Министерство образования и науки Российской федерации

Федеральное государственное учреждение высшего образования

РЫБИНСКИЙ ГОСУДАРСТВЕННЫЙ АВИАЦИОННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ имени П. А. СОЛОВЬЕВА

Институт Информационных технологий и систем управления

Кафедра вычислительных систем

Пояснительная записка

к программной части курсового проекта

по дисциплине «Микропроцессорные системы»

на тему «Устройство охранной сигнализации»

Студент группы ИВБ2-20. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .Баженова А. П.  
Руководитель канд. тех. наук, проф. . . . . . . . . . . . . . . . . . . . . Комаров В. М.  
Нормоконтролер канд. тех. наук, проф. . . . . . . . . . . . . . . . . .. . . . . Комаров В. М.

Рыбинск 2023

Содержание

[Введение 3](#_Toc136706964)

[1 Анализ технического задания 4](#_Toc136706965)

[2 Декомпозиция программы 6](#_Toc136706966)

[3 Разработка структуры данных программы 11](#_Toc136706967)

[4 Алгоритмизация программы 13](#_Toc136706978)

[5 Кодирование программы 35](#_Toc136706979)

[6 Тестирование и отладка программы 36](#_Toc136706980)

[7 Руководство пользователя 37](#_Toc136706981)

[Заключение 38](#_Toc136706982)

[Список использованных источников 39](#_Toc136706983)

[Приложение А 40](#_Toc136706984)

##### Введение

Использование микропроцессорной техники является одним из средств автоматизации технологических процессов. При современном уровне развития элементной базы вычислительной техники для построения устройств обработки информации целесообразно использовать микропроцессорные системы. Использование микропроцессоров обеспечивает возможность реализации таких устройств на одной небольшой печатной плате.

##### 1 Анализ технического задания

Из анализа технического задания следует, устройство должно обеспечивать:

1. формат здания – 2 этажа по 8 помещений;
2. постановку помещения под охрану (светодиодный индикатор соответствующего помещения загорается);
3. индикацию состояния каждого помещения с помощью светодиодных индикаторов;
4. ввод информации с датчиков обнаружения несанкционированного проникновения;
5. ввод информации с датчиков возникновения внештатной ситуации;
6. сигнализацию при обнаружении несанкционированного проникновения с помощью мигания светодиодного индикатора «Тревога» и включения звонка;
7. сигнализацию при возникновении внештатной ситуации с помощью мигания светодиодного индикатора «Тревога» и включения звонка;
8. прекращение мигания светодиодного индикатора при отсутствии тревоги или снятии помещения с охраны;
9. определение причины тревоги с помощью кнопки «Пожар» или кнопки «Взлом», при нажатии на которую загораются светодиодные индикаторы тех помещений, в которых обнаружен пожар или взлом;
10. имитацию показаний датчиков в среде «Design Microsystem» с помощью кнопок;

На основании перечисленных требований можно представить разрабатываемое устройство в виде «черной сферы» (рисунок 1.1), а также изобразить лицевую панель устройства (рисунок 1.2). Исходя из этих представлений начинается техническое проектирование устройства.



Рисунок 1.1 – Представление устройства охранной сигнализации в виде «черной сферы»



Рисунок 1.2 – Лицевая панель устройства охранной сигнализации

##### 2 Декомпозиция программы

Анализируя требования к программе с учётом лицевой панели устройства, получим исходную схему представления поставленной задачи.



Рисунок 2.1 – Исходная схема представления задачи «Устройство охранной сигнализации»

Из этой схемы следует, что проектируемая программа должна обрабатывать входную информацию с 32 датчиков пожара и взлома, выходная информация с которых имеет два значения: лог. «0» и лог. «1» и входную информацию с 18 кнопок с фиксацией, отображать выходную информацию на 19 двоичных индикаторах и управлять реле.

На первом этапе (рисунок 2.2) представим программу в виде трех задач: «Ввод информации», «Обработка входной информации», «Вывод информации». Подзадачи, выделенные на первом этапе, являются сложными и требуют дальнейшего разбиения.



Рисунок 2.2 – Статическая модель программы (после первого этапа декомпозиции)

На втором этапе (рисунки 2.3 – 2.5) подзадачу «Ввод информации» разобьем на подзадачи «Ввод с кнопок постановки под охрану», «Ввод с кнопок управления индикацией», «Ввод с датчиков пожара», «Ввод с датчиков взлома».

Подзадачу «Обработка входной информации» разобьем на подзадачи «Формирование массива охраняемых комнаты или комнат, в которых сработали датчики пожара», «Формирование массива комнат, в которых сработали датчики», «Формирование массива охраняемых комнаты, в которых сработали датчики взлома», «Формирование флага тревоги».

