**Система контроля доступа – управляющий контроллер**

**Назначение**

#### Управление и контроль системой контроля доступа, построенной на автономных контроллерах по сети RS485.

#### Взаимосвязь с оператором осуществляется по сети Ethernet через WEB интерфейс .

#### Технические характеристики

#### Энергонезависимая память на 250 пользователей

#### Считыватель бесконтактных карт

#### Полное управление и контроль восьми автономными контроллерами, суммарно 16 проходов.

#### Политика прав доступа для каждого пользователя

#### База пользователей, включая ФИО каждого

#### Отчеты о последних событиях в системе с привязкой ко времени

#### Отчеты о базах автономных контроллеров

#### Управление, настройка , программирование с любого компьютера/планшета/телефона через WEB

#### Символьный дисплей с отображением состояния ситемы

#### Напряжение питания - 12 вольт 250мА.

#### Интерфейс управления автономных контроллеров и получения отчетов – MODBUS RTU RS485

#### Открытый исходный код и свободное программное обеспечение на языке PICµBASIC

#### Легкая настройка и возможность модификации под собственные нужды.

Аппаратная платформа - WEB PICµBASIC LOGIC CONTROLLER

#### 

#### 

#### Настройка и программирование

#### Контроллер  реализует функцию управления всей системой контроля доступа.

#### Начальные настройки WEB контроллера задаются при установке через терминал

#### - IP адрес и другие сетевые настройки

#### - логин/пароль для входа в систему и его включение/отключение.

#### В дальнейшем все настройки производятся через WEB интерфейс

#### Первое включение может занять несколько минут, пока будут создаваться пустые базы. Ход будет отображаться на индикаторе.

#### В браузере набрать <http://192.168.0.25/> , попадаем на экран ввода логина / пароля, если включено

#### 

#### 

#### Вводим логин пароль и попадаем на главную страницу, где отображается текущее состояние системы:

#### Адреса и доступность автономных контроллеров, состояние самого управляющего контроллера

#### Приступим к начальным настройкам – нажать «Установки»

#### При выключенной проверке логина/пароля попадаем на главную сразу

#### 

#### 

#### На этом экране отображается текущее время контроллера , имя контроллера, сетевые настройки.

#### Для установки времени/даты следует ввести значения и нажать «Применить»

#### Если контроллер занят, что отображается в окне статуса, следует дождаться завершения операций, нажать кнопку «Обновить».

#### 

#### Послу установки времени переходим к разделу «Пользователи»

#### Отображается пять окон с информацией.

#### Статус

#### В этом окне состояние контроллера и код ключа в считывателе управляющего контроллера, если он там есть

#### иначе будут прочерки.

#### Нажав кнопку «Обновить» обновляется информация о состоянии контроллера и коде ключа.

#### Для более простой работы, этот код также автоматически копируется в поля ввода других разделов

#### Добавить пользователя

#### Маленькое отступление

#### Вся основная информация о пользователях хранится в файлах на управляющем контроллере и также доступна через WEB интерфейс. В автономных контроллерах хранится база конкретно их пользователей, что позволяет всей системе надежно работать при потере связи между контроллерами и даже при отключении управляющего контроллера.

#### Также управляющий контроллер позволяет выполнять все настройки автономных контроллеров и управлять удаленными базами в подчиненных контроллерах.

#### 

#### На этой странице доступны следующие операции:

#### - добавление пользователя в базу управляющего контроллера

#### Для этого следует ввести ФИО пользователя и код ключа ( если он есть в локальном считывателе – скопируется автоматически) в разделе «Добавить пользователя» и нажать кнопку «Добавить»

#### Если такого пользователя нет и нет такого кода ключа, то он будет добавлен. Максимальное количество - 250 человек.

#### Операции с основной базой пользователей:

#### Поиск пользователя

#### Поиск может осуществляться по коду ключа, (если он есть в считывателе то будет скопирован) или по ФИО пользователя. Для поиска по ФИО достаточно ввести начальные символы ФИО, полностью можно не набирать.

#### Поиск по ФИО осуществляется, если поле кода ключа не заполнено или прочерки.

#### Операции с найденным пользователем:

#### Удаление из основной базы

#### Окно Удалить пользователя

#### Как правило, после удачного поиска, все поля заполняются автоматически, осталось только нажать кнопку «Удалить»

#### И пользователь будет удален из основной базы. Также поля могут быть заполнены вручную , при наличии информации.

#### Требуется точное соответствие ФИО и кода ключа.

#### Добавление / удаление пользователя в/из автономные контроллеры.

#### Найденного пользователя можно добавить или удалить в/из автономных баз автономных контроллеров.

#### Для этого требуется указать номер автономного контроллера для проведения действий, в случае добавления - указать права пользователя для данного автономного контроллера. Для разных контроллеров могут быть назначены разные права.

#### Привилегии (права) задаются числом, которое является суммой цифр в соответствующем поле. Подсказка как заполнять, указанна сразу на странице ниже.

#### Автономный контролер содержит только код ключа и права, ФИО в него не копируется.

#### Теперь согласно выбранному действию нажимаем кнопку «Добавить» или «Удалить»

#### После обработки запроса о результатах обработки или возникших ошибках (например такой пользовать уже существует) будет сообщено в окне статуса вверху страницы. Если обработка запроса еще не завершена то нажмем кнопку «Обновить» через несколько секунд.

#### Перед началом работы следует настроить каждый автономный контроллер на странице «Настройки контроллеров»

#### 

#### При загрузке страницы будет показан номер и адрес в сети MODBUS текущего выбранного контроллера и также его текущие настройки. При необходимости можно выбрать другой текущий контроллер установив переключатель с номером в строке «Выбрать контроллер» на нужный номер и нажать кнопку “OK”

#### Для текущего контроллера отображаются следующие настройки:

#### - Количество ключей в локальной базе автономного контроллера

#### - Время открытия двери

#### - Состояние блокировки кнопок управления на самом автономном контроллере

#### - Состояние блокировок считывателя

#### - Состояние блокировок дверей

#### Команды:

#### 

#### Сменить адрес контроллера – изменяется адрес, по которому производится доступ главным контроллером к подчиненному. Сам адрес подчиненного этой командой не меняется. Он меняется согласно инструкции самого автономного контроллера.

