

# Система контроля доступа – автономный контроллер

## Назначение

Автономное управления доступом в помещения и управления замками, используя в качестве ключа бесконтактные радиочастотные (PROXIMITY) карты 125кГц стандарта EM-Marin.

## Технические характеристики

Энергонезависимая память на 170 пользователей

Два считывателя бесконтактных карт

Два выхода реле для управления замками

Программируемое время открытия 0,1-25 секунд с шагом 0,1 сек.

Световая и звуковая сигнализация

Кнопка открытия дверей

Политика прав доступа для каждого пользователя

Автономное программирование

Символьный дисплей и кнопки управления

Напряжение питания - 12 вольт 150мА.

Интерфейс дистанционного управления и получения отчетов – MODBUS RTU RS485

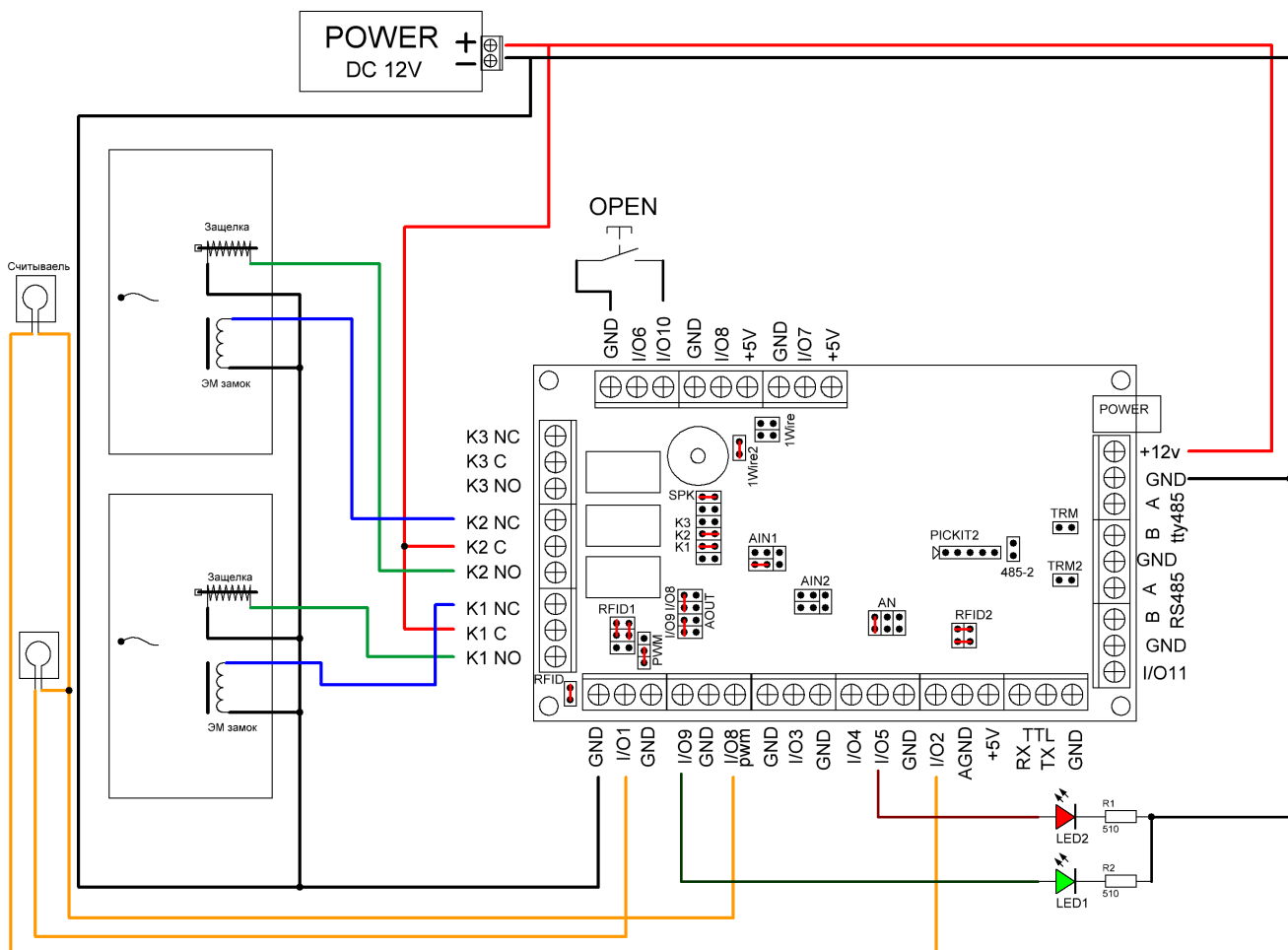
Открытый исходный код и свободное программное обеспечение на языке PICμBASIC

Легкая настройка и возможность модификации под собственные нужды.