Подзадачу «Вывод информации» разобьем на подзадачи «Свечение индикаторов охраняемых комнат и всех комнат, где пожар», «Гашение индикаторов комнат, где сработали датчики», «Вывод на индикаторы режима отображения и управление звонком», «Вывод состояния датчиков пожара», «Вывод состояния датчиков взлома».

Подзадачи, выделенные на втором этапе, являются простыми и не требуют дальнейшей декомпозиции.



Рисунок 2.3 – Статическая модель программы – подзадача «Ввод информации» (после второго этапа декомпозиции)



Рисунок 2.4 – Статическая модель подзадачи «Обработка входной информации» (после второго этапа декомпозиции)



Рисунок 2.5 – Статическая модель подзадачи «Вывод информации» (после второго этапа декомпозиции)

##### 3 Разработка структуры данных программы

Процесс решения любой задачи состоит из активации тех или иных программных модулей, выполняющих некоторую подзадачу. Передача информации между модулями осуществляется с помощью наборов данных.

Дальнейшее проектирование программы заключается в переходе от ее статической модели к динамической, т. е. к ее алгоритмическому описанию.

| Наименование  данных | Символьное имя | Формат данных | Кодирование данных | | | | | | | | Приме-чание |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *D*7 | *D*6 | *D*5 | *D*4 | *D*3 | *D*2 | *D*1 | *D*0 |
| Образ кнопок управления индикацией | *CtrlBtnStatus* | Упако-ванный байт | 0 | 0 | 0 | 0 | 0 | 0 | \* | \* | \* - 0 (не нажата) или 1 (нажата) |
| Флаг тревоги | *AlarmFlag* | Флаговый байт | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | Пассивен  Активен |
| Флаг свечения индикаторов охраняемых комнат и всех комнат, где пожар | *IndFlag* | Флаговый байт | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | Пассивен  Активен |
| Флаг гашения индикаторов комнат, где сработали датчики | *FadeFlag* | Флаговый байт | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | 0  1 | Пассивен  Активен |
| Длительность включения звонка | *CountBell* | Слово | 0 | 0 | 0 | 0 | \* | \* | \* | \* | От 0 до 65535 |
| Длительность паузы при мигании индикаторов | *CountPause* | Слово | 0 | 0 | 0 | 0 | \* | \* | \* | \* | От 0 до 65535 |
| Длительность включения при мигании индикаторов | *CountPulse* | Слово | 0 | 0 | 0 | 0 | 0 | 0 | \* | \* | От 0 до 65535 |
| Состояние датчиков пожара | *SensFire-Status* | Массив (2 байта) | \* | \* | \* | \* | \* | \* | \* | \* | 1 – пожар в комнате |
| Состояние датчиков взлома | *SensBreach-Status* | Массив (2 байта) | \* | \* | \* | \* | \* | \* | \* | \* | 1 – взлом в комнате |
| Состояние кнопок постановки комнат под охрану | *SecureRoom-Btn* | Массив (2 байта) | \* | \* | \* | \* | \* | \* | \* | \* | 1 - нажата |
| Комнаты под охраной или комнаты, в которых пожар | *SecureOrFire-Status* | Массив (2 байта) | \* | \* | \* | \* | \* | \* | \* | \* | 1 - тревога |
| Комнаты, в которых сработали датчики | *BreachOr-FireStatus* | Массив (2 байта) | \* | \* | \* | \* | \* | \* | \* | \* | 1 - тревога |
| Комнаты под охраной, в которых взлома | *SecureAnd-BreachStatus* | Массив (2 байта) | \* | \* | \* | \* | \* | \* | \* | \* | 1 - тревога |

Таблица 3.1 – Структура данных

##### 4 Алгоритмизация программы

Вычислительный процесс, реализуемый программой, представляет собой последовательность дискретных шагов по преобразованию данных.

Для решения поставленной задачи необходимы следующие программные модули с соответствующими символьными именами:

– «Функциональная подготовка» (*FuncPrep*);

– «Ввод с кнопок постановки комнат под охрану» (*SecureButtonsInput*);

– «Ввод с кнопок управления отображением» (*CtrlBtnInput*);

– «Ввод с датчиков взлома» (*SensBreachInput*);

– «Ввод с датчиков пожара» (*SensFireInput*);

– «Формирование массива охраняемых комнаты или комнат, в которых сработали датчики пожара» (*SecureOrFireForm*);

– «Формирование массива комнат, в которых сработали датчики» (*BreachOrFireForm*);

– «Формирование массива охраняемых комнаты, в которых сработали датчики взлома» (*SecureAndBreachForm*);

– «Формирование флага тревоги» (*AlarmForm*);

– «Свечение индикаторов охраняемых комнат и всех комнат, где пожар» (*SecureOrFireInd*);

– «Гашение индикаторов комнат, где сработали датчики» (*BreachOrFireFade*);

– «Вывод на индикаторы режима отображения и управление звонком» (*ControlOutput*);

– «Вывод состояния датчиков взлома» (*BreachOutput*);

– «Вывод состояния датчиков пожара» (*FireOutput*);

Символьные имена присвоены соответствующим программным модулям с целью их дальнейшего использования. Выбранные имена отражают содержательный смысл этих модулей.