#### Блокировка кнопок – если ее включить, то кнопки на подчиненном контроллере перестанут работать, и соответственно, получив доступ к автономному контроллеру, злоумышленник не сможет изменить в нем настройки и изменить базу локальны пользователей.

#### Если включена – то автономный контроллер можно программировать автономно , согласно его инструкции.

#### Блокировка считывателей.

#### Автономный контроллер имеет два считывателя. Можно запретить / разрешить работу любого из них , независимо от прав пользователя, например для ограничения / разрешения доступа или за ненадобностью.

#### Блокировка дверей.

#### Можно заблокировать или снять блокировку на открытие любую дверь автономного контроллера. Блокировка имеет приоритет выше прав пользователя.

#### Дистанционно можно открыть любую дверь на запрограммированное время , например для пропуска человека без ключа.

#### Дистанционно можно открыть все двери, до следующего любого события. Двери будут находиться в открытом состоянии до отмены команды или до закрытия ключом доступа на местном считывателе.

#### Дистанционно можно отменить команду постоянного открытия двери.

#### Последние два режима используются при неординарных событиях или например при проведении погрузочно–разгрузочных работ - открываем дверь и беспрепятственно ходим. После окончания погрузки дверь закрываем дистанционно или приложив валидный ключ по месту к считывателю.

#### Все указанные команды выполняются нажатием кнопки в соответствующей строке раздела «Отправить команду»

#### Непосредственное управление пользователями автономного контроллера

#### Страница «Настройки пользователей»

#### 

#### Этот раздел будет полезен для новых автономных контроллеров, при переносе автономных контроллеров из одной системы доступа в другую , для создания полностью автономных контроллеров перед установкой и в других случаях.

#### Добавить / изменить права.

#### Операция производится непосредственно в автономном контроллере и не затрагивает базы главного контроллера.

#### Например, для настройки полностью автономных контроллеров или изменении прав существующих пользователей.

#### Следует указать код ключа(карты доступа) , номер контроллера и действие. Если пользователя не существует , он будет добавлен, если существует - то будут обновлены его права. При наличии ключа в локальном считывателе его код будет скопирован в поле ввода.

#### Удалить ключ

#### Аналогично заполняются поля. Действие - удаление ключа в базе автономного контроллера.

#### Удалить ключ по номеру.

#### Все ключи в автономных контроллерах нумеруются. Их полный список и нумерацию можно посмотреть в отчете базы автономного контроллера.

#### После добавления/удаления пользователя нумерация ключей пересчитывается, требуется загрузка обновленного отчета.

#### 

#### Это действие производится при физической утере карты доступа. Из отчета автономного контроллера выбираем номер нужного ключа и вводим. Ключ будет удален.

#### Удалить все ключи – полностью очищает всю базу ключей автономного контроллера.

#### Будьте осторожны, дополнительных вопросов не будет.

#### Просмотр отчетов, базы главного и баз автономных контроллеров

#### Страница «Отчеты»

#### 

#### Работа с базами данных и их пересылка по сети MODBUS может занимать некоторое время, иногда требуется дождаться окончания обработки и переход главного контроллера в состояние готовности. Кнопка «Обновить»

#### База пользователей

#### Доступно две операции просмотр и полная очистка

#### Просмотр показывает всех добавленный пользователей в главный контроллер.

#### «Очистить» - удаляет всех пользователей

#### Будьте осторожны, дополнительных вопросов не будет.

#### База автономного контроллера

#### Сначала требуется запросить данные нужного контроллера. После окончания обработки запроса можно ее просмотреть. Или сразу посмотреть загруженную последний раз базу.

#### 

#### Лог последних событий - отображаются все, включая ложные ключи и ключи других контроллеров

#### 

#### Код события:= сумма чисел

#### 10 ключ в 1 считывателе

#### 20 ключ во 2 считывателе

#### 40 ключ не опознан

#### 80 нажата кнопка

#### 01 открыта 1 дверь

#### 02 открыта 2 дверь

#### 03 открыто обе двери

#### Дистанционное управление Автономными контроллерами

#### Спецификация протокола обмена

#### Дистанционное управление и получение состояния для отчетов выполняется через интерфейс RS485 по протоколу MODBUS RTU.

Диапазон адресов – 1-127

#### Адрес по умолчанию = 16

#### Контроллер поддерживает следующие стандартные функции протокола MODBUS RTU:

**3 (0x03)** — чтение значений из одного или нескольких регистров хранения *(Read Holding Registers)*.

Запрос состоит из адреса первого элемента таблицы, значение которого требуется прочитать, и количества считываемых элементов. Адрес и количество данных задаются 16-битными числами, старший байт каждого из них передается первым.  
В ответе передаются запрошенные данные. Количество байт данных зависит от количества запрошенных элементов. Перед данными передается один байт, значение которого равно количеству байт данных.

Формат пакетов MODBUS

Запрос

1 байт – Адрес устройства

2 байт – Функция 0x03

3 байт - Адрес первой ячейки(параметра) (HIGH)

4 байт - Адрес первой ячейки(параметра) (LOW)

5 байт - Число ячеек(HIGH)

6 байт - Число ячеек(LOW)

7 байт - CRC (LOW)

8 байт - CRC (HIGH)

Ответ

1 байт – Адрес устройства

2 байт – Функция 0x03

3 байт - Счетчик байт данных

4 байт - Данные(addr) singed (HIGH)

5 байт - Данные(addr) singed (LOW)

--/ \*+1 байт --/ Данные(addr+1) singed (HIGH)

--/ \*+2 байт --/ Данные(addr+1) singed (LOW)

...

