных областей памяти), о распределении памяти см. в п. 2.3.2.1.

Входная информация (управляющая): вызов одного из макросов переключения страницы. *Результат:* изменение значений переменных состояния программы (регистров переключения страниц), а также изменение *выходной* информации – номера страницы, смещения в странице (на чертеже не показано).

Ниже будут приведены описания некоторых блоков с точки зрения их внутреннего содержания, то есть функций, а также связей между функциями. Вначале рассмотрим основные структуры данных, используемые в программе, а также блок управления данными.

2.3.2. Структуры данных

Основные структуры данных, используемых программной системой, изображены в графической части дипломного проекта, лист 5. Основными используемыми структурами данных являются следующие:

- база данных пользователей (включает две таблицы: основную и резервную) – до 100 тыс. записей в основной таблице;
- список привилегированных кодов пропуска;
- индексный массив кодов пропуска;
- список соответствия уровней доступа временным зонам;
- список временных интервалов;
- структуры временных интервалов, дня определения временного интервала; список праздничных дней и их структура;
- база (память) событий;
- байты состояния (поллинга ТК и ответов на поллинг).

2.3.2.1. Распределение памяти данных контроллера

Всю используемую память данных контроллера можно разделить на 3 группы:

- Память, занимаемая **переменными**, используемыми программой, статическими массивами данных, а также отводимой под стек (компилятор размещает статические переменные по адресам $\$F0000 \dots \$F6FFF$) – всего $\$F7000 - \$F0000 = 28672$ байт;
- Память, занимаемая **основной и вспомогательной таблицами справочной БД**, загружаемых с управляющего компьютера;
- Память, отводимая под **события**, поступающие от ТК.

Вся память данных контроллера может быть представлена в виде последовательности блоков (страниц) размером 4096 байт каждая.

Таблицы базы данных пользователей

Согласно техническому заданию на дипломный проект, необходимо обеспечить **максимальную** (фактическая может быть и меньшей!) численность работающих на предприятии **100 тысяч** человек. Под каждого проходящего отводится одна строчка справочной таблицы БД размером в 5 или 8 байт (**3 байта** – код карточки; **1 байт** – номер уровня доступа, соответствующего одному или нескольким временным интервалам; **1 байт** – признак прохода (пользователь не входил, вошел и не вышел, вышел); в случае присутствия клавиатуры ввода и использования так называемого пин-кода (для повышения безопасности на объектах строгого режима) на проходе могут дополнительно использоваться **3 байта пин-кода**), следовательно, необходимо иметь **максимум 500000** или **800000** байт памяти, чтобы разместить данную таблицу (**122** страницы+**288** байт или **195** страниц+**1280** байт).

В основную таблицу должны помещаться записи, отсортированные по коду пропуска, именно по данной таблице строится индексный массив кодов пропусков для ускорения поиска. Загрузка данной таблицы с управляющего компьютера должна производиться один раз в сутки. Помимо основной таблицы БД, была введена и вспомогательная, в которую будут помещаться неотсортированные по коду пропуска записи, поступающие с управляющего компьютера в те моменты времени, когда, например, был динамически добавлен новый сотрудник в основную базу данных АССаД32. Размер вспомогательной таблицы задается в долях от размера основной таблицы ReserveFactor (например, ReserveFactor = 0,05).

Статическая память

За данной областью памяти аппаратно закреплены адреса \$F0000...\$F7000, то есть ее размер составляет 28672 байт, но практически под внутренние регистры и прочие нужды отводится определенная часть адресов этой области, так что под определяемые программистом переменные следует отвести область адресов \$2000 до \$7000 (20480 байт). К данным, располагающимся в данной области памяти, можно обращаться без предварительной инициализации регистров управления страничной памятью данных PORTG и DPAGE (их назначение будет пояснено ниже). В терминах компилятора [4] это также означает, что переменные различных типов в данной области памяти можно объявлять без всякого рода дополнительных директив.

Помимо программного стека, данная область памяти содержит, например, следующие данные:

- Индексный массив кодов пропусков `char IndexedCodes[MAX_INDEX*CODE_LEN]`, где размер массива определяется произведением длины кода пропуска `CODE_LEN` (3) на максимальное число кодов пропусков `MAX_INDEX` (определяется как квадратный корень из максимального числа записей основной таблицы БД: $\sqrt{100000} \approx 316$). Таким образом, под индексный массив отводится $316*3 = 948$ байт. Индексный массив помогает заметно ускорить процесс поиска записи по коду пропуска (см. п.);
- список соответствия уровней доступа временным зонам `char AccLev_TZone[MAX_ACCLEV_CNT]` (номер элемента списка является номером уровня доступа, а значение элемента списка – номером временной зоны). `MAX_ACCLEV_CNT` – максимальное число уровней доступа (256). Итак, под данный список отводится

256 байт. В случае, если константа `DIR_SENSITIVE = 1`, вместо одного вводится два списка соответствия (на вход и на выход);

- список временных зон
`struct T_zone TimeZones[MAX_ZONE_CNT-2]`, где
`MAX_ZONE_CNT = 256` – наибольшее число временных зон.
 Временные зоны 0 и 1 не подлежат переопределению (зона «0» - «проход не разрешен ни в какое время»; зона «1» - «проход разрешен всегда»). Так как каждая временная зона представляется восьмью временными интервалами, а каждый временной интервал – пятью байтами, то под список временных зон отводится $5*8*256 = 10240$ байт;
- список привилегированных кодов пропуска (8 кодов по 3 байта каждый, всего 24 байта);
- список праздничных дней (всего до 32-х праздников, каждый из которых представляется месяцем и днем месяца (2 байта); итого $32*2=64$ байта);
- карта терминальных контроллеров (по биту на каждый ТК) – 5 бит;
- хранимые в памяти аналоги байтов состояния, пересылаемого на ТК во время поллинга (опроса) `tcPollSTATE_toTC[MAX_TC_CNT]`, где `MAX_TC_CNT` – максимальное число терминальных контроллеров (турникетов); всего 5 байт;
- хранимые в памяти аналоги байтов состояния, получаемых от ТК в ответ на поллинг (опрос) `tcPollSTATE_toKPV[MAX_TC_CNT]`, всего 5 байт;
- «видеопамять», состояния индикаторов `IndicsData[MAX_TC_CNT*2]`, всего 10 байт;
- прочие (служебные) переменные.

Память событий ТК

Можно выделить события ТК, связанные с проходом, и прочие события ТК. События должны по запросу отправляться на УК и накапливаться в памяти событий, размер которой определяется как разность общего объема памяти данных и числа байт, отводимых под статическую память и память таблиц БД пользователей (минимальный размер в случае отсутствия пин-кода: $1048576 - (28672 + 500000 + 0,05 * 500000) = 494904$ байта).

В приводимой ниже таблице наглядно представлены области памяти данных контроллера с указанием констант и переменных, задающих границы областей.

Таблица 2.2. Распределение памяти данных

Название блока памяти данных	Адреса блоков памяти (включая границы)
Статическая память	MemHoleOrig...MemHoleEnd (\$F0000...\$F2000)
Основная таблица БД пользователей	StartCDBAdr...StartRDBAdr-1 (\$00001...\$00001+USER_CNT*SizeOfUser)
Вспомогательная таблица БД пользователей	StartRDBAdr...StartTCEAdr-1 (StartRDBAdr... StartRDBAdr + +USER_CNT*SizeOfUser*ReserveFactor)
Память текущих событий ТК	StartTCEAdr... MEM_END (MEM_END = \$FFFFFF)

Распределение памяти реализовано в функции AllocateMemory (модуль memory.c).

2.3.2.2. Особенности страничной адресации памяти

Для обращения к той или иной странице памяти необходимо проинициализировать следующие регистры микроконтроллера MC68HC812A4 [1]:

- DPAGE – этот регистр должен содержать номер страницы, или, что то же самое, биты 12...19 адреса (соответственно биты 0...7 регистра DPAGE);
- PORTG – регистр, биты 0...3 которого соответствуют битам 16...19 адреса, а бит 4 устанавливается в единицу, если только задействована линия 20 шины адреса. Остальные (старшие) биты в данной конфигурации должны равняться нулю.

Заметим, что линии ADDR0 и ADDR20 микроконтроллера используются для формирования сигналов CS1 и CS2 выборки нужной микросхемы памяти.

Как видно, значения для записи в указанные выше регистры могут быть получены из полного адреса байта путем выполнения соответствующих сдвигов и выделения нужных частей.

Охарактеризуем назначение макроопределений для работы со страничной памятью, описанных в файле memopy.h:

- GetPageNo(adr) – получение в служебной переменной page битов 12...20 адреса adr;
- GetOffsetNo(adr) – получение в переменной offset смещения байта относительно начала страницы;
- PageSw(page) – переключение на страницу page путем инициализации соответствующих регистров;
- SwitchPage(adr) – включает в себя три описанных выше макроса;
- SwitchPageNew(adr, size) – то же, но с проверкой на конец страницы: вычисляется значение offset+size (где size – размер

выделяемой памяти), и если оно превышает 4096, то переключение производится на следующую страницу;

- `PrewriteEvent(adr_, size_)` – макрос, переключающий страницу в памяти событий перед записью очередного события, где `size_` - размер события. Учитывает особенности распределения памяти событий, ее кольцевую организацию (в случае, если достигнут конец памяти событий, очередное событие запишется в начало). `ev_area_end_marker` – признак того, что до конца памяти событий никаких событий не располагается, так как последнее событие в эту область не поместилось;
- `PrereadEvent (adr_, size_)` - макрос, переключающий страницу в памяти событий перед чтением очередного события. Имеется анализ присутствия маркера блока и границы памяти данных `MEM_END`.

2.3.3. Основной блок программной системы `main`

2.3.3.1. Основные положения

Блок предназначается для инициализации регистров микроконтроллера (определяющих режимы работы, разрешающих или запрещающих прерывания и т.д.) в самом начале исполнения программы, а также инициализации и тестирования отдельных компонентов контроллера. Текст кода, относящегося к данному блоку, приводится в приложении П8.1.

2.3.3.2. Структура блока. Взаимосвязь между функциями

Структурная схема основного блока приведена на листе 7 графической части дипломного проекта.

Основной функцией блока и программной системы является **функция main**, именно с нее начинается работа контроллера. В ходе выполнения функции main производится вызов следующих функций блока:

- функции `AnalyseConfiguration` анализа конфигурации контроллера (конфигурация считывается с dip- переключателей посредством аппаратного блока считывания конфигурации контроллера, см. лист 2 графической части) – возвращается адрес контроллера, режим работы контроллера (например, тестовый режим либо основной);
- функции `ResetSVControl` сброса супервизора контроллера (супервизор необходимо сбрасывать не реже раза в 1 сек., в противном случае он сгенерирует сигнал сброса микроконтроллера);
- функции `ini_portNJ` инициализации портов N и J микроконтроллера, ответственных за передачу и прием сигналов связи с микросхемой CE210;
- функции `CheckHoliday` проверки того, является ли текущий день праздничным (нерабочим) – полученная информация далее используется в процессе принятия решений о разрешении проходов.