Последовательность модулей в логической конструкции «Следование» определяется логикой решения задачи.

Алгоритм макроуровня программы проектируемого устройства в одноуровневом представлении приведен на рисунке 4.1. Для повышения информативности ГСА на ней изображены входные и выходные данные, обрабатываемые каждым программным модулем. Это позволяет проследить, что каждый модуль располагается в той точке программного кольца, в которой существуют все обрабатываемые им данные.

Далее необходимо перейти к двухуровневому представлению алгоритма. Этот переход осуществляется формально. Алгоритм программы проектируемого устройства в двухуровневом представлении приведен на рисунке 4.2. Из этого алгоритма следует, что на макроуровне программы находятся лишь команды вызова программных модулей.

После разработки алгоритма макроуровня требуется разработать алгоритмы всех программных модулей нижнего уровня. Алгоритмы программных модулей и подмодулей приведены на рисунках 4.3 – 4.17.



Рисунок 4.1 – Алгоритм программы «Устройство охранной сигнализации» (одноуровневое представление)



Рисунок 4.2 – Алгоритм программы «Устройство охранной сигнализации» (двухуровневое представление)

*Модуль «Ввод с кнопок постановки под охрану» (SecureButtonsInput)*

Этот модуль служит для ввода состояния кнопок постановки под охрану с последующим размещением их образа в памяти.

ГСА этого модуля приведена на рисунке 4.3.

Электромеханическим переключателям свойственно явление дребезга контактов, который необходимо гасить. Гашение дребезга контактов можно выполнить программным путем. ГСА подмодуля гашения дребезга *VibrDelete* приведена на рисунке 4.4. Критерием окончания дребезга является считывание с кнопок одного и того же состояния заданное количество раз.





Рисунок 4.3 – Алгоритм модуля «Ввод с кнопок постановки под охрану» (общий алгоритм)



Рисунок 4.4 – Алгоритм подмодуля гашения дребезга

*Модуль «Ввод с кнопок управления индикацией» (CtrlBtnInput)*

Этот модуль служит для ввода состояния кнопок управления с последующим размещением его образа в памяти.

ГСА этого модуля приведена на рисунке 4.5.





Рисунок 4.5 – Алгоритм модуля «Ввод c кнопок управления индикацией» (общий алгоритм)

*Модуль «Ввод с датчиков пожара» (SensFireInput)*

Этот модуль служит для ввода состояния датчиков пожара с последующим размещением образа датчиков в памяти.

ГСА этого модуля приведена на рисунке 4.6.





Рисунок 4.6 – Алгоритм модуля «Ввод c датчиков пожара» (общий алгоритм)

*Модуль «Ввод с датчиков взлома» (SensBreachInput)*

Этот модуль служит для ввода состояния датчиков взлома с последующим размещением образа датчиков в памяти.

ГСА этого модуля приведена на рисунке 4.7.





Рисунок 4.7 – Алгоритм модуля «Ввод c датчиков взлома» (общий алгоритм)

*Модуль «Формирование массива комнат, в которых сработали датчики» (BreachOrFireForm)*

В комнатах находятся датчики пожара и датчики взлома, а для индикации тревожной ситуации путем мигания индикаторов достаточно, чтобы сработал хотя бы один из датчиков. Этот модуль предназначен для нахождения дизъюнкции значений битов состояния датчиков взлома и датчиков пожара.

ГСА модуля изображена на рисунке 4.8.





Рисунок 4.8 – Алгоритм модуля «Формирование, массива комнат в которых сработали датчики»

*Модуль «Формирование массива охраняемых комнаты или комнат, в которых сработали датчики пожара» (SecureOrFireForm)*

Этот модуль предназначен для объединения комнат, в которых сработали датчики пожара с комнатами, которые поставлены под охрану с помощью дизъюнкции значений битов массива состояния датчиков пожара и массива комнат, поставленных под охрану. Полученный массив используется для реализации мигания датчиков комнат, в которых сработали датчики.

ГСА модуля изображена на рисунке 4.9.





Рисунок 4.9 – Алгоритм модуля «Формирование массива охраняемых комнаты или комнат, в которых сработали датчики пожара»

*Модуль «Формирование массива охраняемых комнаты, в которых сработали датчики взлома» (SecureAndBreachForm)*

Этот модуль предназначен для вычисления пересечения комнат, в которых сработали датчики взлома с комнатами, которые поставлены под охрану с помощью конъюнкции значений битов массива состояния датчиков взлома и массива комнат, поставленных под охрану. Полученный массив используется для реализации мигания датчиков комнат, в которых сработали датчики.