Аппаратная платформа - ПРОГРАММИРУЕМЫЙ МОДУЛЬ РАСШИРЕНИЯ DEVICE PICμBASIC LOGIC CONTROLLER

<http://moycontroller.ru/products/programmiruemyj-modul-rasshireniya-basic>

## Схема подключения



Установленные перемычки показаны красным цветом

## Функционирование

Контроллер реализует функцию управления электромеханическим, электромагнитным, ригельным замком, электрозаслелкой. Под конкретную дверь подбирается оптимальный замок. Для принудительного закрытия двери может использоваться доводчик двери. Позволяет обслуживать один или два прохода.

### Рабочий режим

В рабочем режиме контроллер опрашивает считыватели. При опознавании карты производится проверка по внутренней базе ключей и прав доступа этой карты при совпадении.

Для каждой карты доступа можно устанавливать следующие привилегии:

- опознается в 1 считывателе
- опознается во 2 считывателе
- опознается в обоих считывателях

При опознании в 1 считывателе

- открыть первую дверь
- открыть вторую дверь
- открыть обе двери
- нет действий

При опознании в 2 считывателе

- открыть первую дверь
- открыть вторую дверь
- открыть обе двери
- нет действий

При опознании карточки звучит короткий сигнал высокого тона

При поднесении чужой карточки звучит длинный сигнал низкого тона

При открытии двери загорается зеленый светодиод и звучит серия звуковых сигналов до закрытия.

Звуковое подтверждение можно отключить.

При нажатии кнопки открываются обе двери.

На дисплее отображается состояние контроллера:

Готов - в режиме ожидания

Код ключа при опознании, действие контроллера – открыть/ не открывать дверь, время открытия.

### Режим программирования

Автономное программирование

Вход в меню - нажать любую кнопку - на дисплее общее количество ключей, затем переход в меню программирования

Пункты меню

Добавить ключ

Удалить ключ

Просмотр ключей

Установка времени открытия

Удаление всех ключей

Выход из режима программирования

Перезагрузка контроллера

Навигация по меню – кнопки влево/ вправо вверх/вниз. Выбор пункта меню – кнопка ОК

Если нет активности – через 10 секунд контроллер автоматически вернется в рабочий режим

### Добавление ключа

На дисплее приглашение поднести ключ к считывателю для добавления в базу, номер ячейки для записи, права пользователя. Время ожидания до выхода из меню – 20 секунд

Права пользователя - по умолчанию все разрешено = 63. Права описываются суммой цифр

1 – разрешен первый считыватель

2 – разрешен второй считыватель

4 – открыть первую дверь для первого считывателя

8 – открыть вторую дверь для первого считывателя

16 – открыть первую дверь для второго считывателя

32 – открыть вторую дверь для второго считывателя

Редактирование прав

В режиме добавления ключа

Кнопка влево перебирает права считывателя

Кнопка вправо перебирает права открытия двери от первого считывателя

Кнопка вверх перебирает права открытия двери от второго считывателя

Кнопка вниз – выход из редактирования прав.

Кнопка ОК выход из режима добавления ключей

После добавления ключа происходит возврат на добавление следующего ключа, с ранее установленными правами.

Возможные ошибки добавления отображаются на индикаторе

- Такой ключ уже есть
- Нет места в памяти

#### Удаление ключей

При входе в режим - на индикаторе отображение ожидания ключа, который нужно удалить.

При поднесении ключа он будет удален и контроллер вернется в меню на пункт удаления.

Время ожидания ключа на удаление до выхода из режима программирования -20 секунд

Выход из режима удаления - любая кнопка.

Возможные ошибки удаления отображаются на индикаторе

- Такого ключа нет в базе

#### Просмотр ключей

На индикаторе номер ключа, его код, права доступа

Просмотр следующего ключа - любая кнопка.

Возможные ошибки добавления отображаются на индикаторе

- Больше нет ключей для просмотра, при нажатии кнопки - вернется в меню

Время ожидания до автоматического выхода из меню не более 20 секунд.