7/n-1 байт - CRC (LOW)

8/n байт - CRC (HIGH)

Поддерживается только чтение/запись одной ячейки за один цикл обмена

**16 (0x10)** — запись значений в несколько регистров хранения (Preset Multiple Registers)

Запрос

1 байт – Адрес устройства

2 байт – Функция 0x10

3 байт - Адрес первой ячейки(параметра) (HIGH)

4 байт - Адрес первой ячейки(параметра) (LOW)

5 байт - Число ячеек (параметра) (HIGH)

6 байт - Число ячеек (параметра) (LOW)

7 байт - Счетчик байт данных

8 байт - Данные(addr) singed (HIGH)

9 байт - Данные(addr) singed (LOW)

--/ \*+1 байт --/ Данные(addr+1) singed (HIGH)

--/ \*+2 байт --/ Данные(addr+1) singed (LOW)

...

7/n-1 байт - CRC (LOW)

8/n байт - CRC (HIGH)

Ответ

1 байт – Адрес устройства

2 байт – Функция 0x06

3 байт - Адрес первой ячейки(параметра) (HIGH)

4 байт - Адрес первой ячейки(параметра) (LOW)

5 байт - Число ячеек (параметра) (HIGH)

6 байт - Число ячеек (параметра) (LOW)

7 байт - CRC (LOW)

8 байт - CRC (HIGH)

Поддерживается только чтение/запись одной ячейки за один цикл обмена

**17 (0x11)** — Чтение информации об устройстве (Report Slave ID)

Пример

-> ADR, 0x11, CRC\_L, CRC\_H

<- ARD, 0x11, кол-во байт, text- xx BYTES , CRC\_L, CRC\_H

Запрос

1 байт – Адрес устройства

2 байт – Функция 0x11

3 байт - CRC (LOW)

4 байт - CRC (HIGH)

Ответ

1 байт – Адрес устройства

2 байт – Функция 0x11

3 байт - Счетчик байт данных

4 байт и далее – Данные TEXT max 64 bute

...

/n-1 байт - CRC (LOW)

/n байт - CRC (HIGH)

**CRC-16 -**  циклически избыточный код - полином A001h

Физический интерфейс – RS485 115200 8N1

**Временные диаграммы**

Обработка запросов с функциями **3 (0x03),** **16 (0x10), 17 (0x11)** всегда выполняется в фонововом режиме, независимо от программы в контроллрере. Одновременно может выполняться только один запрос. Минимальное время тишины  1,5 ms. Минимальное время до начала ответа – 0mS, типичное 1mS, максимальное 200 mS (если контроллер занят функциями работы с другими критичными во времени интерфейсами). Время ожидания ответа мастером до ошибки таймаута - 500 mS.

**Номера адресов ячеек и их значение**

Структура данных и алгоритм приема команд от MODBUS MASTER

СТРУКТУРА 8 байт - команды / запросы от мастера - #(xxx) – адрес ячейки

#(976) - код запроса команды

#(977) - статус выполнения команды

#(978),#(979),#(980),#(981),#(982),#(983) - данные

МАСТЕР - сначала записывает необходимые данные, потом записывает код запроса.

Контроллер (Слейв) ожидает код запроса , обрабатывает данные, обнуляет код запроса команды

записывает данные в ячейки, записывает статус выполнения команды.

Мастер должен прочесть статус выполнения, при его наличии забрать данные , обнулить статус выполнения

**Расшифровка кодов запроса и данных**

#(976) - код запроса команды (#(xxx) адрес и допустимые значения)

0 - Нет команды Slave ничего не делает

1 - Добавить ключ(обновить) - данные #(978)-#(983)

#(978) – пава доступа 0-63

#(979),#(980),#(981),#(982),#(983) код карточки 0-255 в каждой ячейке

Если код карточки совпадает а права нет – права будут обновлены для текущего номера

2 - Удалить ключ - данные #(978)-#(983)

#(979),#(980),#(981),#(982),#(983) код карточки 0-255 в каждой ячейке

3 - Удалить все ключи

4 - Установить время - данные #(978)

#(978) время \* 0,1 сек = 0-255; 0- запрос времени

ответ в #(978)

5 - Запрос статуса

Ответ

Количество ключей - #(978) 0-170

Время - #(979) 1-255

Блокировка кнопок - #(980) 0/1

Блокировка считывателя - #(981) 0/1/2/3

Блокировка двери - #(982) 0/1/2/3

Состояние дверей - #(983) 0/1/2/3

6 - Получить ключ по номеру диапазон =(1 – количество ключей)

запрос данные номера #(978)

ответ

#(978) – пава доступа 0-63

#(979),#(980),#(981),#(982),#(983) код карточки 0-255 в каждой ячейке

7 - заблокировать/разблокировать работу кнопок на устройстве #(978)

данные #(978) = 1- lock 0 = unlock

8 - заблокировать считыватель #(978)

данные #(978) 0 = unlock 1,2,3 -lock 1 – первый 2 – второй 3 - оба

9 - заблокировать дверь #(978)

данные #(978) = 0-unlock 1,2,3 -lock

10 - открыть дверь #(978)

данные #(978) = 1,2,3

11 - открыть двери до следующей команды/ события (alarm)

12 - отменить команду 11

13 - перезагрузка

14 - удалить ключ по номеру диапазон =(1 – количество ключей)

запрос данные номера #(978)

#(977) - статус выполнения команды = 1 - 31 команда выполнена успешно 32-64 - ошибка

Подробная расшифровка

0 - действий не производилось

1 - команда выполнена успешно

2 - ключ добавлен

3 - ключ переписан (обновлены права)

4 - ключ удален

5 - все ключи удалены

6 - время установлено / выполнено

32 - не выполнено

33 - команда не распознана

34 - нет места

35 - нет в базе

36 - есть в базе

37 - ошибка входных данных

**Данные для ведения отчетов**

Структура 8 байт - для ведения логов

#(984) - код данных

#(985) - статус выполнения

#(986),#(987),#(988),#(989),#(990),#(991) - данные

Контроллер (Slave) – записывает данные, потом код данных.