ГСА модуля изображена на рисунке 4.10.





Рисунок 4.10 – Алгоритм модуля «Формирование массива охраняемых комнат, в которых сработали датчики взлома»

*Модуль «Формирование флага тревоги» (AlarmForm)*

Данный модуль формирует флаг тревоги, если хотя бы в одной из комнат сработал датчик пожара или хотя бы в одной из охраняемых комнат сработал датчик взлома.

ГСА модуля изображена на рисунке 4.11.





Рисунок 4.11 – Алгоритм модуля «Формирование флага тревоги»

Реализация мигания индикаторов

Для реализации мигания индикаторов комнат, в которых сработали датчики выделим период мигания и разобьем его на 2 интервала. В первом интервале все индикаторы комнат, поставленных под охрану, а также все индикаторы комнат, в которых сработали датчики пожара, независимо от того, поставлены ли они под охрану, светятся. Во втором интервале индикаторы тех комнат, в которых сработали датчики не светятся.

Для формирования временного интервала необходимо реализовать задержку. Задержка может быть реализована отдельной процедурой, выход из которой произойдет лишь после окончания задержки. Недостатком такой реализации является полный останов вычислительного процесса на время задержки, при этом программа перестанет реагировать на изменение каких-либо входных сигналов в течение времени задержки. Так как датчики могут сработать в любой момент времени, то будем считать полный останов вычислительного процесса на время задержки недопустимым и потребуем, чтобы программа реагировала на изменение входных сигналов в любой момент времени.

Для этого задержка должна быть реализована путем подсчета количества проходов программного кольца на макроуровне. Для реализации такого алгоритма задержки необходимо организовать счетчик проходов программного кольца, загрузить в него значение, соответствующее максимальной задержке, и на каждом проходе декрементировать этот счетчик с анализом его значения. При ненулевом значении счетчика проходов задержка продолжается, и не надо ничего менять. При нулевом значении счетчика проходов задержка закончилась, и необходимо перейти на другой интервал.

*Модуль «Свечение индикаторов охраняемых комнат и всех комнат, где пожар» (SecureOrFireInd)*

Для реализации счетчика проходов программного кольца используем переменную *PauseTime*.

При нулевом значении счетчика проходов задержка закончилась, и необходимо сбросить флаг формирования свечения индикаторов всех охраняемых комнат и комнат, где сработали датчики пожара и установить флаг формирования гашения индикаторов комнат, где сработали датчики.

После этого в счетчике проходов восстанавливается число, соответствующее максимальной задержке, и формирование времени задержки начинается сначала.

При первом проходе устанавливается флаг формирования свечения индикаторов всех охраняемых комнат и комнат, где сработали датчики пожара и сбрасывается флаг формирования гашения индикаторов комнат, где сработали датчики.

ГСА модуля изображена на рисунке 4.12.





Рисунок 4.12 – Алгоритм модуля «Свечение индикаторов охраняемых комнат и всех комнат, где пожар»

*Модуль «Гашение индикаторов комнат, где сработали датчики» (BreachOrFireFade)*

Для реализации счетчика проходов программного кольца используем переменную *PauseTime*.

При нулевом значении счетчика проходов задержка закончилась, и необходимо сбросить флаг формирования гашения индикаторов комнат, где сработали датчики и установить флаг формирования свечения индикаторов всех охраняемых комнат и комнат, где сработали датчики пожара.

После этого в счетчике проходов восстанавливается число, соответствующее максимальной задержке, и формирование времени задержки начинается сначала.

При первом проходе устанавливается флаг формирования свечения индикаторов всех охраняемых комнат и комнат, где сработали датчики пожара и сбрасывается флаг формирования гашения индикаторов комнат, где сработали датчики.

ГСА модуля изображена на рисунке 4.13.





Рисунок 4.13 – Алгоритм модуля «Гашение индикаторов комнат, где сработали датчики»

*Модуль «Вывод на индикаторы режима отображения и управление звонком» (ControlOutput)*

Этот модуль предназначен для индикации светодиодов режима управления при нажатии на кнопку «Взлом» или «Пожар», а также обеспечения мигания индикатора «Тревога» и включения звонка при взведенном флаге тревоги.

ГСА модуля изображена на рисунке 4.14.





Рисунок 4.14– Алгоритм модуля «Вывод на индикаторы режима отображения и управление звонком»

*Модуль «Вывод состояния датчиков пожара» (BreachOutput)*

Данный модуль выводит состояние датчиков пожара, независимо от того, поставлено это помещение под охрану или нет, если нажата кнопка отображения пожара.