Функция `CheckHoliday` вызывается в основном режиме работы. В тестовом же режиме из функции `test_`, являющейся вложенной в функцию `main`, вызываются следующие функции блока:

- `test_RS485` – функция тестирования схемы интерфейса и линий 0 и 1 последовательного интерфейса RS-485 (обмена с УК и ТК соответственно*);
- `test_ROM` – функция тестирования ПЗУ контроллера;

* Перед тестированием выходные линии данных обеих интерфейсных схем (D+0 и D+1, D-0 и D-1, см. приложение П9) должны быть физически объединены

- test_RAM – функция тестирования ОЗУ контроллера.
- TestCE210 - функция тестирования микросхемы CE210 контроллера.

Кроме того, из функции main вызываются функции и макроопределения, *относящиеся к другим блокам программной системы:*

- функции инициализации внутреннего таймера микроконтроллера ini_real_timer (функция блока timer);
- функции установки setalarm и проверки tstalarm истечения времени будильника (эти функции определены в блоке timer);
- макроопределения запрещения (DisableInterrupts) и разрешения (EnableInterrupts) прерываний микроконтроллера (определены в одной из стандартных библиотек hodef.h, поставляемых разработчиком компилятора [4] – фирмой HIWARE);
- функции write_data_time_clock записи в микросхему часов команды и данных (для инициализации часов);
- функции инициализации (InitCE210) микросхемы CE210 управления вводом сигналов ручного управления и индикацией.

Функция фонового процесса **main_fon** вызывается из блока timer (управление на нее передается из обработчика прерывания по внутреннему таймеру, вызываемому *один раз в 4 миллисекунды*). Выполнение данной функции может быть на время приостановлено при возникновении любого прерывания. В процессе выполнения функции вызываются (перед вызовом функций проверяются флажки, следует ли вызывать ту или иную функцию) следующие функции:

- функция Handle_comp_whole_package (относящуюся к блоку xchg_comp) анализа поступившего от управляющего компьютера пакета, выполнения команды пакета и ответа на него;

- функция `Handle_tc_whole_package` (относящуюся к блоку `xchg_tc`) анализа поступившего от одного из ТК пакета, выполнения команды пакета и ответа на него (если требуется);
- функция `CreateIndex` создания индексного массива (данная функция относится к блоку `decide`);
- функция `PollTCs` поллинга (периодического очередного опроса) ТК (подробнее о поллинге см. в п. 2.3.1.4, п. 1.2.3);
- функция `tstalarm` (блока `timer`) проверки состояния заведенного ранее будильника для определения момента создания индексного массива.

Все функции тестирования производят вызов функций `printf_` и `sprintf` для вывода строк с сообщениями для настройщика контроллера. Кроме того, функции тестирования ОЗУ (`test_RAM`) и ПЗУ (`test_ROM`) вызывают функцию чтения символа (`getchar_no_wait`), вводимого настройщиком с консоли управляющего компьютера.

Функции вывода байта в линии 0 и 1 (`putchar_rs_0` и `putchar_rs_1`), а также получения байта с этих линий вызываются из функции тестирования `test_RS485`.

Наконец, функции `test_ROM` и `test_RAM` производят вызов функций блока `memory`, обеспечивающих считывание или запись слов памяти по указываемым адресам. Это соответственно функция `read_char_ROM` для `test_ROM` (в нее передается адрес считываемого из ПЗУ байта, возвращается считанное значение); функции `write_char` и `read_char`, вызываемые из функции тестирования ОЗУ `test_RAM`. В `read_char` передается адрес считываемого из ОЗУ байта (возвращается значение байта), в `write_char` передается значение и адрес записываемого байта.

Тестирование RS-485 основано на том, что одна из интерфейсных микросхем передает, а другая – принимает данные. Отправленное и полученное значение сравниваются.

Тестирование ПЗУ основано на сверке контрольной суммы (КС), подсчитанной по всем ячейкам, со значением КС, записанным единственный раз в одной из свободных ячеек сразу после прошивки микросхемы с помощью специальной программы¹.

Тестирование ОЗУ основано на записи в ячейки оперативной памяти значений \$55 (01010101) и проверке записанного и считываемого значений по каждой из ячеек.

2.3.4. Блок принятия решений *decide*

2.3.4.1. Основные положения

Блок содержит функции поиска в таблицах базы данных пользователей, располагающейся в памяти данных контроллера, а также принятия решения по проходу и фиксации типа события, связанного с проходом. Текст кода, относящегося к данному блоку, приводится в приложении П8.2.

2.3.4.2. Структура блока. Взаимосвязь между функциями

Структурная схема блока «Блок принятия решений» приведена на листе 8 графической части дипломного проекта.

Как видно из структурной схемы, данный блок представлен четырьмя функциями и четырьмя макроопределениями (макросами). С логической точки зрения разницы между макросами и функциями не существует, хотя способ их подключения к блоку программной системы в процессе компиляции различается: функция вызывается путем передачи на нее управления и занесения в стек необходимых параметров, а макрос просто вставляется в нужный участок программы с подставлением вместо

¹ Данная программа является универсальной и была разработана специалистами фирмы «Алгонт»

параметров формальных - параметров вызова. В дальнейшем будем использовать слово «Вызов» как для функций, так и для макроопределений.

Функция создания индексного массива **CreateIndex** *вызывается* из блока main (из функции фонового процесса main_fon) после того, как была загружена последняя запись БД пользователей в память контроллера в процессе его инициализации. В качестве *аргументов* функции можно рассматривать следующие константы и глобальные переменные: StartCDBAdr (константа) – адрес первого байта первой записи БД; LastOrigCDBAdr – адрес первого байта последней загруженной записи БД; USER_CNT – наибольшее число пользователей в данной реализации системы; SizeOfUser – размер записи БД. Функция приводит к заполнению индексного массива кодов пропусков IndexedCodes. В ходе своего выполнения функция вызывает макрос добавления элемента в индексный массив AddToIndex, в качестве *аргументов* которого можно рассматривать адрес adr записи индексируемой БД и номер добавляемого элемента индексного массива index_itemno. В свою очередь, для чтения кода пропуска из памяти макрос AddToIndex использует макросы переключения страниц памяти SwitchPage (получить номер страницы, смещения и переключить страницу по ее адресу) и PageSw (переключить страницу по ее номеру).

Для принятия решения о проходе пользователя и опознания возникшего события используется функция **IdentifyUser**. Она вызывается из блока обмена с ТК xchg_tc (из функции Handle_tc_whole_package обработки поступившего командного пакета). *Аргументами* функции являются: (1) время прохода (текущие месяц Month, день месяца Day, день недели Day_Of_Week, час Hour, минута Minute); (2) поступивший код пропуска, направление прохода, признак наличия пин-кода, пин-код (если присутствует) – байты буфера tc_buf; (3) признак необходимости учета пин-кода PinCodPresent, список привилегированных кодов PrivilCodes, список соответствия между уровнями доступа и временными зонами

AccLev_TZone, описания временных зон TimeZones, признак праздничного дня TodayIsHoliday. *Возвращаемое значение:* результат идентификации пользователя (в настоящей версии ПО реализовано получение семи различных результатов идентификации, они описаны в decide.h и имеют префикс *ir*: *irNotTime* – проход не в положенное время; *irOK* – проход может быть разрешен ; *irDupIn* – попытка повторного входа; *irDupOut* – попытка повторного выхода; *irNoUser* – пользователь не найден в БД контроллера; *irWrongPin* – введен неверный пин-код; *irNoPin* – не введен пин-код). Для определения наличия поступившего пин-кода в БД контроллера и (в случае наличия) получения адреса записи для дальнейшего анализа вызывается функция этого же блока FindCode. Кроме того, в процессе принятия решения используется вызов макроса сравнения кодов Compare (передаются байты сравниваемых пин-кодов, возвращается результат сравнения CmpRes), а также макроса чтения трехбайтного кода из памяти GetCodeVals (передается адрес записи FoundAdr, возвращаются байты пин-кода пропуска dbcode_byte1...dbcode_byte3). Также используется вызов макроса SwitchPage блока memory для переключения страницы памяти (на схеме вызов не показан).

В свою очередь, функция **FindCode** поиска кода пользователя в БД контроллера вызывает функцию **SearchInReserve** для поиска кода пропуска в резервной таблице БД в том случае, если запись не найдена в основной таблице (передаются байты цифр кода пропуска Key1_2 (цифры 1 и 2)...Key5_6 (цифры 5 и 6); возвращается адрес искомой записи или 0, если запись не найдена). Обе функции используют вызовы макросов Compare (для сравнения кодов пропусков), GetCodeVals (для чтения цифр кода пропуска из памяти), а также GetPassByte (для определения того, была ли удалена запись из памяти). В обоих случаях в макрос Compare передаются байты текущего (Key1_2...Key5_6) и хранящегося в БД (dbcode_byte1...dbcode_byte3) кодов пропусков, а возвращается результат

сравнения CmpRes. В макрос GetCodeVals передается адрес текущей записи adr, а возвращаются байты с цифрами кода пропуска dbcode_byte1...dbcode_byte3. В макрос GetPassByte передается адрес adr текущей записи, а возвращается значение value считанного байта признака повторного прохода, комбинация \$FC которого означает, что запись отсутствует в БД (была удалена из нее).

Аргументами функции FindCode являются: SizeOfUser – размер записи БД пользователей; IndexedCodes – индексный массив кодов пропусков; ICnt – количество элементов данного массива; StartCDBAdr – адрес первой записи основной таблицы БД пользователей. *Аргументами* функции SearchInReserve являются: SizeOfUser; StartRDBAdr - адрес первой записи вспомогательной таблицы БД пользователей; StartTCEAdr - адрес первой записи памяти событий ТК (для определения адреса последней записи резервной таблицы БД пользователей). Возвращаемыми значениями функций FindCode и SearchInReserve являются адрес найденной записи (или «0», если запись не найдена), а также значение смещения offset найденной записи относительно начала переключенной страницы памяти.

Наконец отметим, что макросы AddToIndex, GetPassByte и GetCodeVals для переключения страницы памяти вызывают макросы SwitchPage и PageSw блока memory.

2.3.4.3. Описание алгоритма поиска и функции

поиска по коду пропуска в БД контроллера

FindCode (char Key1_2, char Key3_4, char Key5_6)

Условно в алгоритме поиска можно выделить 4 этапа, один подготовительный и три основных.

Далее в данном разделе при описании алгоритма и функции в скобках будем указывать номера блоков схемы алгоритма (см. графическую часть ДП, лист 10).

В ходе подготовительного этапа (2) производится инициализация некоторых управляющих переменных функции (устанавливается в единицу признак **FoundImplicit** – признак того, что запись присутствует в основной таблице БД; вычисляется значение коэффициента (**IndToDBAdrScale**) перевода номера элемента индексного массива в адрес записи таблицы БД относительно начала основной таблицы БД).

Первый этап поиска

В ходе первого этапа производится поиск кода пропуска в индексном массиве методом золотого сечения (каждый раз интервал поиска уменьшается в два раза). Следует отметить, что вначале сравнение производится с граничными значениями индексного массива (блоки 3, 4, 10, 11, 5) для того, чтобы узнать, имеется ли вообще запись с искомым кодом в основной таблице (напомним, что в индексном массиве обязательно должны присутствовать коды самой первой и самой последней записи основной таблицы). В случае, если запись не найдена в индексном массиве, это означает, что она отсутствует и в основной таблице базы данных. В этом случае вызывается функция поиска в резервной таблице БД **SearchInReserve** (блок 12), т.е. производится переход к *третьему этапу*

поиска. В противном случае производится инициализация индексов текущих левой и правой границ поиска (7) и начинает выполняться цикл поиска (блоки 8, 9, 18, 19, 24).