Мастер - опрашивает код, забирает данные данных, обнуляет код данных, записывает статус выполнения

#(984) - код данных расшифровка

0 - нет данных

Код события:= сумма чисел

01 открыта 1 дверь

02 открыта 2 дверь

(03 открыто обе двери)

10 ключ в 1 считывателе

20 ключ во 2 считывателе

40 ключ не опознан

80 нажата кнопка

#(986) - права доступа, если ключ опознан 0-63

#(987),#(988),#(989),#(990),#(991) - код карточки 0-255 в каждой ячейке , если не кнопка

**Приложения**

Программа контроллера

*0000 NOERR 2*

*0001 GOTO 456*

*0002 CLR:CLOSE:REM 'ОЧИСТКА ВСЕГО*

*0003 BEEP 15,5:PAUSE 200:BEEP 15,5:PAUSE 200:BEEP 15,5*

*0004 NOERR 2*

*0005 STOP*

*0006 GOTO 456*

*0007 §=DIR{"base.htm"}:IF §=1 THEN GOTO 18*

*0008 CREATE 32,11057,"base.htm"*

*0009 OPENWR "base.htm"*

*0010 FPRINT "<PRE> База пользователей"*

*0011 FPRINT "Ключ,";*

*0012 FPRINT "ФИО пользователя"*

*0013 FOR §=0 TO 249*

*0014 FPRINT " ,";*

*0015 FPRINT " "*

*0016 NEXT §*

*0017 FPRINT "</PRE>":F=49:CLOSE*

*0018 GOTO 201*

*0019 OPENRD "base.htm"*

*0020 FOR F=0 TO 249*

*0021 §=F\*44+49:SEEK §:FREAD@ §*

*0022 IF §=8224 THEN F=F\*44+49:EXFOR 25*

*0023 NEXT F*

*0024 F=0*

*0025 CLOSE*

*0026 RETURN*

*0027 VPRINT $(570) @(987),@(988),@(989),@(990),@(991);*

*0028 DATA $(520)," "*

*0029 #(520)=0*

*0030 OPENRD "base.htm"*

*0031 FOR G=0 TO 249*

*0032 H=G\*44+49:SEEK H:FREAD$ 10,$(555)*

*0033 IF #(556)=32 THEN GOTO 36*

*0034 I=SCOMP${$(570),$(555)}*

*0035 IF I=1 THEN EXFOR 38*

*0036 NEXT G*

*0037 GOTO 40*

*0038 H=G\*44+49+11*

*0039 SEEK H:FREAD$ 31,$(520):#(520)=31*

*0040 CLOSE*

*0041 RETURN*

*0042 IF A>=32497 THEN A=97*

*0043 IF A<97 THEN A=97*

*0044 OPENWR "log.htm":SEEK A*

*0045 SDT $(0):FPRINT %3,z,",";*

*0046 FPRINT %2,#(984),",";*

*0047 FPRINT \*$(0),",";*

*0048 FPRINT %2,#(986),",";*

*0049 FPRINT @(987),@(988),@(989),@(990),@(991),",";*

*0050 K=LOC{}:CLOSE*

*0051 GOSUB 27*

*0052 IF #(520)=31 THEN GOTO 55*

*0053 DATA $(520)," "*

*0054 #(520)=31*

*0055 OPENWR "log.htm":SEEK K*

*0056 FPRINT$ #(520)*

*0057 A=LOC{}:A=A+2:IF A>=32497 THEN A=97*

*0058 CLOSE*

*0059 RETURN*

*0060 §=DIR{"log.htm"}:IF §=1 THEN GOTO 78*

*0061 CREATE 32,32505,"log.htm"*

*0062 OPENWR "log.htm"*

*0063 FPRINT "<PRE> Лог событий"*

*0064 FPRINT "Адрес контроллера,";*

*0065 FPRINT "Код События,";*

*0066 FPRINT "Время-Дата,";*

*0067 FPRINT "Права доступа,";*

*0068 FPRINT "Ключ,";*

*0069 FPRINT "ФИО пользователя"*

*0070 FOR §=0 TO 449*

*0071 FPRINT " , , ";*

*0072 FPRINT ", , ,";*

*0073 FPRINT " "*

*0074 NEXT §*

*0075 FPRINT "</PRE>"*

*0076 A=97*

*0077 GOTO 84*

*0078 OPENRD "log.htm"*

*0079 FOR A=0 TO 449*

*0080 §=A\*72+97:SEEK §:FREAD@ §*

*0081 IF §=8224 THEN A=A\*72+97:EXFOR 84*

*0082 NEXT A*

*0083 A=97*

*0084 CLOSE:GOSUB 201*

*0085 RETURN*

*0086 §=DIR{"base\_kn.htm"}:IF §=1 THEN GOTO 88*

*0087 CREATE 32,8755,"base\_kn.htm"*

*0088 OPENWR "base\_kn.htm"*

*0089 FPRINT "<PRE> База контроллера #0."*

*0090 FPRINT "Номер ключа,";*

*0091 FPRINT "Права доступа,";*

*0092 FPRINT "Ключ,";*

*0093 FPRINT "ФИО пользователя"*

*0094 FOR §=0 TO 169*

*0095 FPRINT " , , ,";*

*0096 FPRINT " "*

*0097 NEXT §*

*0098 FPRINT "</PRE>":CLOSE*

*0099 RETURN*

*0100 #(750)=RFID{1}*

*0101 IF #(750)=0 THEN GOTO 219*

*0102 VPRINT $(720) @(751),@(752),@(753),@(754),@(755);*

*0103 RETURN*

*0104 j=C*

*0105 IF C=1 THEN i=a:GOTO 114*