ГСА модуля изображена на рисунке 4.15.





Рисунок 4.15 – Алгоритм модуля «Вывод состояния датчиков пожара»

*Модуль «Вывод состояния датчиков взлома» (FireOutput)*

Данный модуль выводит состояние датчиков пожара в тех помещениях, которые поставлены под охрану, если нажата кнопка отображения пожара.

ГСА модуля изображена на рисунке 4.16.





Рисунок 4.16 – Алгоритм модуля «Вывод состояния датчиков взлома»

*Модуль «Функциональная подготовка» (FuncPrep)*

Модуль предназначен для начальной установки некоторых наборов данных или служебных ячеек, которые служат для хранения вспомогательной информации. В данный момент известно, что в проектируемой программе нужна, по крайней мере, одна служебная ячейка, в которой будет храниться значение счетчика проходов программного кольца, определяющее время задержки.

В данном модуле в эту ячейку необходимо записать значение, соответствующее максимальной задержке.

ГСА модуля изображена на рисунке 4.17.



Рисунок 4.17 – Алгоритм модуля «Функциональная подготовка» (*FuncPrep*)

##### 5 Кодирование программы

Кодирование программы представляет собой запись алгоритмов проектируемой программы на ассемблере.

При кодировании программы используются ранее разработанные структура данных (таблица 3.1) и ГСА (рис. 4.1 – 4.17).

Полный исходный текст проектируемой программы, который приведен в Приложении А. В процессе трансляции и компоновки программы выявляются и исправляются все синтаксические ошибки, содержащиеся в ее исходном тексте.

После устранения синтаксических ошибок выполняется тестирование и отладка программы, в процессе которых выявляются и исправляются все семантические ошибки.

##### 6 Тестирование и отладка программы

Для тестирования и отладки разработанной программы будем использовать интегрированную программную среду *Demis*. Эта среда позволяет на архитектурном уровне представить проектируемое устройство и работать с ним как с реальным устройством.

В состав архитектуры интерфейса устройства входят 7 портов ввода, 3 порта вывода, 16 кнопок с фиксацией – имитация датчиков пожара, 16 кнопок с фиксацией – имитация датчиков взлома, 16 кнопок с фиксацией – кнопки постановки комнат под охрану, 16 двоичных индикаторов для индикации состояния комнат, 2 кнопки и 3 двоичных индикатора отображения типа тревожной ситуации, 1 вспомогательный двоичный индикатор для имитации включения звонка. Для составления архитектуры устройства выберем необходимые интерфейсные элементы и разместим их на экране, соединяя в соответствии с архитектурой. После этого установим интерфейсным элементам необходимые свойства:

* для портов ввода/вывода зададим адреса в соответствии с исходным текстом программы;
* для двоичных индикаторов выберем уровень зажигания, равный «Зажигается еденицей»;
* для кнопок определим исходное состояние, соответствующее ненажатой кнопке, как «нормально незамкнута

Далее с помощью текстового редактора занесем исходный текст программы.

После занесения программы выполним ее ассемблирование и компоновку, т. е. сборку программы, а затем выполним ее.

##### 7 Руководство пользователя

При включении устройства на короткое время включается звонок и загораются все двоичные индикаторы.

Светодиодные индикаторы состояния помещений погашены. При постановке помещения под охрану светодиодный индикатор соответствующего помещения загорается.

Светодиоды, отображающие тип тревожной ситуации, погашены и загораются при нажатии на соответствующие кнопки «Взлом» или «Пожар».

Звонок выключен.

Устройство опрашивает датчики пожара, взлома, кнопки управления отображением и кнопки постановки комнат под охрану.

При наличии пожара в любой из комнат или взлома в комнате, находящейся под охраной, начинает мигать светодиод соответствующей комнаты, светодиод «Тревога», периодически включается звонок. Если пожар потушен или взлом ликвидирован, звонок перестает звенеть, светодиод соответствующей комнаты перестает мигать, светодиод «Тревога» гаснет.

Для точного определения типа тревожной ситуации нужно нажать одну из кнопок «Пожар» или «Взлом», при нажатии на которую загораются светодиодные индикаторы тех помещений, в которых обнаружен пожар или взлом.

После отжатия соответствующей кнопки постановки комнаты под охрану светодиод соответствующей комнаты гаснет, и устройство снова находится в режиме опроса датчиков, кнопок управления отображением и кнопок постановки под охрану.

##### Заключение

В ходе выполнения курсового проекта мною было разработано устройство «Устройство охранной сигнализации». Из анализа технического задания была произведена постановка задачи. Далее была проделана декомпозиция поставленной задачи и разработана структура данных. После этого выполнена алгоритмизация программы и ее отдельных частей, написан текст разрабатываемой программы.

При выполнении поставленной задачи были соблюдены все требования технического задания.