#### Установка времени открытия

На индикаторе отображается текущее время открытия

Кнопки влево и вправо уменьшают /увеличивают время с шагом 0,1 сек

Кнопки вверх и вниз уменьшают /увеличивают время с шагом 1 сек

Кнопка ОК подтверждает значение, записывает в память и выход в меню

Время ожидания до автоматического выхода из меню не более 20 секунд, при этом значения не записываются

#### Удаление всех ключей

При нажатии кнопки ОК выдается приглашение «Вы уверены?»

Нажатие кнопки ОК удалит все ключи

Нажатие любой другой или автоматический выход по времени ожидания ключи не удаляет.

#### Перезагрузка

При выборе этого пункта и нажатии кнопки ОК контроллер будет перезагружен.

#### Выход из меню

Кнопка ОК возвращает в рабочий режим.

## Дистанционное управление и данные для отчета

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) signed (HIGH)  
5 байт - Данные(addr) signed (LOW)  
--/ \*+1 байт --/ Данные(addr+1) signed (HIGH)  
--/ \*+2 байт --/ Данные(addr+1) signed (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) signed (HIGH)  
9 байт – Данные(addr) signed (LOW)  
--/ \*+1 байт --/ Данные(addr+1) signed (HIGH)  
--/ \*+2 байт --/ Данные(addr+1) signed (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

-> ADDR, 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 byte  
...  
/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 фоновом режиме, независимо от программы в контроллере. 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 329
0002 CLR:CLOSE
0003 BEEP 15,5:PAUSE 200:BEEP 15,5:PAUSE 200:BEEP 15,5
0004 NOERR 2
0005 STOP
0006 GOTO 329
0007 §=GKEY{:}:IF §=0 THEN RETURN
0008 #(975)=RDEE{1021}:IF #(975)=0 THEN RETURN
0009 BEEP 15,2
0010 §=0:§=0:§=0:§=0
0011 §=GKEY{:}:IF §<>0 THEN GOTO 11
0012 GOSUB 167
0013 CLS:LPRINT "KEY in base =",§:PAUSE 1000
0014 CLS:LPRINT "Select MODE"
0015 IF A=0 THEN AT 64:LPRINT "ADD key      "
0016 IF A=1 THEN AT 64:LPRINT "DEL key      "
0017 IF A=2 THEN AT 64:LPRINT "VIEW key     "
0018 IF A=3 THEN AT 64:LPRINT "SEL Time open "
0019 IF A=4 THEN AT 64:LPRINT "DEL ALL key  "
0020 IF A=5 THEN AT 64:LPRINT "Reboot      "
0021 IF A=6 THEN AT 64:LPRINT "EXIT        "
0022 §=KEY{10}:BEEP 15,2
0023 CASE §,0,123,1,25,2,27,4,29,8,25,16,27
0024 GOTO 14
0025 A=A+1:IF A>6 THEN A=0
0026 GOTO 15
0027 A=A-1:IF A<0 THEN A=6
0028 GOTO 15
0029 B=63:CASE A,0,43,1,88,2,32,3,107
0030 CASE A,4,82,6,123
0031 §^:CLR:GOTO 329:REM "***** REBOOT
0032 B=0
0033 FOR §=B TO 169
0034 #(975)=RDEE{A*6}:IF #(975)<>255 THEN Г=§:EXFOR 37
0035 NEXT §
0036 AT 64:LPRINT "NO next key  ":PAUSE 1000:GOTO 14
0037 Г=B*6:RDPE *6,§,(960),Г
0038 CLS:LPRINT %3,"#KEY=",B,"Perm=",§,(960):AT 64
0039 LPRINT "KEY= ",@(961),@(962),@(963),@(964),@(965)
0040 §=KEY{15}:BEEP 15,2
0041 IF §=0 THEN GOTO 123
0042 B=B+1:GOTO 33
0043 PAUSE 20:§=GKEY{:}:IF §<>0 THEN GOTO 43
0044 CLS:LPRINT "ADD KEY"
0045 GOSUB 181
0046 Б=#(975):IF Б=255 THEN GOTO 81
0047 AT 7:LPRINT %3,"#",Б," R="
0048 PAUSE 20:§=GKEY{:}:IF §<>0 THEN GOTO 48
0049 AT 14:LPRINT %2,Б
0050 AT 64:LPRINT "WAIT KEY      "
0051 z=0
0052 z=z+1:IF z>100 THEN BEEP 15,2:GOTO 123
0053 §=GKEY{:}:IF §=0 THEN GOTO 67
0054 BEEP 15,2:z=0
0055 AT 64:Г=(Б&3):LPRINT %1,"in:",Г,"out:"
0056 Г=(Б&12)/4:LPRINT %1," A:",Г
0057 Г=(Б&48)/16:LPRINT %1," B:",Г
0058 Г=KEY{10}:BEEP 15,2
0059 CASE Г,16,61,1,63,8,65,4,14
0060 GOTO 48
0061 Г=(Б&3):Г=Г+1:Г=Г&3:Б=Б&60:Б=Б|Г
0062 AT 14:LPRINT %2,Б:GOTO 55
0063 Г=(Б&12):Г=Г+4:Г=Г&12:Б=Б&51:Б=Б|Г
0064 AT 14:LPRINT %2,Б:GOTO 55
0065 Г=(Б&48):Г=Г+16:Г=Г&48:Б=Б&15:Б=Б|Г
0066 AT 14:LPRINT %2,Б:GOTO 55
0067 #(968)=RFID{1}
0068 IF #(968)=1 THEN GOTO 72
0069 #(968)=RFID{2}
0070 IF #(968)=1 THEN GOTO 72
0071 GOTO 52
0072 GOSUB 172
0073 IF #(966)=255 THEN GOTO 76
0074 BEEP 5,8:AT 64
0075 LPRINT "THIS KEY PRESENT":PAUSE 1000:GOTO 43
0076 #(968)=Б:§=Б*6
0077 WRPE *6,§,(968),§:BEEP 15,2
0078 AT 64:LPRINT "DONE      "
0079 §=KEY{2}:BEEP 15,2:IF §>0 THEN GOTO 14
0080 GOTO 43
0081 AT 64:LPRINT "NO FREE MEMORY ":PAUSE 1000:GOTO 14
0082 CLS:LPRINT "DEL ALL KEY "
0083 AT 64:LPRINT "ARE YOU SURE ?"
0084 §=KEY{5}:BEEP 15,2:IF §<>4 THEN GOTO 14
0085 GOSUB 187
0086 AT 64:LPRINT "DONE      "
```