*0106 IF C=2 THEN i=b:GOTO 114*

*0107 IF C=3 THEN i=c:GOTO 114*

*0108 IF C=4 THEN i=d:GOTO 114*

*0109 IF C=5 THEN i=e:GOTO 114*

*0110 IF C=6 THEN i=f:GOTO 114*

*0111 IF C=7 THEN i=g:GOTO 114*

*0112 IF C=8 THEN i=h:GOTO 114*

*0113 i=0*

*0114 RETURN*

*0115 FOR §=0 TO 100*

*0116 PAUSE 10:§=MDRD{i,#(977)}:IF §=0 THEN GOTO 118*

*0117 IF #(977)>0 THEN §-:EXFOR 120*

*0118 NEXT §*

*0119 #(977)=32*

*0120 §=#(977):#(977)=0:§=MDWR{i,#(977)}*

*0121 IF §=0 THEN §-:§=38*

*0122 #(977)=§*

*0123 RETURN*

*0124 k=0:l=0:m=0:n=0:o=0*

*0125 IF i=0 THEN GOTO 216*

*0126 IF i>127 THEN GOTO 216*

*0127 #(977)=0:§=MDWR{i,#(977)}*

*0128 IF §=0 THEN GOTO 218*

*0129 #(976)=5:§=MDWR{i,#(976)}*

*0130 IF §=0 THEN GOTO 218*

*0131 GOSUB 115*

*0132 IF #(977)>31 THEN GOTO 141*

*0133 §=MDRD{i,#(978)}:IF §=0 THEN GOTO 218*

*0134 §=MDRD{i,#(979)}:IF §=0 THEN GOTO 218*

*0135 §=MDRD{i,#(980)}:IF §=0 THEN GOTO 218*

*0136 §=MDRD{i,#(981)}:IF §=0 THEN GOTO 218*

*0137 §=MDRD{i,#(982)}:IF §=0 THEN GOTO 218*

*0138 k=#(978):l=#(979):m=#(980):n=#(981):o=#(982)*

*0139 GOSUB 201*

*0140 RETURN*

*0141 IF #(977)=1 THEN GOTO 201*

*0142 IF #(977)=2 THEN GOTO 202*

*0143 IF #(977)=3 THEN GOTO 203*

*0144 IF #(977)=4 THEN GOTO 204*

*0145 IF #(977)=5 THEN GOTO 205*

*0146 IF #(977)=6 THEN GOTO 206*

*0147 IF #(977)<32 THEN GOTO 201*

*0148 IF #(977)=33 THEN GOTO 208*

*0149 IF #(977)=34 THEN GOTO 211*

*0150 IF #(977)=35 THEN GOTO 212*

*0151 IF #(977)=36 THEN GOTO 214*

*0152 IF #(977)=37 THEN GOTO 209*

*0153 IF #(977)=38 THEN GOTO 218*

*0154 IF #(977)=39 THEN GOTO 216*

*0155 IF #(977)=40 THEN GOTO 213*

*0156 GOTO 210*

*0157 z=a:B=1*

*0158 GOSUB 174*

*0159 z=b:B=2*

*0160 GOSUB 174*

*0161 z=c:B=3*

*0162 GOSUB 174*

*0163 z=d:B=4*

*0164 GOSUB 174*

*0165 z=e:B=5*

*0166 GOSUB 174*

*0167 z=f:B=6*

*0168 GOSUB 174*

*0169 z=g:B=7*

*0170 GOSUB 174*

*0171 z=h:B=8*

*0172 GOSUB 174*

*0173 RETURN*

*0174 IF z=0 THEN GOTO 196*

*0175 IF z>127 THEN GOTO 196*

*0176 §=MDRD{z,#(984)}:IF §=0 THEN GOTO 193*

*0177 IF #(984)=0 THEN GOTO 190*

*0178 §=MDRD{z,#(986)}:IF §=0 THEN GOTO 193*

*0179 §=MDRD{z,#(987)}:IF §=0 THEN GOTO 193*

*0180 §=MDRD{z,#(988)}:IF §=0 THEN GOTO 193*

*0181 §=MDRD{z,#(989)}:IF §=0 THEN GOTO 193*

*0182 §=MDRD{z,#(990)}:IF §=0 THEN GOTO 193*

*0183 §=MDRD{z,#(991)}:IF §=0 THEN GOTO 193*

*0184 CLS:LPRINT "Dev=",z," Kod=",#(984)*

*0185 AT 64:LPRINT "KEY="*

*0186 LPRINT @(986),@(987),@(988),@(989),@(990),@(991)*

*0187 GOSUB 42*

*0188 #(984)=0:§=MDWR{16,#(984)}*

*0189 IF §=0 THEN GOTO 193*

*0190 CASE B,1,220,2,223,3,226,4,229*

*0191 CASE B,5,232,6,235,7,238,8,241*

*0192 GOTO 198*

*0193 CASE B,1,222,2,225,3,228,4,231*

*0194 CASE B,5,234,6,237,7,240,8,243*

*0195 GOTO 198*

*0196 CASE B,1,221,2,224,3,227,4,230*

*0197 CASE B,5,233,6,236,7,239,8,242*

*0198 RETURN*

*0199 DATA $(735),"Готов ":RETURN*

*0200 DATA $(735),"Обработка ":RETURN*

*0201 DATA $(770),"Удачное завершение ":RETURN*

*0202 DATA $(770),"Ключ добавлен ":RETURN*

*0203 DATA $(770),"Ключ обновлен ":RETURN*

*0204 DATA $(770),"Ключ удален ":RETURN*

*0205 DATA $(770),"Все ключи удалены ":RETURN*

*0206 DATA $(770),"Время установлено ":RETURN*

*0207 DATA $(770),"Пользов. добавлен ":RETURN*

*0208 DATA $(770),"Ошибка команды ":RETURN*

*0209 DATA $(770),"Ошибка данных ":RETURN*

*0210 DATA $(770),"Не выполнено ":RETURN*