##### Список использованных источников

1. Комаров В. М. Микропроцессорные системы: Учебное пособие /РГАТА.-Рыбинск, 2006. – 92 с.

2 Комаров В. М. Микропроцессорные системы. Проектирование аппаратного и программного обеспечения: Учебное пособие. – 2 изд. перераб. и доп. – Рыбинск, РГАТА, 2004. – 176 с.

##### Приложение А

.386

;Задайте объём ПЗУ в байтах

RomSize EQU 1000h ;4Кб для программного кода

NMax = 50 ;Число повторений при вводе с кнопки

SensFirePort1 = 00FEh ;Порт ввода состояния датчиков пожара - этаж 1

SensFirePort2 = 00FDh ;Порт ввода состояния датчиков пожара - этаж 2

SensBrechPort1 = 00FBh ;Порт ввода состояния датчиков взлома - этаж 1

SensBrechPort2 = 00F7h ;Порт ввода состояния датчиков взлома - этаж 2

SecureBtnPort1 = 00EFh ;Порт ввода кнопок постановки комнат под охрану - этаж 1

SecureBtnPort2 = 00DFh ;Порт ввода кнопок постановки комнат под охрану - этаж 2

CtrlBtnPort = 00BFh ;Порт ввода с кнопок управления индикацией

RoomIndPort1 = 00EFh ;Адрес 1-го порта вывода для двоичных индикаторов

;этаж 1

RoomIndPort2 = 00DFh ;Адрес 2-го порта вывода для двоичных индикаторов

;этаж 2

CtrlIndPort = 00BFh ;Порт управления звонком и индикаторами режимов

AlarmTime = 20 ;Длительность включения звонка

PulseTime = 20 ;Длительность включения при мигании индикаторов

PauseTime = 20 ;Длительность паузы при мигании индикаторов

Data SEGMENT use16 AT 0040h

;Здесь размещаются описания переменных

AlarmFlag db ? ;Флаг тревоги

IndFlag db ? ;Флаг формирования свечения индикаторов всех

;охраняемых комнат и комнат где

FadeFlag db ? ;Флаг формирования свечения индикаторов

;охраняемых комнат, в которых не сработали

;датчики

SensFireStatus db 2 dup (?) ;Состояние датчиков пожара

SensBreachStatus db 2 dup (?) ;Состояние датчиков взлома

SecureRoomBtn db 2 dup (?) ;Состояние кнопок постановки комнат под охрану

CtrlBtnStatus db ? ;Состояние кнопок управления индикацией

;и звонком - биты 0, 1, 2, 3

SecureOrFireStatus db 2 dup (?) ;Комнаты, находящиеся под охраной или комнаты

;в которых сработали датчики пожара

BreachOrFireStatus db 2 dup (?) ;Комнаты в которых сработали датчики

;1 этаж (мл. байт) и 2 этаж (ст. байт)

SecureAndBreachStatus db 2 dup (?) ;Комнаты находящиеся под охраной,

;в которых сработали датчики взлома

CountAlarm dw ? ;Длительность включения тревоги

CountPause dw ? ;Длительность паузы при мигании индикаторов

CountPulse dw ? ;Длительность включения при мигании

;индикаторов

Data ENDS

;Задайте необходимый адрес стека

Stk SEGMENT use16 AT 60h

;Задайте необходимый размер стека

dw 5 dup (?)

StkTop Label Word

Stk ENDS

InitData SEGMENT use16

InitDataStart:

;Здесь размещаются описания констант

InitDataEnd:

InitData ENDS

Code SEGMENT use16

ASSUME cs:Code,ds:Data,ss:Stk

;Здесь размещаются описания программных модулей

FuncPrep Proc

mov word ptr SecureRoomBtn,0 ;Вначале все кнопки не нажаты

mov word ptr SensFireStatus,0 ;Вначале все датчики пожара не активны

mov word ptr SensBreachStatus,0 ;Вначале все датчики взлома не активны

mov word ptr SecureOrFireStatus,0 ;все комнаты в порядке

mov word ptr BreachOrFireStatus, 0

mov word ptr SecureAndBreachStatus, 0

mov AlarmFlag,0 ;Флаг управления звонком

mov IndFlag, 0FFh ;Флаг формирования свечения установлен

mov FadeFlag,0 ;Флаг формирования гашения 2 сброшен

mov CountAlarm, AlarmTime ;Длительность включения звонка

mov CountPause, PauseTime ;Длительность паузы при мигании индикаторов

mov CountPulse, PulseTime ;Длительность включения при мигании

;индикаторов

call Flashing ;Сигнализация начала работы устройства

ret

FuncPrep Endp

Flashing PROC

;Выключить на короткое время звонок и сигнал на пульт пожарной охраны

mov al, 0FFh

out RoomIndPort1, al

out RoomIndPort2, al

out CtrlIndPort, al

mov dx, 0010h

mov bx, 0000h

DDelay0:

sub bx, 1

sbb dx, 0

mov si, bx

or si, dx

jnz DDelay0

mov al, 0

out RoomIndPort1, al

out RoomIndPort2, al

out CtrlIndPort, al

ret

Flashing ENDP

;Подмодуль "Гашение дребезга кнопок"