В теле данного цикла определяется текущий номер индекса середины отрезка поиска MidNo (блок 8), а также производится сравнение ключа поиска со значением, соответствующим середине текущего отрезка поиска, т.е. медианой (блоки 9, 18). Здесь возможны три исхода: *совпадение значений ключа и медианы* (в этом случае в блоке 14 вычисляется адрес искомой записи, после чего в блоках 40 – 45 проверяется ее фактическое присутствие в БД и возвращается результат); *значение ключа меньше значения медианы* (в этом случае текущая левая граница становится медианой, блок 19); *значение ключа больше значения медианы* (в этом случае текущая правая граница становится медианой, блок 24).

Цикл поиска выполняется до тех пор, пока расстояние между номерами левой (left) и правой (right) границ текущего отрезка поиска превышает длину кода пропуска CODE_LEN (блок сравнения 15). Таким образом, по завершении поиска остается проанализировать еще один код пропуска, соответствующий одной из текущих границ (номер которой отличен от номера медианы MidNo, полученного на последней итерации) (блоки 30, 31, 32, 35, 36).

В процессе поиска используется макрос сравнения трехбайтных кодов вида Compare(N1_1, N1_2, N1_3, N2_1, N2_2, N2_3), где первые три и последние три параметра – сравниваемые коды. Результаты сравнения **crEqual:** N1=N2; **crLess:** N1<N2; **crMore:** N1>N2. Данный макрос вызывается в блоках 3, 9, 10, 22, 29, 31, 35.

В том случае, если *ключ поиска совпадает с одним из элементов индексного массива* (положительный результат сравнения в блоке 6, а также в блоке 32 или 36), вычисляется значение адреса искомой записи в основной таблице БД на основании номера элемента индексного массива и

коэффициента перевода IndToDBAdrScale (блок 34 или 37), после чего производится вызов макроса **GetPassByte** для определения фактического наличия записи в таблице (если запись была динамически удалена, четвертый байт ее структуры будет иметь шестнадцатеричное значение \$FC) (блок 40, сравнение в блоке 41), после чего возвращается значение 0 (если запись была удалена) (42, 45) или адрес найденной записи adr (44, 45).

Второй этап поиска

В том случае, если поиск произведен успешно (при этом FoundImplicit.b = 1 (положительный результат сравнения в блоке 6, но отрицательный – в блоке 32 или 36)), но ключ поиска не совпадает непосредственно ни с одним из элементов индексного массива (т.е. выявлено, что искомый код располагается в базе между кодами, соответствующими i-ому и (i+1)-ому элементам индексного массива), то дальнейший поиск производится уже в основной таблице базы данных на нужном интервале адресов с тем, чтобы определить адрес искомой записи (блоки 46, 20, 21, 22, 16, 17, 29, 25, 26, 27, 33, 28, 29, 38, 43). Цикл поиска выполняется до тех пор, пока адрес правой границы поиска превышает адрес левой хотя бы на 1 байт (сравнение – в блоке 25). По окончании поиска определяется адрес оставшейся непросмотренной записи таблицы на интервале поиска (26, 27, 38), после чего ее значение сравнивается с ключом (38). В случае, если запись найдена (в цикле или после него), проверяется ее фактическое присутствие в таблице и возвращается результат (блоки 40 - 45).

Макрос **GetCodeVals**(adr,dbcode_p0,dbcode_p1,dbcode_p2) позволяет по адресу adr текущей записи определить код пропуска (параметры dbcode_p0...dbcode_p2), так как записи БД, в отличие от событий, в общем случае не являются выравненными по началу страницы памяти (см. п. 2.3.2). Он вызывается в блоках 21 и 28.

Третий этап поиска

Функция **SearchInReserve** вызывается в том случае, если искомая запись не найдена в основной таблице, хотя численно ее код находится между значениями каких-либо двух элементов индексного массива (блок 39).

Так как резервная таблица неупорядочена и имеет сравнительно небольшой объем, поиск в данном случае реализуется путем последовательного сравнения кодов пропусков с ключом поиска.

2.3.4.4. Описание функции идентификации пользователя **IdentifyUser**

Вначале производится поиск в базе данных путем вызова функции **FindCode** (см. выше). Если запись не найдена, возвращается **irNoUser**, в противном случае анализируются оставшиеся поля найденной записи (поле пин-кода, если необходимо; поле «Признак повторного прохода»* для анализа повторного прохода (если установлен флажок контроля повторного прохода **PASSBACK_CONTROL**)) и возвращаются по необходимости соответствующие значения.

При анализе времени прохода выполняются следующие действия. Проверяется, является ли код пропуска привилегированным (просматривается список **PrivilCodes**), и если да, то возвращается значение **irOK**. В противном случае по уровню доступа определяется временная зона. Производится проверка на зоны 0 («никогда», т.е. озвращается **irNotTime**) и 1 («всегда»). В противном случае анализируются восемь соответствующих временной зоне временных интервалов, для каждого из которых сравниваются день определения интервала (один из дней недели или праздник), а также, попадает ли текущее время в диапазон между временами (час, минута) начала и окончания интервала. Если время прохода

* Имеет три значения: 00 – неопределенное состояние; 01 – «пользователь вошел, но еще не вышел»; 11 – «пользователь вышел, но еще не вошел снова»

соответствует времени протяженности одного из интервалов, то возвращается `irOK`, иначе – `irNotTime`.

Выше были рассмотрены аспекты, связанные с анализом конструкции, структуры программной системы. Однако для того, чтобы устройство (контроллер) могло функционировать под управлением данной программной системы, необходимо выполнить ряд операций, связанных с записью на носитель, тестированием и эксплуатацией программного комплекса, которые будут изложены в технологической части дипломного проекта.

3. Технологическая часть

В рамках данной части дипломного проекта производится разработка документации по записи программного продукта на носитель, его тестированию в условиях единичного и мелкосерийного производства, а также по эксплуатации программного продукта. Единичное и мелкосерийное производство аппаратных и программных составляющих пультов вахтера как компонентов систем безопасности имеет место в силу позаказной системы изготовления и сбыта, практикуемой фирмой «Алгонт» и обусловленной спецификой разрабатываемой и производимой продукции.

Приводимые ниже разделы данной части «Структурная схема рабочего места настройщика контроллера», «Запись программного продукта на носитель» и «Программное тестирование аппаратных компонентов контроллера» имеют особое значение для инженеров-настройщиков и внедренцев в процессе настройки, внедрения, опытной эксплуатации и сопровождения контроллера пульта вахтера.

3.1. Структурная схема рабочего места настройщика контроллера АЛГО-421ПВ

Одна из возможных разновидностей структурной схемы рабочего места настройщика контроллера АЛГО-421ПВ приведена в графической части дипломного проекта на листе 11.

Можно выделить *объект* настройки и тестирования и *субъект* настройки и тестирования. *Объектом* в данном случае будет являться пульт вахтера (точнее, его контроллер и панель управления), а *субъектом* – весь комплекс инструментальных (программных и аппаратных) средств, с помощью которых настройщик выполняет технологические операции тестирования и настройки.

К составным частям субъекта тестирования можно отнести следующие компоненты, изображенные на структурной схеме:

- персональный компьютер (ПК);
- специализированное программное обеспечение, эксплуатируемое на данном ПК (*программатор РАС-DIP40 и диспетчер приборов, включая цифровой электронный осциллограф ЛА-10, реализованный программно*);
- программатор РАС-DIP40 – аппаратная часть;
- универсальный тестер-авометр;
- блок питания, обеспечивающий энергией пульт вахтера (+5В, +12В), а также программатор и компьютер (~220В);
- компоненты, обеспечивающие связь между некоторыми из перечисленных выше блоков - интерфейсная карта программатора, клавиатура персонального компьютера.

Перечислим основные функции компонентов структурной схемы.

Программатор (его аппаратурная и программная компоненты) предназначен для прошивки микросхем ПЗУ новой версией программного обеспечения, поставляемой программистом в процессе отладки и тестирования программного и аппаратурного обеспечения контроллера. Операция программирования микросхем будет описана в разделе 3.2.

Для тестирования контроллера средствами программного обеспечения контроллера используется клавиатура ПК (для обеспечения ввода команд интерактивной связи настройщика с объектом в процессе тестирования). Осведомительная информация отображается на экране монитора компьютера. Операция подобного тестирования описана в разделе 3.3.

В случае, если предусмотренное программным обеспечением контроллера тестирование выявило наличие неисправности, для определения ее вида зачастую настройщику приходится применять дополнительное оборудование (осциллограф, тестер).

Программно реализованный двухлучевой осциллограф предназначен для снятия временных разверток сигналов контроллера на различных участках его электрической схемы с целью их сравнения с эталонными и дальнейшего выявления вида неисправностей.

Универсальный тестер-авометр может быть реализован как программно, так и аппаратно (на схеме рассмотрен случай его аппаратной реализации). Предназначен для определения значений напряжений и силы токов на различных участках электрической схемы контроллера и панели управления пульта вахтера с целью их сравнения с эталонными значениями и дальнейшего выявления вида неисправностей.

Рассмотрим теперь две технологических операции, связанных с тестированием и настройкой, - операцию программирования микросхемы (записи программного продукта на носитель) и операцию программного тестирования аппаратурных компонентов контроллера.

3.2. Руководство по записи программного продукта на носитель

Понятия «запись в микросхему» и «прошивка микросхемы» будем употреблять в качестве синонимов.

3.2.1. Требования к носителю, техническим и программным средствам записи

Программный продукт подлежит записи в ПЗУ с электрическим стиранием AT29C512 фирмы ATMEL.

Файл с программным продуктом для записи, включая его обновленные версии, поставляются разработчиком программного обеспечения.

Для записи используется любой программатор (аппаратное и программное обеспечение), поддерживающий данный тип микросхем. Ниже приводится описание операции записи в ПЗУ с помощью универсального программатора **PAC-DIP40 ALL-07 Universal Programmer & Tester** фирмы HI-LO Systems. Для его использования необходимо наличие персонального компьютера (i80x86) с установленной операционной системой MS-DOS или MS-Windows, с жестким диском (объем которого лимитируется операционной системой, но не должен быть менее 10...15 МБайт), монитором и клавиатурой. Программатор подсоединяется к ПК посредством специальной интерфейсной карты, входящей в комплект поставки программатора и вставляемой в один из ISA- слотов основной платы (main board) ПК. Программное обеспечение программатора (версия 9.2x) представляется в виде совокупности файлов, располагающихся в каталоге C:\ROMNEW.

3.2.2. Пользовательская маркировка микросхем

Маркировка микросхем, указываемая на приклеенной к микросхеме этикетке, должна содержать следующие элементы (построчно):

- букву «Е» в случае, если микросхема должна содержать только четные байты записываемого кода;
- обозначение контроллера (АЛГО-421ПВ);
- версию программного обеспечения;
- дату последней прошивки;
- краткое наименование объекта, на котором планируется установить контроллер, либо надпись «отл.», если микросхема используется для отладочных целей.

Наносимая пользователем маркировка не должна закрывать фабричную.

3.2.3. Подготовка к записи

Перед тем, как произвести запись программного продукта на носитель (носители), необходимо поступить следующим образом.