*0211 DATA $(770),"Нет места ":RETURN*

*0212 DATA $(770),"Ключ не найден ":RETURN*

*0213 DATA $(770),"Пользов. не найден ":RETURN*

*0214 DATA $(770),"Ключ есть в базе ":RETURN*

*0215 DATA $(770),"Пользов.есть в базе":RETURN*

*0216 DATA $(770),"Нет такого контрол.":RETURN*

*0217 DATA $(770),"Контроллер занят ":RETURN*

*0218 DATA $(770),"Нет связи с контр. ":RETURN*

*0219 DATA $(720),"- - - - -":RETURN*

*0220 DATA $(600),"Доступен ":RETURN*

*0221 DATA $(600),"Не подключен":RETURN*

*0222 DATA $(600),"Нет связи ":RETURN*

*0223 DATA $(615),"Доступен ":RETURN*

*0224 DATA $(615),"Не подключен":RETURN*

*0225 DATA $(615),"Нет связи ":RETURN*

*0226 DATA $(630),"Доступен ":RETURN*

*0227 DATA $(630),"Не подключен":RETURN*

*0228 DATA $(630),"Нет связи ":RETURN*

*0229 DATA $(645),"Доступен ":RETURN*

*0230 DATA $(645),"Не подключен":RETURN*

*0231 DATA $(645),"Нет связи ":RETURN*

*0232 DATA $(660),"Доступен ":RETURN*

*0233 DATA $(660),"Не подключен":RETURN*

*0234 DATA $(660),"Нет связи ":RETURN*

*0235 DATA $(675),"Доступен ":RETURN*

*0236 DATA $(675),"Не подключен":RETURN*

*0237 DATA $(675),"Нет связи ":RETURN*

*0238 DATA $(690),"Доступен ":RETURN*

*0239 DATA $(690),"Не подключен":RETURN*

*0240 DATA $(690),"Нет связи ":RETURN*

*0241 DATA $(705),"Доступен ":RETURN*

*0242 DATA $(705),"Не подключен":RETURN*

*0243 DATA $(705),"Нет связи ":RETURN*

*0244 GOSUB 104*

*0245 IF i=0 THEN GOTO 216*

*0246 IF i>127 THEN GOTO 216*

*0247 #(977)=0:§=MDWR{i,#(977)}:§-*

*0248 §=MDWR{i,#(978)}:IF §=0 THEN GOTO 218*

*0249 §=MDWR{i,#(976)}:IF §=0 THEN GOTO 218*

*0250 GOSUB 115*

*0251 IF #(977)>31 THEN GOTO 141*

*0252 §=#(977)*

*0253 GOSUB 124*

*0254 IF l=0 THEN §-:GOTO 210*

*0255 #(977)=§*

*0256 GOSUB 141*

*0257 RETURN*

*0258 GOSUB 104*

*0259 IF i=0 THEN GOTO 216*

*0260 IF i>127 THEN GOTO 216*

*0261 j=C:§=MDRD{i,#(976)}:IF §=0 THEN GOTO 218*

*0262 IF #(976)<>0 THEN GOTO 217*

*0263 GOSUB 88*

*0264 GOSUB 124*

*0265 IF l=0 THEN GOTO 218*

*0266 IF k=0 THEN GOTO 212*

*0267 IF k>170 THEN GOTO 209*

*0268 OPENWR "base\_kn.htm"*

*0269 SEEK 24:FPRINT %1,j;*

*0270 SEEK 77:z=0:REM ' z - СЧЕТЧИК ЦИКЛА*

*0271 z=z+1:IF z>k THEN GOTO 292*

*0272 #(978)=z:#(976)=6*

*0273 §=MDWR{i,#(978)}:IF §=0 THEN GOTO 218*

*0274 §=MDWR{i,#(976)}:IF §=0 THEN GOTO 218*

*0275 GOSUB 115*

*0276 IF #(977)>31 THEN GOTO 141*

*0277 §=MDRD{i,#(978)}:IF §=0 THEN GOTO 218*

*0278 §=MDRD{i,#(979)}:IF §=0 THEN GOTO 218*

*0279 §=MDRD{i,#(980)}:IF §=0 THEN GOTO 218*

*0280 §=MDRD{i,#(981)}:IF §=0 THEN GOTO 218*

*0281 §=MDRD{i,#(982)}:IF §=0 THEN GOTO 218*

*0282 §=MDRD{i,#(983)}:IF §=0 THEN GOTO 218*

*0283 FPRINT %3,z,",";*

*0284 FPRINT %2,#(978),",";*

*0285 FPRINT @(979),@(980),@(981),@(982),@(983),",";*

*0286 K=LOC{}:CLOSE*

*0287 #(987)=#(979):#(988)=#(980):#(989)=#(981)*

*0288 #(990)=#(982):#(991)=#(983):GOSUB 27*

*0289 OPENWR "base\_kn.htm":SEEK K*

*0290 #(520)=31:FPRINT$ #(520):I=LOC{}:I=I+2:SEEK I*

*0291 GOTO 271*

*0292 CLOSE:GOSUB 201*

*0293 RETURN*

*0294 j=C:GOSUB 104*

*0295 GOSUB 124*

*0296 RETURN*

*0297 j=C*

*0298 IF C=1 THEN i=E:a=E:WREE a,0:GOTO 124*

*0299 IF C=2 THEN i=E:b=E:WREE b,2:GOTO 124*

*0300 IF C=3 THEN i=E:c=E:WREE c,4:GOTO 124*

*0301 IF C=4 THEN i=E:d=E:WREE d,6:GOTO 124*

*0302 IF C=5 THEN i=E:e=E:WREE e,8:GOTO 124*

*0303 IF C=6 THEN i=E:f=E:WREE f,10:GOTO 124*

*0304 IF C=7 THEN i=E:g=E:WREE g,12:GOTO 124*