VibrDelete PROC NEAR

vd1:

mov ah,al ;Сохранение исходного состояния

mov cl,0 ;Сброс счётчика повторений

vd2:

in al,dx ;Ввод текущего состояния

cmp ah,al ;Текущее состояние=исходному?

jnz vd1 ;Переход, если нет

inc cl ;Инкремент счётчика повторений

cmp cl,NMax ;Конец дребезга?

jnz vd2 ;Переход, если нет

mov al,ah ;Восстановление местоположения данных

ret

VibrDelete ENDP

SecureButtonsInput Proc

;ввод состояния кнопок постановки комнат под охрану

lea bx,SecureRoomBtn ;Массив состояния кнопок

mov dx,SecureBtnPort1

in al,dx ;Ввод из порта

call VibrDelete ;Гашение дребезга

not al

mov [bx], al ;Сохранить состояние кнопок

;постановки комнат под охрану - этаж 1

mov dx, SecureBtnPort2

in al,dx ;Ввод из порта

call VibrDelete ;Гашение дребезга

not al

mov [bx+1], al ;Сохранить состояние кнопок

;постановки комнат под охрану - этаж 2

ret

SecureButtonsInput endp

CtrlBtnInput Proc

;ввод состояния кнопок управления индикацией

mov dx,CtrlBtnPort

in al,dx ;Ввод из порта

call VibrDelete ;Гашение дребезга

mov CtrlBtnStatus, al ;Сохранить состояние кнопок

ret

CtrlBtnInput Endp

SensBreachInput Proc

;ввод с датчиков взлома

lea si,SensBreachStatus ;Массив состояния датчиков

mov dx, SensBrechPort1 ;SensBrechPort1 - Адрес порта ввода (этаж 1)

in al,dx ;Ввод из порта

call VibrDelete ;Гашение дребезга

not al

mov [si], al ;Сохранить состояние датчиков взлома - этаж 1

mov dx, SensBrechPort2 ;SensBrechPort2 - Адрес порта ввода (этаж 2)

in al,dx ;Ввод из порта

call VibrDelete ;Гашение дребезга

not al

mov [si+1], al ;Сохранить состояние датчиков взлома - этаж 2

mov ax, Word ptr SensBreachStatus

ret

SensBreachInput endp

SensFireInput Proc

;ввод с датчиков пожара

lea si,SensFireStatus ;Массив состояния датчиков

mov dx, SensFirePort1 ;SensFirePort1 - Адрес порта ввода (этаж 1)

in al,dx ;Ввод из порта

call VibrDelete ;Гашение дребезга

not al

mov [si], al ;Сохранить состояние датчиков пожара - этаж 1

mov dx, SensFirePort2 ;SensFirePort2 - Адрес порта ввода (этаж 2)

in al,dx ;Ввод из порта

call VibrDelete ;Гашение дребезга

not al

mov [si+1], al ;Сохранить состояние датчиков пожара - этаж 2

mov ax, Word ptr SensFireStatus

ret

SensFireInput endp

SecureOrFireForm Proc

;формирование массива охраняемых комнат и всех комнат, где пожар

mov ax,word ptr SensFireStatus ;датчики пожара

mov dx,word ptr SecureRoomBtn ;охраняемые комнаты

or ax,dx

mov word ptr SecureOrFireStatus,ax ;сохранить

ret

SecureOrFireForm Endp

;формирование массива комнат где сработали датчики

BreachOrFireForm Proc

mov ax,word ptr SensFireStatus ;датчики пожара

mov dx,word ptr SensBreachStatus ;датчики взлома

or ax,dx

mov word ptr BreachOrFireStatus,ax ;сохранить комнаты где сработали датчики

ret

BreachOrFireForm Endp

SecureAndBreachForm PROC

mov ax,word ptr SensBreachStatus ;датчики взлома

mov dx,word ptr SecureRoomBtn ;комната под охраной

and ax,dx

mov word ptr SecureAndBreachStatus,ax ;сохранить охраняемые комнаты,

;где сработали датчики взлома

ret

SecureAndBreachForm ENDP

;Модуль "Формирование флаг тревоги"