1. Переписать файл **main.0** в каталог программатора C:\ROMNEW.
2. Включить программатор, если он был выключен, с помощью переключателя, располагающегося на его задней панели.
3. Запустить файл **access.exe** из каталога программатора. После запуска в верхней части экрана появится заголовок вида «PAC-DIP 32/40/48. Device Driver File Menu V9.21» и главное меню под данным заголовком.
4. *Произвести разделение файла main.0 на четные и нечетные байты.* В пункте главного меню **UTILITY** следует выбрать (с помощью клавиш направлений, подтверждение выбора - Enter)

опцию **2-Way Splitter**. Появится диалоговое окно с предложением ввести имя входного файла. Следует набрать *main.0*. Далее появится диалоговое окно с предложением ввести имя выходного файла четного байта (по умолчанию – *main.evn*) и имя выходного файла нечетного байта (по умолчанию – *main.odd*). Следует нажать два раза Enter для того, чтобы дать этим файлам имена по умолчанию. После успешного разделения файла на экран будут выведены строки сообщений: «2 ways splitting now...», «Ok.», «Press any key to return». Далее следует нажать любую клавишу.

5. В пункте меню **DEVICE** следует выбрать подпункт **EEPROM**. Появится список производителей (MFR). В нем следует выбрать строку **ATMEL** (номер 04). Появится список типов микросхем, из которых следует выбрать **AT29C512** (либо ввести «031»). Появится строка вида «Running C:\ROMNEW\EEP1.EXE», после чего загрузится оболочка управления программированием и тестированием данного типа микросхем.

3.2.4. Операции записи и сравнения

Для записи очередной микросхемы следует выполнить следующие действия.

1. Вставить микросхему в колодку программатора. Убедиться в том, что микросхема установлена в колодке правильно (ее ключ должен быть ориентирован в направлении рукоятки защелки колодки программатора). Микросхема должна быть выравнена в колодке по торцу, противоположному ключевому.
2. Закрепить микросхему в колодке программатора. Для этого следует опустить вниз рукоятку защелки возле колодки.

3. Загрузить в буфер программатора один из файлов, полученный в результате разделения, следующим образом. Необходимо нажать клавишу «2» (Load BIN or HEX file to buffer). После этого появится диалоговое окно с предложением ввести имя загружаемого файла. В случае, если микросхема (микросхемы), подлежащая записи, должна содержать четные байты программного кода (обычно такие микросхемы маркируются с указанием буквы E на приклеенной этикетке), необходимо ввести *main.evn*, в противном случае ввести *main.odd*. Появится запрос на ввод типа (формата) файла с программным кодом (in, <I>ntel HEX, <M>otorola S HEX:). Необходимо ввести **B** (двоичный формат). Далее будет предложено ввести адрес, начиная с которого будет загружен файл, по умолчанию – нулевой адрес (Load address (00000):). Следует нажать Enter, так как загрузку любого файла в данном случае следует осуществлять с нулевого адреса. По успешном окончании загрузки файла в буфер появятся сообщения «Loading...Ok! END ADDR.: 00FFFF», «Press any key to continue Or press <CR> to back to main menu». Это означает, что для перехода к следующему этапу технологической операции записи необходимо нажать любую клавишу; для возврата же в главное меню следует нажать клавишу Enter. В том случае, если файл не был загружен успешно (в этом случае появится сообщение об ошибке на красном фоне), следует сделать еще 1...3 попытки по выполнению действий данного пункта. Если и они не увенчаются успехом, то следует обратиться к системному администратору.
4. Для программирования микросхемы далее нажмите клавишу с латинской буквой **P**. Появится вопрос о готовности (Ready to program (Y/Even/Odd/<CR>)?). Для начала операции записи следует нажать клавишу с буквой **Y**. Во время выполнения

операции записи на экран будет выводиться мигающее сообщение «Programming now...». По успешном завершении операции будет выведено сообщение «Ok!», в противном случае – сообщение об ошибке с указанием адреса, начиная с которого возникла ошибка записи. Далее будет автоматически произведен процесс верификации (сравнения контрольной суммы содержимого буфера с контрольной суммой записанной в микросхему информации). При этом на экран будет выводиться мигающее сообщение «Verifying now...». В случае совпадения контрольных сумм будет выведено сообщение «Ok!» (в этом случае следует нажать любую клавишу), в противном случае – предупреждающее сообщение об ошибке верификации. В случае возникновения ошибки записи либо ошибки верификации следует произвести еще 1...2 попытки по выполнению пунктов 1...4. В случае неудачи следует обратиться к системному администратору либо заменить микросхему.

5. Рекомендуется произвести полное сравнение содержимого записанного в микросхему кода с содержимым буфера. Для этого следует нажать клавишу **C** (Compare & Display error). Во время сравнения на экран будет выводиться мигающее сообщение «Comparing now...». В случае полного побайтного совпадения информации в буфере и в микросхеме будет выведено сообщение «Ok!» (в этом случае следует нажать любую клавишу), в противном случае – адреса, по которым информация различается. В случае ошибки сравнения следует попытаться повторить 1...3 раза операции 1 – 4. В случае неудачи следует попытаться заменить микросхему, в случае повторной неудачи – обратиться к системному администратору.

6. Поднять рукоятку колодки вверх, после чего, соблюдая осторожность, извлечь микросхему из колодки программатора.
7. Снять старую этикетку пользовательской маркировки микросхемы. Наклеить новую этикетку. Нанести на нее новую маркировку в соответствии с п. 3.1.2.
8. Для прошивки тем же кодом очередной микросхемы следует повторить операции 1 – 7 по отношению к очередной микросхеме. В противном случае следует перейти к пунктам 9 и 10.
9. Произвести выход (выгрузить из памяти компьютера) из программной системы тестирования и контроля PASC-DIP. Для этого следует ввести Q, затем нажать Enter еще раз. Выбрать пункт главного меню «Quit», в ответ на предупреждающее сообщение ввести Y.
10. В случае прошивки микросхем новыми версиями программного обеспечения следует получить файл новой версии, повторить операции п. 3.1.3 (подготовка к записи), после чего повторить операции 1 – 7 данного пункта. В противном случае следует выключить программатор с помощью переключателя, располагающегося на его задней панели.
11. В целях возможности будущего тестирования ПЗУ необходимо записать в одну из его свободных ячеек значение контрольной суммы. Для этого следует запустить программу WROM_CRC.EXE из того же каталога, где находится файл main.0.

Рассмотрим еще одну важную с точки зрения эксплуатации контроллера технологическую операцию – операцию тестирования его аппаратурных компонентов.

3.3. Руководство по программному тестированию компонентов контроллера

Выше (в конструкторской части) уже говорилось о том, что спроектированная программная система способна работать в двух режимах: основном и тестовом.

3.3.1. Предварительные установки контроллера

Для установки адреса контроллера, а также режима его работы используются специальные dip- переключатели предварительных установок. Всего присутствует семь переключателей, из которых седьмой (7) ответственен за режим работы, а переключатели 0 – 5 используются для установки адреса контроллера (положение «Вкл.» соответствует логической «1», а «Выкл.» - логическому «0»). Единица («1») в седьмом бите означает, что текущий режим работы – тестовый, а «0» - означает, что текущий режим работы – основной.

3.3.2. Тестирование компонентов контроллера: подготовительный этап

Тестирование контроллера возможно исключительно в тестовом режиме работы. Перед тестированием компонентов контроллера следует выполнить следующие подготовительные операции:

1. Выставить тестовый режим на переключателе предварительных установок (см. п. 3.2.1);
2. Произвести соединение свободного COM- порта (например, COM1) компьютера с COM- портами (0 и 1) контроллера с помощью специального кабеля. При этом для тестирования

интерфейса RS-485 необходимо, чтобы контакт 6 разъема XP2 был объединен с контактом 7 разъема XS1, а также контакт 7 разъема XP2 с контактом 8 разъема XS1 (см. электрическую принципиальную схему, приложение П1);

3. Включить компьютер.
4. Запустить на компьютере программу NCTERM95.EXE из каталога C:\NC5.
5. В настройках программы NCTERM95 Установить следующие параметры обмена: скорость обмена 19200 бит/сек., номер порта (COM1 или COM2), остальные параметры – по умолчанию.
6. Для тестирования клавиатуры и индикации следует подсоединить шлейф клавиатуры к разъему XP1 контроллера.
7. Включить источник питания.
8. Вставить вилку от источника питания (+12В) в разъем XT1 питания контроллера.

3.3.3. Тестирование компонентов контроллера: основной этап

После выполнения всех операций подготовительного этапа на экране эмулятора терминала NCTERM95 появится строка с указанием версии программного обеспечения контроллера вида:

АЛГО-421ПВ Версия 1.1

Далее будет предложено нажать одну из клавиш клавиатуры компьютера с цифрами 1...5 для выполнения какого-либо одного теста либо нажать Enter для выполнения всех тестов. Ниже поясняется назначение клавиш:

- 1 – **тест ОЗУ** (микросхем оперативной памяти);
- 2 – **тест ПЗУ** (микросхем постоянной памяти);

- 3 – **тест 0 интерфейса RS-485** (интерфейсная схема линии 0 – передатчик, интерфейсная схема линии 1 - приемник);
- 4 - **тест 1 интерфейса RS-485** (интерфейсная схема линии 1 – передатчик; интерфейсная схема линии 0 - приемник);
- 5 - **тест клавиатуры и индикации** панели управления пульта вахтера.

3.3.3.1. Тестирование ОЗУ

После появления на экране сообщения «Тестирование ОЗУ..» запускается процесс тестирования.

Тестирование ОЗУ основано на записи в ячейки оперативной памяти значений \$55 (01010101) и проверке записанного и считываемого значений по каждой из ячеек.

Для прекращения тестирования ОЗУ следует нажать любую клавишу. В этом случае появится сообщение «Тест прерван.»

В случае успешного завершения теста появится сообщение «Ок.». В случае возникновения ошибки тестирования (т.е. когда считанное и записанное значение не совпали) будет выведено сообщение об ошибке тестирования с указанием адреса, при тестировании ячейки по которому произошла ошибка.

3.3.3.2. Тестирование ПЗУ

После появления на экране сообщения «Тестирование ПЗУ..» запускается процесс тестирования.

Тестирование ПЗУ основано на сверке контрольной суммы (КС), подсчитанной по всем ячейкам, со значением КС, записанным единственным

раз в одной из свободных ячеек сразу после прошивки микросхемы с помощью специальной программы WROM_CRC.EXE.

Для прекращения процесса тестирования ПЗУ следует нажать любую клавишу. В этом случае появится сообщение «Тест прерван.»

В случае успешного завершения теста появится сообщение «Ок.». В случае возникновения ошибки тестирования (т.е. когда подсчитанная и записанная ранее, эталонная контрольные суммы не совпали) будет выведено сообщение об ошибке тестирования.

3.3.3.3. Тестирование интерфейса RS-485

После появления на экране сообщения «Тестирование RS-485-0..» или «Тестирование RS-485-0..» начинается тест 0 или 1 интерфейса RS-485.

Тестирование RS-485 основано на том, что одна из интерфейсных микросхем передает, а другая – принимает данные. Отправленное и полученное значение сравниваются. В случае теста 0 происходит передача байта данных с линии 0 на линию 1. В случае теста 1 происходит передача байта данных с линии 1 на линию 0.

В случае успешного завершения теста появится сообщение «Ок.». В случае возникновения ошибки тестирования (т.е. когда переданное и принятое значения не совпали) будет выведено сообщение об ошибке тестирования.