*0305 IF C=8 THEN i=E:h=E:WREE h,14:GOTO 124*

*0306 C=1:j=1:i=a:p=63*

*0307 GOSUB 124*

*0308 RETURN*

*0309 #(977)=0:§=MDWR{i,#(977)}:§-*

*0310 IF #(790)<>10 THEN GOTO 209*

*0311 GOSUB 104*

*0312 IF i=0 THEN GOTO 216*

*0313 IF i>127 THEN GOTO 216*

*0314 #(979)=VAL{@(790),1}*

*0315 #(980)=VAL{@(790),3}:#(981)=VAL{@(790),5}*

*0316 #(982)=VAL{@(790),7}:#(983)=VAL{@(790),9}*

*0317 §=MDWR{i,#(978)}:IF §=0 THEN GOTO 218*

*0318 §=MDWR{i,#(979)}:IF §=0 THEN GOTO 218*

*0319 §=MDWR{i,#(980)}:IF §=0 THEN GOTO 218*

*0320 §=MDWR{i,#(981)}:IF §=0 THEN GOTO 218*

*0321 §=MDWR{i,#(982)}:IF §=0 THEN GOTO 218*

*0322 §=MDWR{i,#(983)}:IF §=0 THEN GOTO 218*

*0323 §=MDWR{i,#(976)}:IF §=0 THEN GOTO 218*

*0324 GOSUB 115*

*0325 IF #(977)>31 THEN GOTO 141*

*0326 §=#(977)*

*0327 GOSUB 124*

*0328 IF l=0 THEN §-:GOTO 210*

*0329 #(977)=§*

*0330 GOSUB 141*

*0331 RETURN*

*0332 #(977)=0:§=MDWR{i,#(977)}:§-*

*0333 IF #(790)<>10 THEN GOTO 209*

*0334 IF #(825)<3 THEN GOTO 209*

*0335 IF #(825)>31 THEN GOTO 209*

*0336 #(570)=#(790):#(571)=#(791):#(572)=#(792):#(573)=#(793)*

*0337 #(574)=#(794):#(575)=#(795):#(576)=#(796):#(577)=#(797)*

*0338 #(578)=#(798):#(579)=#(799):#(580)=#(800):#(581)=#(801)*

*0339 GOSUB 28*

*0340 IF #(520)<>0 THEN GOTO 214*

*0341 OPENRD "base.htm"*

*0342 FOR G=0 TO 249*

*0343 H=G\*44+49+11*

*0344 SEEK H:FREAD$ 31,$(520):#(520)=31*

*0345 I=SCOMP${$(825),$(520)}*

*0346 IF I=1 THEN EXFOR 215*

*0347 NEXT G*

*0348 CLOSE*

*0349 GOSUB 19*

*0350 IF F=0 THEN GOTO 211*

*0351 OPENWR "base.htm"*

*0352 SEEK F*

*0353 FPRINT \*$(790),",";*

*0354 F=LOC{}*

*0355 FPRINT " ";*

*0356 SEEK F*

*0357 FPRINT \*$(825);*

*0358 CLOSE:GOTO 207*

*0359 DATA $(814),"- - - - -"*

*0360 DATA $(900)," "*

*0361 #(977)=0:§=MDWR{i,#(977)}:§-*

*0362 IF #(790)<>10 THEN GOTO 376*

*0363 #(570)=#(790):#(571)=#(791):#(572)=#(792):#(573)=#(793)*

*0364 #(574)=#(794):#(575)=#(795):#(576)=#(796):#(577)=#(797)*

*0365 #(578)=#(798):#(579)=#(799):#(580)=#(800):#(581)=#(801)*

*0366 GOSUB 28*

*0367 IF #(520)=0 THEN GOTO 213*

*0368 #(814)=#(790):#(815)=#(791):#(816)=#(792):#(817)=#(793)*

*0369 #(818)=#(794):#(819)=#(795):#(820)=#(796):#(821)=#(797)*

*0370 #(822)=#(798):#(823)=#(799):#(824)=#(800)*

*0371 H=520:I=900*

*0372 FOR §=0 TO 31*

*0373 #(I)=#(H):H=H+1:I=I+1*

*0374 NEXT §*

*0375 GOTO 201*

*0376 OPENRD "base.htm"*

*0377 FOR G=0 TO 249*

*0378 H=G\*44+49+11*

*0379 SEEK H:FREAD$ 31,$(520):#(520)=31*

*0380 I=SCOMP${$(825),$(520)}*

*0381 IF I=1 THEN EXFOR 385*

*0382 NEXT G*

*0383 CLOSE*

*0384 GOTO 213*

*0385 H=G\*44+49:SEEK H:FREAD$ 10,$(814):#(814)=10*

*0386 CLOSE*

*0387 IF #(815)=32 THEN GOTO 212*

*0388 H=520:I=900*

*0389 FOR §=0 TO 31*

*0390 #(I)=#(H):H=H+1:I=I+1*

*0391 NEXT §*

*0392 GOTO 201*

*0393 #(977)=0:§=MDWR{i,#(977)}:§-*

*0394 IF #(790)<>10 THEN GOTO 209*

*0395 IF #(825)<3 THEN GOTO 209*

*0396 IF #(825)>31 THEN GOTO 209*

*0397 #(570)=#(790):#(571)=#(791):#(572)=#(792):#(573)=#(793)*

*0398 #(574)=#(794):#(575)=#(795):#(576)=#(796):#(577)=#(797)*

*0399 #(578)=#(798):#(579)=#(799):#(580)=#(800):#(581)=#(801)*

*0400 GOSUB 28*

*0401 IF #(520)=0 THEN GOTO 212*

*0402 I=SCOMP${$(825),$(520)}*

*0403 IF I=0 THEN GOTO 213*

*0404 OPENWR "base.htm"*

*0405 H=G\*44+49:SEEK H:FPRINT " ,";*

*0406 FPRINT " "*

*0407 CLOSE:GOTO 204*

*0408 IF #(880)<>5 THEN GOTO 209*