AlarmForm Proc

cmp word ptr SensFireStatus, 0 ;есть пожар в любой из комнат

jnz m1 ;да

cmp word ptr SecureAndBreachStatus, 0 ;есть взлом в охраняемой комнате

jz AlarmOff ;нет

;да, реализовать временную задержку

m1:

dec word ptr CountAlarm ;декремент счетчика числа проходов

cmp word ptr CountAlarm,0 ;все проходы?

jnz AlarmOut ;нет, ничего не делать

;да, задержка закончилась

mov CountAlarm,AlarmTime ;перезагрузить счетчик числа проходов

not AlarmFlag ;Инвертировать флаг управления звонком

jmp AlarmOut ;На выход

AlarmOff:

mov AlarmFlag,0

AlarmOut:

ret

AlarmForm Endp

SecureOrFireInd PROC

;свечение индикаторов охраняемых комнат и всех комнат, где пожар

cmp IndFlag,0FFh

jnz IndOut

cmp CtrlBtnStatus, 0FFh ;нажаты кнопки "пожар" или "взлом"

jnz IndOut ;да, выходим

mov al, SecureOrFireStatus

out RoomIndPort1, al

mov al, SecureOrFireStatus+1

out RoomIndPort2, al

dec word ptr CountPause ;Декремент счетчика числа проходов

jnz IndOut ;Больше ничего не делаем. если не все проходы

;все проходы

mov IndFlag,0

mov FadeFlag,0FFh

mov CountPause,PauseTime ;Перезагрузить счетчик числа проходов

IndOut:

ret

SecureOrFireInd ENDP

BreachOrFireFade PROC

;свечение индикаторов охраняемых комнат, где нет тревоги

;гашение индикаторов комнат, где сработали датчики

cmp FadeFlag,0FFh

jnz FadeOut

cmp CtrlBtnStatus, 0FFh ;нажаты кнопки "пожар" или "взлом"

jnz FadeOut ;да

mov ax, word ptr BreachOrFireStatus

not ax

mov dx, word ptr SecureRoomBtn

and ax, dx

out RoomIndPort1, al ;выводим все охрвняемые комнаты,

;где не сработали датчики

xchg ah, al

out RoomIndPort2, al

dec word ptr CountPulse ;Декремент счетчика числа проходов

jnz FadeOut ;Ничего не делаем. если не все проходы

;все проходы

mov FadeFlag,0

mov IndFlag,0FFh

mov CountPulse,PulseTime ;Перезагрузить счетчик числа проходов

FadeOut:

ret

BreachOrFireFade ENDP

ControlOutput PROC

;Вывод на индикаторы режима отображения

mov al, CtrlBtnStatus

not al

cmp AlarmFlag, 0FFh ;Включить звонок?

jnz m1 ;Нет, не нужно включать

mov dl, 0Ch

or al, dl

m1: out CtrlIndPort, al

ret

ControlOutput ENDP

BreachOutput Proc

cmp CtrlBtnStatus, 0FEh

jnz m1

mov al, SecureAndBreachStatus

out RoomIndPort1, al

mov al, SecureAndBreachStatus+1

out RoomIndPort2, al

m1:

ret

BreachOutput ENDP

FireOutput Proc

cmp CtrlBtnStatus, 0FDh

jnz m1

mov al, SensFireStatus

out RoomIndPort1, al

mov al, SensFireStatus+1

out RoomIndPort2, al

m1:

ret

FireOutput ENDP

Start: ;системная подготовка

mov ax,Data

mov ds,ax

mov ax,Stk

mov ss,ax

lea sp,StkTop

;Здесь размещается код программы

call FuncPrep ;Функциональная подготовка

cycle:

;ввод

call SecureButtonsInput ;Ввод с кнопок постановки под охрану

call CtrlBtnInput ;Ввод с кнопок управления

call SensBreachInput ;ввод с датчиков взлома

call SensFireInput ;ввод с датчиков пожара

;обработка

call SecureOrFireForm ;формирование массива охраняемых комнаты или комнат,

;в которых сработали датчики пожара

call BreachOrFireForm ;формирование массива комнат,

;в которых сработали датчики

call SecureAndBreachForm ;формирование массива охраняемых комнаты,

;в которых сработали датчики взлома

call AlarmForm ;формирование флага тревоги

;вывод

call SecureOrFireInd ;свечение индикаторов охраняемых комнат и всех комнат, где пожар

call BreachOrFireFade ;пауза в свечении индикаторов комнат, где сработали датчики

call ControlOutput ;Вывод на индикаторы режима отображения и звонок

call BreachOutput ;вывод комнат при нажатии на кнопку "взлом"

call FireOutput ;вывод комнат при нажатии на кнопку "пожар"

jmp cycle ;зациклить

;В следующей строке необходимо указать смещение стартовой точки

org RomSize-16-((InitDataEnd-InitDataStart+15) and 0FFF0h)

ASSUME cs:NOTHING

jmp Far Ptr Start

Code ENDS

END Start