3.3.3.4. Тестирование микросхемы SE210, клавиатуры и индикации панели управления пульта вахтера

После появления на экране сообщения «Тестирование SE210..» начинается тестирование клавиатуры и индикации панели управления ПВ.

Тестирование клавиатуры состоит в том, что в ответ на нажатую клавишу с новой строки выводится сообщение, содержащее номер ТК, номер прохода и функцию нажатой клавиши: <№ ТК> <№ прохода> <Функция>, где

<№ ТК> – число в диапазоне от 00 до 04;

<№ прохода> – число в диапазоне от 00 до 01;

<Функция> имеет следующие значения:

00 – нажата клавиша «Заблокировать проход»;

01 - нажата клавиша «Освободить проход»;

03 - нажата клавиша сброса задержания.

Тестирование индикаторов состоит в том, что при нажатии на клавиши «Заблокировать проход» и «Освободить проход» изменяются состояния индикаторов типов «Освобождение прохода», «Блокировка прохода» и «Автоматический режим работы» точно так же, как и в случае основного (не тестового) режима работы контроллера:

- при нажатии клавиши «Заблокировать проход» *состояние* соответствующего проходу *индикатора блокировки* прохода меняется на противоположное; *состояние индикатора освобождения* прохода становится противоположным изменившемуся состоянию индикатора блокировки (то есть если состояние индикатора блокировки изменилось на «Выкл.», (индикатор погашен), то индикатор освобождения должен загореться, и наоборот); в случае, если новое

состояние индикатора блокировки – «Вкл.», то индикатор автоматического режима приобретает состояние «Выкл.»;

- при нажатии клавиши «Освободить проход» *состояние* соответствующего проходу *индикатора* *освобождения* прохода меняется на противоположное; *состояние индикатора блокировки* прохода становится противоположным изменившемуся состоянию индикатора блокировки; в случае, если новое состояние индикатора освобождения – «Вкл.», то индикатор автоматического режима приобретает состояние «Выкл.».

Кроме того, всякое нажатие на клавишу «Сброс задержания» *в тестовом режиме* приводит к изменению состояния соответствующего ей индикатора на противоположное.

Для тестирования индикаторов «Несанкционированный проход» и «Нет связи с управляющим компьютером», в соответствии которым не поставлено ни одной клавиши панели управления пульта вахтера, используются клавиши клавиатуры компьютера с цифрами **6** и **7** соответственно.

Для прекращения теста следует нажать клавишу *Q* *компьютера*.

Если имеют место описываемые ниже реакции на нажатие клавиш, это означает, что микросхема SE210, а также панель управления пульта вахтера работают нормально. В противном случае имеет место одна из следующих неисправностей:

- неисправность микросхемы SE210;
- нарушена целостность печатных проводников, связывающих микросхему SE210 с микроконтроллером;
- электрическая либо механическая неисправность одного из компонентов панели управления пульта вахтера.

В случае, если настоящий тест выявил неисправность, но ее характер априорно не известен, требуется дополнительное исследование специалиста-настройщика электронного оборудования с привлечением специальных приборов (осциллографа, вольтметра, амперметра), описание которых и методики использования которых выходит за рамки данной работы.

3.3.4. Тестирование компонентов контроллера: заключительный этап

По завершении операции программного тестирования компонентов контроллера следует:

1. выключить его питание, *если* контроллер не подлежит эксплуатации и более тщательному анализу неисправностей в данный момент;
2. произвести выход из программы NCTERM95 (комбинация клавиш Alt+X).

Наконец, в заключение технологической части дипломного проекта приведем руководство по эксплуатации пульта вахтера автоматизированной проходной. Знание приведенных ниже сведений необходимо пользователю данного пульта – вахтеру автоматизированной проходной.

3.4. Руководство пользователя пульта вахтера

В данном разделе излагаются основные положения, связанные с назначением и устройством панели управления пульта вахтера. Приводится описание функциональных клавиш и индикаторов, а также описание ситуаций, в которых может возникнуть потребность воспользоваться той или иной клавишей. Рассмотрен наиболее общий случай десятипроходного варианта автоматизированной проходной.

Примем следующие соглашения. Под *пользователем пульта* (ППВ) будем понимать вахтера, обслуживающего автоматизированную проходную посредством пульта вахтера, а под *пользователями проходной* (ППР) - работников данного предприятия, эксплуатирующих в своей совокупности данную проходную.

3.4.1. Общие положения

Панель управления пульта вахтера (см. рис. 3.1) представлена совокупностью функциональных клавиш и индикаторов, отображающих текущее состояние пульта вахтера.

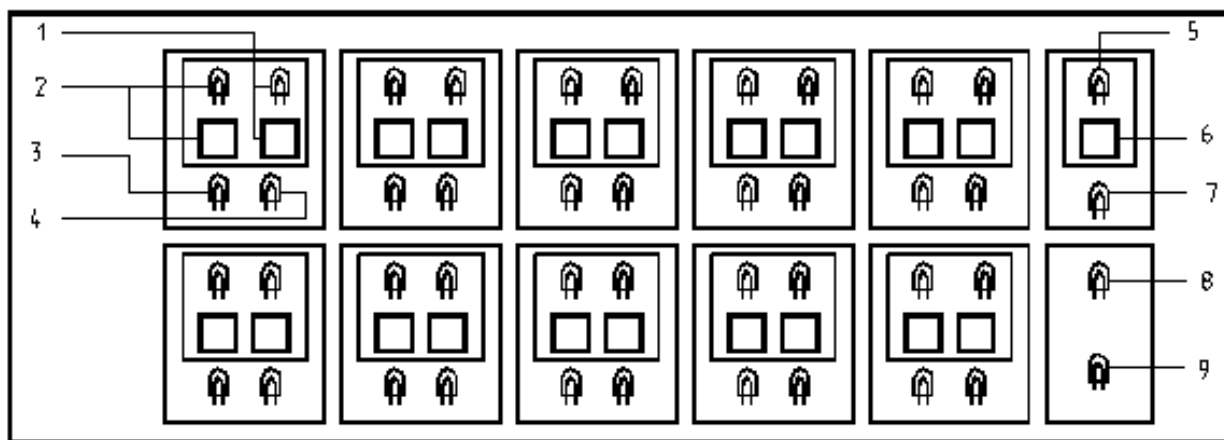


Рис. 3.1. Панель управления пульта вахтера

Как видно из рисунка, панель управления в общем случае содержит двадцать клавиш, относящихся к проходам, и одну клавишу особого назначения «Сброс задержания» (позиция б). Панель также содержит 40 индикаторов по каждому из проходов, и три индикатора особого назначения («Задержание» - позиция 5; «Несанкционированный проход» - позиция 7; «Нет связи с управляющим компьютером» - позиция 9; «Питание панели управления» - позиция 8).

3.4.2. Описание режимов работы проходов, функциональных клавиш и индикаторов проходов панели управления пульта вахтера

Различают два основных режима работы каждого из проходов: *автоматический* и *ручного управления*.

В автоматическом режиме (в авторежиме) открывать и закрывать проходы может только сам контроллер под управлением его программного обеспечения.

Режим ручного управления подразумевает, что пользователем (вахтером) была нажата одна из клавиш группы 1 или 2 (см. рисунок 3.1). При этом авторежим может принимать две формы:

- *нажата клавиша «Заблокировать проход»* - проход закрывается на неопределенное время (то есть делается недоступным для прохода какого-либо ППР), но до тех пор, пока не будет освобожден либо не будет произведен возврат в авторежим по инициативе ППВ;
- *нажата клавиша «Освободить проход»* - проход при этом проход открывается на неопределенное время (то есть делается доступным для прохода всех ППР), но до тех пор, пока не будет заблокирован либо не будет произведен возврат в авторежим по инициативе ППВ.

Цифрой 1 на рис. 3.1 помечены клавиши «Освободить проход» и индикаторы освобождения прохода, а цифрой 2 - соответственно клавиши «Заблокировать проход» и индикаторы блокировки прохода по каждому из проходов. *Для возврата в авторежим следует повторно нажать ту же клавишу.*

Отметим, что *по каждому из проходов* имеется следующая индикация состояний и режима:

- *индикатор освобождения прохода* – располагается возле соответствующей клавиши; загорается, когда проход находится в состоянии освобождения (был освобожден ППВ);
- *индикатор блокировки прохода* – располагается возле соответствующей клавиши; загорается, когда проход находится в состоянии блокировки (был заблокирован ППВ);
- *индикатор авторежима прохода* (позиция 3) - загорается, когда проход переходит в авторежим (см. выше);
- *индикатор аварии прохода* (позиция 4) - загорается, когда произошел отказ в работе прохода (авария).

3.4.3. Описание функциональных клавиш и индикаторов особого назначения панели управления пульта вахтера

Индикация задержания (позиция 5). Данный индикатор горит в случае, если было произведено временное задержание в автоматическом режиме прохода (в силу несоответствия кода пропуска ППР, времени и дня прохода ППР либо его пин-кода допустимым значениям) и проход продолжает находиться в данном состоянии. В этом случае вахтер (ППВ) сам определяет, пропустить или нет данного работника.

Клавиша «Сброс задержания» (позиция 6). Позволяет вывести проход из состояния задержания, открыв его на короткое время (так, чтобы ППР смог пройти через данный рубеж) по инициативе ППВ.

Индикация несанкционированного прохода (позиция 7). Данный индикатор горит в случае, если имела место попытка *несанкционированного прохода* (то есть когда пользователь пытался пройти через проходную без ввода даже кода карточки). В случае возникновения данной ситуации вахтер (работник службы безопасности) обязан принять меры по пресечению попытки несанкционированного прохода в соответствии с его полномочиями.

Индикация «Нет связи с УК». Сигнализирует о ситуации, когда произошел отказ в линии связи либо вышел из строя сам управляющий компьютер. В этом случае возможно возникновение ситуации, когда в авторежиме проход остается в своем *нормальном* состоянии (нормально-закрытым или нормально-открытым, в зависимости от вида турникета), хотя должен был бы изменить свое состояние в ответ на запрос со стороны ППР (в том случае, если работник был принят или уволен в данный день). Кроме этого, нарушается обмен текущими событиями с компьютером. *Действия ППВ в данном случае:*

- сообщить о неполадке техническому специалисту предприятия;
- в случае возникновения описанной выше критической ситуации (к примеру, работник утверждает, что он сегодня был принят на работу, но БД контроллера еще не была обновлена), решения о проходе возлагаются на ППВ (он может манипулировать блокировкой или освобождением прохода путем нажатия соответствующих клавиш, см. п. 3.4.2).

4. Организационно-экономическая часть

В рамках данной части дипломного проекта производится расчет необходимых *затрат на разработку* программного продукта и получение его себестоимости, а также оценивается *экономическая эффективность* проекта в зависимости от выбранной стратегии реализации на рынке программного продукта и контроллера пульта вахтера как компонентов системы безопасности предприятий.

4.1. Расчет затрат на разработку программного продукта

Трудоемкость разработки (в человекомесяцах, ч/м) программного обеспечения рассчитывается по формуле

$$T_n = \frac{(L \cdot K_n \cdot K_{yn})}{\Phi \cdot nn}, \quad (4.1)$$

где

L - объем программы в строках, шт.,

nn - норма производительности труда программиста, представляющая собой объем полезной работы (количество написанных строк), выполняемой в единицу времени;

Φ - среднемесячный фонд рабочего времени, суток;

K_n - коэффициент новизны разработки;

K_{yn} - коэффициент унификации (показывает степень использования в разработке типовых программ).