*0409 IF #(888)<>10 THEN GOTO 209*

*0410 #(590)=VAL{$(880),1}:IF #(590)>23 THEN GOTO 209*

*0411 #(591)=VAL{$(880),4}:IF #(591)>59 THEN GOTO 209*

*0412 #(592)=VAL{$(888),3}:IF #(592)>50 THEN GOTO 209*

*0413 IF #(592)<17 THEN GOTO 209*

*0414 #(593)=VAL{$(888),6}:IF #(593)>12 THEN GOTO 209*

*0415 IF #(593)<1 THEN GOTO 209*

*0416 #(594)=VAL{$(888),9}:IF #(593)>31 THEN GOTO 209*

*0417 IF #(594)<1 THEN GOTO 209*

*0418 YEAR #(592):MONTH #(593):DAY #(594)*

*0419 HOUR #(590):MIN #(591):SEC 0*

*0420 SDT $(860)*

*0421 GOTO 206*

*0422 IF D=1 THEN GOTO 258*

*0423 IF D=2 THEN GOTO 294*

*0424 IF D=3 THEN #(978)=E:#(976)=4:GOTO 244*

*0425 IF D=4 THEN #(978)=1:#(976)=7:GOTO 244*

*0426 IF D=5 THEN #(978)=0:#(976)=7:GOTO 244*

*0427 IF D=6 THEN #(978)=1:#(976)=8:GOTO 244*

*0428 IF D=7 THEN #(978)=2:#(976)=8:GOTO 244*

*0429 IF D=8 THEN #(978)=3:#(976)=8:GOTO 244*

*0430 IF D=9 THEN #(978)=0:#(976)=8:GOTO 244*

*0431 IF D=10 THEN #(978)=1:#(976)=9:GOTO 244*

*0432 IF D=11 THEN #(978)=2:#(976)=9:GOTO 244*

*0433 IF D=12 THEN #(978)=3:#(976)=9:GOTO 244*

*0434 IF D=13 THEN #(978)=0:#(976)=9:GOTO 244*

*0435 IF D=14 THEN #(978)=1:#(976)=10:GOTO 244*

*0436 IF D=15 THEN #(978)=2:#(976)=10:GOTO 244*

*0437 IF D=16 THEN #(978)=3:#(976)=10:GOTO 244*

*0438 IF D=17 THEN D=0 GOTO 201*

*0439 IF D=18 THEN #(978)=0:#(976)=11:GOTO 244*

*0440 IF D=19 THEN #(978)=0:#(976)=12:GOTO 244*

*0441 IF D=20 THEN #(978)=0:#(976)=13:GOTO 244*

*0442 IF D=21 THEN GOTO 297*

*0443 IF D=22 THEN GOTO 62*

*0444 IF D=23 THEN GOTO 201*

*0445 IF D=24 THEN #(976)=1:#(978)=E:p=E:GOTO 309*

*0446 IF D=25 THEN #(976)=2:#(978)=255:GOTO 309*

*0447 IF D=26 THEN #(978)=E:#(976)=14:GOTO 244*

*0448 IF D=27 THEN #(978)=0:#(976)=3:GOTO 244*

*0449 IF D=28 THEN GOTO 332*

*0450 IF D=29 THEN GOTO 359*

*0451 IF D=30 THEN GOTO 9*

*0452 IF D=31 THEN GOTO 408*

*0453 IF D=32 THEN GOTO 393*

*0454 GOSUB 208*

*0455 RETURN*

*0456 LINIT 1:BEEP 15,2:UART 2,7:LPRINT "Starting..."*

*0457 a=RDEE{0}:b=RDEE{2}:c=RDEE{4}:d=RDEE{6}*

*0458 e=RDEE{8}:f=RDEE{10}:g=RDEE{12}:h=RDEE{14}*

*0459 IF a>=0 THEN IF a<127 THEN GOTO 463*

*0460 a=16:b=0:c=0:d=0:e=0:f=0:g=0:h=0*

*0461 WREE a,0:WREE b,2:WREE c,4:WREE d,6*

*0462 WREE e,8:WREE f,10:WREE g,12:WREE h,14*

*0463 GOSUB 60*

*0464 IPCFG #(235):LPRINT " IP="*

*0465 AT 64:LPRINT %3,#(244),".",#(245),".",#(246),".",#(247)*

*0466 q=#(244):r=#(245):s=#(246):t=#(247):v=#(248):x=#(249)*

*0467 GOSUB 7*

*0468 GOSUB 86*

*0469 GOSUB 219*

*0470 GOSUB 199*

*0471 GOSUB 201*

*0472 GOSUB 222*

*0473 IF a=0 THEN GOSUB 221*

*0474 GOSUB 225*

*0475 IF b=0 THEN GOSUB 224*

*0476 GOSUB 228*

*0477 IF c=0 THEN GOSUB 227*

*0478 GOSUB 231*

*0479 IF d=0 THEN GOSUB 230*

*0480 GOSUB 234*

*0481 IF e=0 THEN GOSUB 233*

*0482 GOSUB 237*

*0483 IF f=0 THEN GOSUB 236*

*0484 GOSUB 240*

*0485 IF g=0 THEN GOSUB 239*

*0486 GOSUB 243*

*0487 IF h=0 THEN GOSUB 242*

*0488 j=1:i=a:p=63*

*0489 GOSUB 124*

*0490 DATA $(814),"- - - - -"*

*0491 CLS:LPRINT "READY"*

*0492 GOTO 493*

*0493 GOSUB 100*

*0494 SDT $(860)*

*0495 GOSUB 157*

*0496 FOR §=0 TO 40*

*0497 IF D<>0 THEN §-:EXFOR 505*

*0498 PAUSE 10*

*0499 NEXT §*

*0500 AT 0:LPRINT "READY "*

*0501 STIME $(520):LPRINT \*$(520)*

*0502 AT 64:LPRINT " ":SDATE $(520)*

*0503 LPRINT \*$(520)," "*

*0504 GOTO 493*

*0505 GOSUB 200*

*0506 GOSUB 422*

*0507 CLOSE:GOSUB 199*

*0508 D=0:GOTO 493*