В нашем случае указанные выше параметры будут иметь следующие значения:

- $L = (130 \cdot 1024) / 80 = 1664$ строки*;
- $m = 20$ строк/сутки;
- $\Phi = 25$ суток;
- $K_n = 0,8$;
- $K_{yn} = 0,7$.

Таким образом, трудоемкость разработки ПО составляет

$$T_n = (1664 \cdot 0,8 \cdot 0,7) / (25 \cdot 20) = 1,86 \text{ ч/м.}$$

Расчет стоимости ч/м программиста:

- заработная плата программиста составляет 1500 руб. в месяц.
- дополнительная заработная плата составляет 20% от основной, то есть 300 руб.

Начисления на заработную плату имеют следующую структуру (в процентах от основной и дополнительной заработной платы):

- в пенсионные фонды – 29%, из них 28% - в общий пенсионный фонд, 1% - во внебюджетный пенсионный фонд;
- отчисления на социальное страхование – 5,4%;
- с 1 января 2000 г. был введен дополнительный налог в фонд возмещения последствий несчастных случаев на производстве, его размер определяется условиями работы, в нашем случае – 1,8%;
- взносы в фонд обязательного медицинского страхования (ОМС) – 3,6%;
- отчисления в фонд занятости – 1,5%.

Итого начисления на основную и дополнительную заработную плату составляют $29 + 5,4 + 1,8 + 3,6 + 1,5 = 41,3\%$.

* Объем исходного текста программного обеспечения на языке высокого уровня составляет примерно 130 Кбайт. Длину строки считаем равной 80-ти символам

В нашем случае это составит сумму $0,413 * (1500 + 300) = 743,4$ руб.

Итого стоимость одного человекомесяца:

$$1500+300+743,4 = 2543,4 \text{ руб.}$$

Накладные расходы составляют 25% от заработной платы:

$$Z_{\text{накл}}=0,25 \cdot 1500 = 375 \text{ руб.}$$

Итого вместе с накладными расходами:

$$Z_{\text{од}} = 2543,4+375=2918,4 \text{ руб.}$$

Стоимость одного машинного часа с учетом первоначальной стоимости ЭВМ (отнесенной к 1 часу работы машины) $S_{\text{маш1}}$, стоимости потребляемой электроэнергии (отнесенной к 1 часу работы машины) $S_{\text{эл}}$ и затрат на профобслуживание и ремонт $S_{\text{рем}}$ (отнесенного к 1 часу работы машины) определяется по формуле:

$$S_{\text{мч}} = S_{\text{маш1}} + S_{\text{эл}} + S_{\text{рем}}. \quad (4.2)$$

Срок службы ЭВМ составляет 4 года. Первоначальная стоимость (включая стоимость системного программного обеспечения – предустановленной операционной системы Windows 98) - 27000 руб. Потребляемая мощность - 0,3 кВт/ч.

При эксплуатации машины 8 часов в день, 22 дня в месяц имеем

$$S_{\text{маш1}} = 27000/(4 \cdot 12 \cdot 22 \cdot 8)=3,2 \text{ руб/ч.}$$

Принимая стоимость электроэнергии 0,3 руб/кВт, стоимость потребляемой энергии в час равна

$$S_{\text{эл}}=0,3 \cdot 0,3=0,09 \text{ руб.}$$

Затраты на профилактическое обслуживание и ремонт составляют 20% от стоимости машины:

$$S_{\text{рем}}=0,2 \cdot 3,2=0,64 \text{ руб.}$$

Тогда стоимость одного машинного часа будет равна

$$C_{\text{м}} = S_{\text{мч}} = 3,2+0,09+0,64=3,93 \text{ руб/ч.}$$

В ходе разработки программного обеспечения были использованы также инструментальные средства в виде специализированного

программного обеспечения – ограниченно свободно распространяемая (shareware) версия комплекта HIWARE, стоимость S_{no} которой может быть оценена приблизительно в 3000 руб., использовать которую планируется в течение шести (6) месяцев.

Себестоимость программного продукта определяется как

$$C_{np} = Z_{од} \cdot T_n + 8 \cdot T_n \cdot C_m \cdot \Phi \cdot K_{исп} + (S_{no} / T_{no}) \cdot T_n, \quad (4.3)$$

где

$Z_{од}$ - сумма основной и дополнительной заработной платы разработчика программы, начисления на нее и накладные расходы, руб./мес.;

T_n – время, затрачиваемое разработчиком на разработку, составление алгоритма и написание программы, мес.;

C_m - себестоимость одного часа машинного времени, руб.;

T_{no} – время, в течение которого предусматривается эксплуатация инструментального программного обеспечения, использованного при разработке программы, мес.;

S_{no} – стоимость инструментального программного обеспечения, руб.;

$K_{исп} = 0,7$ - коэффициент использования ЭВМ.

Итак, в соответствии с (4.3) полная себестоимость программного продукта равна

$$\begin{aligned} C_{np} &= 2918,4 \cdot 1,86 + 8 \cdot 1,86 \cdot 3,93 \cdot 25 \cdot 0,7 + (3000/6) \cdot 1,86 = \\ &= 1,86 \cdot (2918,4 + 3,93 \cdot 25 \cdot 0,7 \cdot 8 + 500) = 7381,6 \text{ руб.} \end{aligned}$$

Нами были подсчитаны затраты на разработку программного обеспечения.

В силу того, что настоящий программный продукт не может принести экономического эффекта сам по себе, а эффект достигается только в результате реализации на рынке всей системы безопасности в комплексе (включая контроллер пульта вахтера, контроллеры турникетов и их ПО,

управляющий компьютер и его ПО, турникеты, считыватели кодов и т.д.), в настоящем дипломном проекте экономический эффект не рассчитывается.

5. Обеспечение безопасности жизнедеятельности и охрана окружающей среды

В разделах «Обеспечение безопасности труда» и «Охрана окружающей среды» данной части дипломного проекта проводится анализ опасных и вредных факторов воздействующих соответственно на человека и окружающую среду в процессе производства контроллера пульта вахтера, а также его эксплуатации. Приводятся необходимые расчеты.

5.1. Обеспечение безопасности труда

Безопасность жизнедеятельности – наука о комфортном и безопасном взаимодействии человека с средой его обитания. Из данного определения следует, что мероприятия по обеспечению безопасности труда являются составной частью мероприятий по обеспечению безопасности жизнедеятельности и главным образом относятся к производственным условиям.

Полностью безопасных и безвредных производств не бывает.

Основные задачи безопасности условий труда направлены на снижение производственного травматизма и профессиональной заболеваемости, сокращение доли ручного труда и рабочих мест с неблагоприятными для здоровья (опасными и вредными) производственными факторами.

В данном разделе производится анализ и классификация *опасных и вредных производственных факторов*, воздействующих на *пользователя пульта вахтера* в процессе эксплуатации пульта вахтера, а также на производственных рабочих в ходе производства печатной платы. Рассматриваются способы борьбы с данными факторами.

5.1.1. Классификация опасных и вредных производственных факторов, воздействующих на человека в процессе эксплуатации пульта вахтера

В процессе эксплуатации пульта вахтера на пользователя (вахтера) воздействуют главным образом следующие опасные и вредные производственные факторы (в соответствии с ГОСТ 12.0.003-83 “Классификация опасных и вредных производственных факторов”).

а) Физические факторы:

- неудовлетворительные параметры микроклимата помещения;
- повышенный уровень шума на рабочем месте;
- опасный уровень напряжения в электрической цепи, замыкание которой может произойти через человека;
- пожарная опасность;
- неудовлетворительное освещение.

б) Психофизиологические факторы:

- гиподинамия;
- эмоциональные перегрузки.

Рассмотрим характеристику и действие каждого из перечисленных выше факторов на вахтера автоматизированной проходной в отдельности, а также меры борьбы с перечисленными факторами.

5.1.1.1. Неудовлетворительные параметры микроклимата

Микроклимат в производственных помещениях определяется главным образом следующими параметрами:

- температурой воздуха, °С;
- относительной влажностью, %;
- скоростью движения воздуха на рабочем месте, м/с.

Работа вахтера автоматизированной проходной относится к работам легкой тяжести (I категория), т.к. выполняется сидя или стоя, может быть связана с ходьбой, но не требует систематического физического напряжения и переноски тяжестей.

Согласно категории проводимых работ и ГОСТ 12.1.005-76, оптимальные параметры микроклимата при проведении данной работы должны соответствовать значениям, указанным в таблице 5.1.

Таблица 5.1. Оптимальные параметры микроклимата

Период года	Температура, °С	Относительная влажность воздуха, %	Скорость движен. воздуха, м/с
Холодный и переходный	20-23	60-40	0,1
Теплый	22-25	60-40	0,1

Запыленность воздуха в кабине вахтера не должна превышать 0,2 мг/м² при размере частиц не более 2 мкм.

Для обеспечения установленных параметров микроклимата и чистоты воздуха применяют *вентиляцию* и *отопление*.

В целях защиты вахтера и прочих работников предприятия от охлаждения проникающим через ворота холодным (в холодное время года) воздухом необходимо применять *воздушно-тепловые завесы* (см. [9], с. 89-90). При работе завес холодный воздух смешивается с теплым. Полученная смесь более теплого воздуха поступает на рабочие места и к кабине вахтера. В ходе работы завес создается дополнительное сопротивление притоку холодного воздуха. Завесы могут быть устроены как с нижней, так и с боковой подачей воздуха. На территории проходной также должна присутствовать общеобменная вентилляция.

Для охлаждения контроллера пульта вахтера и контроллеров турникетов следует применять соответствующую местную вентиляцию. Естественный воздухообмен кабины вахтера с окружающей его проходной осуществляется через дверные и оконные щели. Может быть также применен механический вентилятор либо система местного кондиционирования для автоматического регулирования параметров микроклимата, которые требуют определенных затрат, но способствуют улучшению состояния работника и росту производительности труда.

Кроме того, в помещении проходной предусматривают систему *отопления*, которая наряду с завесами должна обеспечивать достаточное, постоянное и равномерное нагревание воздуха в холодный период года.

5.1.1.2. Повышенный уровень шума на рабочем месте

Шум в кабине вахтера может возникать при работе вентиляторов аппаратуры, системы вентиляции кабины вахтера (если таковая присутствует), а также источника питания пульта.

Кроме того, в качестве источников шума могут выступать работающие турникеты проходной и система вентиляции проходной, аппаратура воздушно-тепловых завес.

При длительном воздействии шума на человека происходят нежелательные явления: снижается острота слуха, повышается кровяное давление. Кроме того, шум оказывает воздействие на общее состояние человека, вызывая чувства неуверенности, стесненности, тревоги, плохого самочувствия, что приводит к снижению производительности труда.

Ослабления шума можно достигнуть следующими способами:

1. уменьшением шума в источнике (турникете);
2. изменением направленности излучения (направленность излучения зависит от типов турникетов и их ориентации по отношению к кабине вахтера);
3. рациональной планировкой рабочего помещения (путем правильного выбора места расположения кабины вахтера);
4. борьбой с шумом на пути его распространения (кабина вахтера должна быть по возможности выполнена с применением *звукоизолирующего* материала).
5. акустической обработкой помещений путем применения *звукопоглощающих* материалов, например, древесноволокнистых плит с окрашенной и профилированной поверхностью, см. [9], с. 192-205.

Инженерами фирмы «Алгонт» были предприняты меры по уменьшению уровня шума в источнике. Так, конструкции *нормально закрытых* турникетов АЛГО-122Т и АЛГО-121Т содержат гидравлические демпферы, которые делают работу механизма более плавной. Механический доводчик обеспечивает доворот механизма в нормальное положение. Что касается *нормально открытых* турникетов, то они издадут шум лишь при попытке несанкционированного прохода.

Уровень шума на рабочем месте должен соответствовать требованиям ГОСТ 12.1.003-83.

Согласно этому ГОСТу, уровень звукового давления на постоянных рабочих местах в производственных помещениях при продолжительности шума более четырех часов должен соответствовать данным в таблице 5.2.

Таблица 5.2. Допустимые уровни шума

Ра- бочие места	Уровни звукового давления в дБ в октавных полосах со среднегеометрическими частотами, Гц								Уровень звука, дБА
	63	125	250	500	1000	2000	4000	8000	
1	94	87	82	78	75	73	71	70	80
2	83	74	68	63	60	57	55	54	65

Цифрами 1 и 2 здесь помечен вид рабочих мест «Кабины наблюдений и дистанционного управления». В таблице цифре 2 в графе «Рабочие места» соответствует случай, когда предусматривается речевая связь по телефону, цифре 1 – когда такая связь не предусматривается. В большинстве случаев возникает необходимость использования вахтером телефонной связи.

5.1.1.3. Опасный уровень напряжения в электрической цепи, замыкание которой может произойти через человека

В зависимости от типа материала пола кабины пульта вахтера, данное помещение может быть отнесено к *помещениям без повышенной опасности* либо к *помещениям с повышенной опасностью* поражения электрическим током (см. [9], с.284-285).

В кабине пульта вахтера используются следующие основные потребители электроэнергии:

- **источник питания пульта вахтера** (потребляет ~220В);
- пульт вахтера (постоянное напряжение +5В и +12В);
- **освещение кабины** – переменное напряжение ~220В.

Причинами поражения человека электрическим током в нашем случае могут стать:

- случайное прикосновение к токоведущим частям, находящимся под напряжением;

- появление напряжения на металлических конструктивных частях электрооборудования – корпусах, кожухах и т.д. в результате повреждения изоляции и других причин;

При работе аппаратуры запрещается:

- проверять на ощупь наличие напряжения токоведущих частей аппаратуры;
- применять для соединения блоков и приборов провода с поврежденной изоляцией;
- проводить работу и монтаж в аппаратуре, находящейся под напряжением;
- подключать блоки и приборы к работающей аппаратуре.

Корпус блока питания должен быть заземлен в соответствии с ПУЭ-85.

5.1.1.4. Пожарная опасность

В качестве *горючего компонента* в кабине вахтера и на проходной могут служить строительные материалы для акустической и эстетической отделки помещения, перегородки, полы, предметы мебели, изоляция силовых кабелей, а также радиотехнические детали и соединительные провода электронной схемы, пластмассовый корпус панели управления пульта вахтера.

Окислитель представлен в виде кислорода атмосферного воздуха.

Источниками воспламенения в кабине вахтера и на проходной могут быть электрические искры, дуги и перегретые участки. Источники воспламенения возникают в электронных схемах, кабельных линиях, а также в осветительной проводке.

Таким образом, на рабочем месте вахтера могут присутствовать все три основных фактора, способствующих возникновению пожара.

Кабельные линии электропитания, а также осветительная проводка состоят из горючего изоляционного материала, а также содержат вероятные источники открытого огня. Они являются наиболее опасными элементами кабины вахтера с точки зрения возникновения и развития пожара.

Для обеспечения своевременных мер по обнаружению и локализации пожара, эвакуации рабочего персонала, а также для уменьшения материальных потерь необходимо предусмотреть:

- наличие системы автоматической пожарной сигнализации;
- наличие эвакуационных путей и выходов;
- наличие первичных средств тушения пожаров: пожарные стволы, внутренние пожарные водопроводы, сухой песок либо огнетушащий порошок, огнетушители.

Производством систем автоматической пожарной сигнализации занимается, в частности, фирма «Алгонт», поставляющая комплексные системы безопасности предприятия, включая автоматизированные проходные.

Сигналы от датчиков сигнализации могут поступать в систему безопасности с различного оборудования. Это могут быть различные панели охранно-пожарной сигнализации, допускающие подключение к компьютеру. При этом на экранах операторов индицируются соответствующие сообщения и выводятся графические планы помещений с указанием места, где произошло возгорание. Кроме того, поставляемые системы способны не только сигнализировать об опасности в результате срабатывания термодатчиков, но и управлять различными устройствами, допускающими релейное управление, например, включать видеокамеры и системы пожаротушения, разблокировать эвакуационные проходы и т.д.

При проектировании моделей нормально закрытых турникетов АЛГО-121Т и АЛГО-122Т, широко используемых при построении автоматизированных проходных, инженерами фирмы «Алгонт» были

учтены требования пожарной безопасности. Рукоятки турникетов снабжены специальным *устройством экстренного открытия*. При приложении к рукоятке значительного усилия, направленного в сторону выхода с объекта, рукоятка складывается и освобождает зону прохода. Необходимое усилие регулируется мощной пружиной.

5.1.1.5. Неудовлетворительное освещение

Свет является важным стимулятором не только зрительного анализатора, но и организма в целом, а также общей работоспособности человека. Положительное влияние его на производительность труда и качество работы в настоящее время не вызывает сомнений.

Вредным фактором производства в нашем случае является недостаток естественного освещения, поэтому необходимо использовать комбинированное (естественное и искусственное) освещение.

Освещенность рабочей зоны (в нашем случае – кабины пульта вахтера) должна соответствовать нормам СНиП 11-4-79 «Искусственное освещение для зрительной работы малой степени точности (разряд V)».

В качестве светильников могут использоваться люминисцентные лампы либо лампы накаливания.

5.1.1.6. Психофизиологические факторы

Из психофизиологических факторов, как уже было отмечено выше, на вахтера особенно ощутимое воздействие оказывают *гиподинамия* и *эмоциональные перегрузки*.

Гиподинамия возникает вследствие недостаточной двигательной активности работника в ходе выполнения работы. Она может привести к возникновению различных заболеваний двигательного аппарата, сердечно-сосудистой, иммунной и нервной систем, способствовать общей

утомляемости работника. В целях уменьшения степени подверженности работника гиподинамии можно порекомендовать использование перерывов в работе с целью выполнения несложных физических упражнений, жесткую и санитарно обоснованную регламентацию сменности работы и часов рабочего времени, а также пропаганду и организацию физически активного досуга работника вне работы.

Эмоциональные перегрузки возникают вследствие воздействия ряда факторов, влияющих на эмоциональную активность работника. На эмоциональное состояние работника оказывают влияние степень эргономической и эстетической проработки интерьера помещения, его цветовой и световой гаммы, характер звуковых осведомительных сигналов и связанная с этим степень комфортности работы. Большая часть эмоциональных перегрузок может быть также обусловлена наличием общения с работниками предприятия в силу специфики работы вахтера, возможностью возникновения конфликтных ситуаций. *Меры борьбы* с влиянием данного фактора: эстетическая проработка помещения (кабины вахтера); подбор оптимальной частоты звучания осведомительных сигналов; предупреждение конфликтных ситуаций.

Выше были перечислены опасные и вредные факторы, возникающие в процессе эксплуатации пульта вахтера. Перечислим теперь опасные и вредные факторы, возникающие в процессе производства печатных плат при изготовлении контроллеров.

5.1.2. Классификация опасных и вредных производственных факторов, воздействующих на человека в процессе производства контроллера

При изготовлении контроллера АЛГО-421ПВ на различных этапах производства присутствуют следующие опасные и вредные производственные факторы (в соответствии с ГОСТ 12.0.003-83 “Классификация опасных и вредных производственных факторов”):

- повышенный уровень электромагнитных излучений;
- сосуды и системы под давлением;
- ионизирующее излучение;
- повышенный уровень вибрации;
- повышенный уровень шума на рабочем месте;
- химический фактор – выделение вредных пылей, газов и паров.

В то же время при изготовлении контроллера ввода-вывода применяется различное оборудование, которое является источником опасных и вредных производственных факторов:

- общеобменная вентиляция;
- системы кондиционирования;
- станки вырубки заготовок ПП;
- сверлильные станки;
- травление;
- гальваническое меднение;
- растворители;
- пайка.

Рассмотрим вредный фактор производства – выделение вредных пылей, газов и паров.

5.1.2.1. Выделение газов, паров и пылей

При производстве контроллера существуют следующие возможности загрязнения воздушной среды рабочей зоны:

- повышенная запыленность и загазованность воздуха рабочей зоны;
- выделение газов на этапах травления, гальванического меднения и покрытия припоем ПП (железа хлорид — класс опасности 2, ПДК в атмосферном воздухе населенных мест 0,04 мг/м, кислота серная — класс опасности 2, ПДК в рабочей зоне 1 мг/м);
- испарение летучих растворителей при покрытии плат защитным лаком;
- испарение тяжелых металлов при пайке ПП (свинец — класс опасности 1, ПДК в рабочей зоне 0,01 мг/м).

При механической обработке ПП в воздушную среду попадает пыль. Попадание пыли в дыхательные пути приводит к их раздражению, вызывает отеки легких, аллергии. Проникновение ядов и твердых частиц пыли в организм рабочего осуществляется также через кожу, что приводит к различным кожным заболеваниям. Повышенная запыленность обуславливает также развитие различных болезней органов зрения.

Поэтому участки механической обработки ПП, травления, гальванического меднения, пайки должны быть оборудованы вытяжной вентиляцией. На участках травления, гальванического меднения необходимо применять индивидуальные средства защиты — халаты, фартуки, перчатки, очки, респираторы и противогазы.

Значение предельно допустимых концентраций вредных примесей в воздухе рабочей зоны приводится по ГОСТ 12.1.005-88 “Общие санитарно-гигиенические требования к воздуху рабочей зоны” и СНиП-245-85.

При травлении и нанесении гальванопокрытий с открытых поверхностей выделяются пары химически агрессивных веществ, оказывающих вредное раздражающее и сенсibiliзирующее действие. Для локализации этих вредных веществ используются *бортовые отсосы* (см. [9]).

Произведем расчет такого бортового отсоса.

5.1.2.2. Расчет бортового отсоса

Расчет бортового отсоса сводится к определению количества воздуха, удаляемого двухбортовым отсосом с ванн для химического травления (см. рис. 5.1). Считаем, что при травлении используются щелочные медно-аммиачные травильные растворы.

Исходные данные:

- ширина ванны: $b=1,1$ м;
- длина ванны: $l=1,4$ м;
- температура раствора в ванне: $t_{ж}=40^{\circ}\text{C}$;
- глубина уровня жидкости: $h=170$ мм;
- температура окружающей среды: $t_{окр}=21^{\circ}\text{C}$;
- подвижность воздуха: $v=0,2$ м/с.

Количество удаляемого воздуха от горячих ванн ($t_{ж} > t_{окр}$) можно определить по методу Баранова:

$$L = a \cdot x \cdot l \cdot S \cdot \sqrt[3]{t_{ж} - t_{окр}}, \quad (5.5)$$

где

a — коэффициент, зависящий от ширины ванны и степени вредности вещества (см. [11]), $a = 350$;

x — поправочный коэффициент на глубину уровня жидкости h ; $x=1,2$;

S — поправочный коэффициент на подвижность ν воздуха в помещении;
 $S = 1,51$.

Тогда для предотвращения распределения паров аммиака, выделяющихся в процессе травления, необходимое количество воздуха, удаляемого двухбортовым отсосом, в соответствии с (5.5) будет равно

$$L = 350 \cdot 1,2 \cdot 1,4 \cdot 1,51 \cdot \sqrt[3]{40 - 21} = 2369,2 \text{ м}^3/\text{ч}.$$

Исходя из требуемого объема отсасываемого воздуха определяются конструктивные параметры отсоса. При работе с одной стороны ванны можно применить бортовой отсос со сдувом.

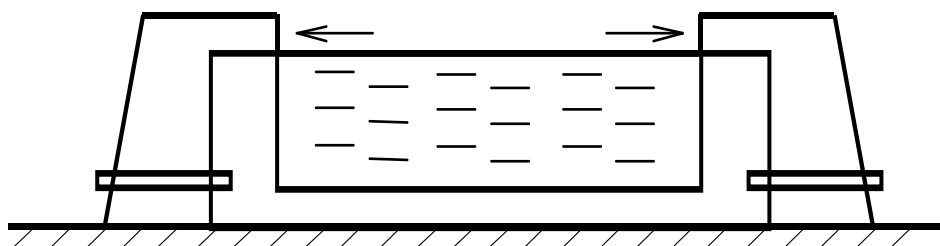


Рис. 5.1. Ванна с двухбортовым отсосом

5.2. Охрана окружающей среды

Охраной окружающей среды занимается раздел науки, именуемый *производственной экологией*.

Производственная экология – раздел науки ОБЖД, призванный идентифицировать негативное воздействие производств и технических средств на биосферу и техносферу, разрабатывать и применять средства для снижения этого воздействия.

За прошедшее столетие уровень антропогенной нагрузки на окружающую среду возрос настолько, что, если не принимать адекватных мер, может быть поставлена под сомнение возможность дальнейшего развития цивилизации и выживания человека как вида. Именно поэтому вопросы охраны окружающей среды приобрели в последние 2...3 десятилетия особую актуальность.

В силу того, что непосредственное негативное влияние на окружающую среду в ходе разработки и эксплуатации собственно программного обеспечения контроллера пульта вахтера отсутствует, в данном разделе производится анализ негативных воздействий на окружающую среду *в ходе производства контроллера пульта вахтера*, а также способов борьбы с ними. В частности, приводится расчет воздушного пылевого фильтра.

5.2.1. Факторы негативного воздействия на окружающую среду в процессе производства контроллера и способы борьбы с ними

В процессе производства контроллера пульта вахтера используется целый комплекс технологических приемов, связанных с переработкой различных по своей природе исходных материалов, последующей

обработкой и сборкой деталей для получения функционально завершенного изделия.

На стадии механической обработки деталей образуется пыль различной степени измельчения. Пыль при помощи общеобменной и местной вентиляции удаляется от рабочих мест. Перед выбросом в атмосферу воздуха из устройств вентиляции он должен пройти очистку от пыли. Основными устройствами, очищающими воздух от пыли, являются *пылеуловители* и *фильтры*.

Пылеуловители делятся на два вида — мокрые и сухие. Мокрые пылеуловители дороже сухих, но они обеспечивают более качественную очистку воздуха.

Фильтры обеспечивают более качественную очистку воздуха от пыли. При высокой концентрации пыли, прежде чем очистить воздух с помощью фильтров, его очищают с помощью пылеуловителей.

При работах, связанных с травлением, пайкой ПП, образуются *вредные вещества и пары вредных соединений*. Применяются следующие методы очистки выбросов промышленных предприятий от газообразных и парообразных примесей:

- абсорбционный;
- каталитический;
- метод рассеивания примесей в атмосфере;
- сжигание органических примесей.

Нормирование содержания вредных веществ в воздухе рабочей зоны производственных помещений и на территории предприятий осуществляется по ГОСТ 17.2.302-88.

При обработке деталей, связанной с применением водоструйных очищающих устройств, возникает *необходимость очистки стоков от мелкой абразивной пыли*.

Очистка от механических примесей может осуществляться следующими способами:

- процеживание:
- фильтрование:
- обработка в поле действия центробежных сил.

Если в воде содержатся маслосодержащие примеси, в зависимости от состава и концентрации примесей, очистка осуществляется следующими способами: отстаиванием, флотацией, фильтрацией, обработка в поле действия центробежных сил.

Гидросфера, использование и охрана вод нормируется ГОСТ 17.0.0.01-76.

5.2.2. Расчет воздушного пылевого фильтра

Участок сверления и рубки печатных плат производит выброс стеклопластиковой пыли – вещества 3 класса опасности по ГОСТ 12.1.007-76 («Вредные вещества. Классификация и общие требования безопасности»).

Для защиты рабочего персонала от воздействия примеси в помещении участка применяется вентиляционная система со скоростью движения воздуха в вытяжных столах 1.5 м/сек и общим расходом очищаемого воздуха на участке сверления и рубки печатных плат $L=18000 \text{ м}^3/\text{ч}$.

После вытяжки загрязненного воздуха системой местной вентиляции, его перед выбросом в атмосферу необходимо очистить. Перед очисткой воздуха от вредных примесей его следует очистить от пыли. Наиболее эффективно очистку можно выполнить при помощи фильтров (см. [9], [10]).

Проведем расчет мощности электродвигателя вентилятора, необходимого для просасывания воздуха через фильтр (см. [10]).

Мощность электродвигателя рассчитывается по формуле

$$N = \frac{K_i \cdot V \cdot P}{3600 \cdot t \cdot n}, \quad (5.1)$$

где

K_i — коэффициент запаса мощности на пусковой момент;

V — объем продуваемого воздуха, м³/час;

P — гидравлическое сопротивление фильтра, кг/(м*с);

n — КПД вентилятора;

t — продолжительность фильтрования (600 сек.);

3600 — коэффициент перевода часов в секунды.

Гидравлическое сопротивление фильтра определяется по формуле:

$$P = P_1 + P_2, \quad (5.2)$$

где

P_1 — гидравлическое сопротивление самого фильтра;

P_2 — гидравлическое сопротивление пыли, оседающей на фильтре.

Гидравлическое сопротивление фильтра — величина постоянная и определяется по формуле

$$P_1 = K_n \cdot W, \quad (5.3)$$

где

K_n — коэффициент гидравлического сопротивления материала фильтра, кг/м²;

W — скорость движения воздуха, м/с.

Гидравлическое сопротивление пыли, оседающей на фильтре, зависит от времени и рассчитывается по формуле:

$$P_2 = Z_n \cdot t \cdot W^2 \cdot K_p, \quad (5.4)$$

где

t — продолжительность цикла фильтрования;

W — скорость движения воздуха;

K_p — коэффициент гидравлического сопротивления пыли;

Z_n — концентрация пыли.

Для пыли, средний размер которой не превышает 10 мкм, $K_p=10^7$. Продолжительность фильтрования t примем равной 600 с. Скорость движения воздуха W возьмем равной 3,3125 м/с.

Остальные исходные данные примем следующими:

K_n для лавсанового фильтра равен $1,2 \cdot 10^5$ кг/м²; $K_i=1,1$; $n=0,7$; $V=1192,5$ м³/час. Z_n принимаем равным 10^{-5} кг/м³.

По формулам (5.3) и (5.4) рассчитаем значения гидравлических сопротивлений:

$$P_2 = 10^{-5} \cdot 600 \cdot 3,3125^2 \cdot 10^7 = 6,58 \cdot 10^5 \text{ кг/(м*с)};$$

$$P_1 = 1,2 \cdot 10^5 \cdot 3,3125 = 3,975 \cdot 10^5 \text{ кг/(м*с)}.$$

Отсюда в соответствии с формулой (5.2)

$$P = P_1 + P_2 = 1,0555 \cdot 10^6 \text{ кг/(м*с)}.$$

Теперь по формуле (5.1) можно определить требуемую мощность электродвигателя:

$$N = \frac{1,1 \cdot 1192,5 \cdot 1,0555 \cdot 10^6}{3600 \cdot 600 \cdot 0,7} = 915,7 \approx 916 \text{ Вт}.$$

Таким образом, следует выбрать электродвигатель номинальной мощностью порядка 0,92 кВт.

Заключение

В результате проектирования было разработано программное обеспечение контроллера пульта вахтера автоматизированной проходной АЛГО-421ПВ с возможностью работы в оффлайновом режиме.

В ходе работы над дипломным проектом была разработана технологическая документация: руководство пользователя пульта вахтера, руководство по записи программного продукта на носитель, руководство по программному тестированию компонентов контроллера.

Был проведен расчет затрат на разработку программного обеспечения.

Рассмотрены вопросы, связанные с охраной окружающей среды и безопасностью работы пользователя пульта вахтера автоматизированной проходной, произведены некоторые расчеты.

Предлагаемая программная система управления контроллером пульта вахтера автоматизированной проходной создана в соответствии с требованиями технического задания (обеспечено выполнение требуемых функций, а также режимов работы программной системы; учтены требования по надежности, программной и информационной совместимости).

Выводы:

- разработанное по теме дипломного проекта программное обеспечение (ПО) полностью *соответствует требованиям технического задания*;
- разработанное ПО в силу использования модульного подхода допускает *возможность расширения* путем добавления новых функций и модулей, а также *возможность варьирования* поведением программной системы путем изменения отдельных управляющих констант (например, количества работающих, количества проходов и ТК, кодов команд обмена с УК и т.д.).

Список литературы

1. CPU12 Reference Manual. Motorola Inc., 1996. (Руководство по системе команд микроконтроллера).
2. Motorola Semiconductor Technical Data.
MC68HC812A4: Technical Summary. Motorola Inc., 1996.
(Руководство по архитектуре и программной модели микроконтроллера).
3. А. А. Голов. Изделия и компоненты, предлагаемые фирмой КТЦ-МК. Справочник. Первое издание. Под ред. В.В.Березутского. М., издательство КТЦ-МК, 1998.
4. HI-CROSS+ Motorola HC12 Compiler. HIWARE, 1997.
(Руководство по кросс-компилятору).
5. HI-CROSS+ ELF Linker. HIWARE, 1997.
6. HI-CROSS+ HIWARE Tools. HIWARE, 1997. (Руководство по средствам разработки).
7. COSMIC v4.1x 68HC12 C Compiler Package - Evaluation Guidelines. Cosmic Software, 1997.
8. М. И. Болски. Язык программирования Си. Справочник. Пер. с англ. С. В. Денисенко. М., Радио и связь, 1988.
9. Охрана труда в машиностроении. Второе изд., перераб. и доп. Под редакцией д.т.н. проф. Е. Я. Юдина и д.т.н. проф. С. В. Белова. М., Машиностроение, 1983.
10. В. Н. Ужов, А. Ю. Вальдберг, Б. И. Мягков и др. Очистка промышленных газов от пыли. М., Химия, 1981.
11. Вредные вещества в промышленности. Справочник. Под ред. Н. В. Лазарева. М., Химия, 1971. Ч. I, II.

Приложения