```

0087 GOTO 14
0088 CLS:LPRINT "DEL KEY "
0089 AT 64:LPRINT "WAIT KEY  "
0090 z=0
0091 z=z+1:IF z>100 THEN BEEP 15,2:GOTO 123
0092 #(968)=RFID{1}
0093 IF #(968)=1 THEN GOTO 98
0094 #(968)=RFID{2}
0095 IF #(968)=1 THEN GOTO 98
0096 §=GKEY{ }:IF §=0 THEN GOTO 91
0097 GOTO 14
0098 BEEP 15,2:GOSUB 172
0099 IF #(966)<170 THEN GOTO 102
0100 AT 64:LPRINT "NO THIS KEY  ":BEEP 5,8:PAUSE 300
0101 GOTO 14
0102 B=#(966)
0103 #(975)=255:§=B*6
0104 WREE #(975),§
0105 AT 64:LPRINT "DONE  "
0106 GOTO 14
0107 CLS:LPRINT "SET OPEN DOOR"
0108 #(975)=RDEE{1020}
0109 AT 64:LPRINT ".1,%3,"Time= ",#(975)," s  "
0110 §=KEY{15}:BEEP 15,2
0111 CASE §,1,114,2,120,4,122,8,118,16,116
0112 AT 64:LPRINT "NO SET  ":PAUSE 500
0113 GOTO 14
0114 #(975)=#(975)+1:IF #(975)>255 THEN #(975)=255
0115 GOTO 109
0116 #(975)=#(975)-1:IF #(975)<1 THEN #(975)=1
0117 GOTO 109
0118 #(975)=#(975)+10:IF #(975)>255 THEN #(975)=255
0119 GOTO 109
0120 #(975)=#(975)-10:IF #(975)<1 THEN #(975)=1
0121 GOTO 109
0122 WREE #(975),1020:GOTO 14
0123 §-:§-:§-:§-
0124 RETURN
0125 #(975)=RDEE{1022}:IF #(975)&1=0 THEN GOTO 128
0126 #(968)=RFID{1}
0127 IF #(968)=1 THEN #(968)=1:#(992)=10:GOTO 132
0128 #(975)=RDEE{1022}:IF #(975)&2=0 THEN GOTO 131
0129 #(968)=RFID{2}
0130 IF #(968)=1 THEN #(968)=2:#(992)=20:GOTO 132
0131 GOTO 166
0132 CLS:LPRINT "KEY= ",@(969),@(970),@(971),@(972)
0133 LPRINT @(973):AT 64:LPRINT "NO OPEN DOOR  "
0134 #(986)=0:#(987)=#(969): #(988)=#(970)
0135 #(989)=#(971): #(990)=#(972): #(991)=#(973)
0136 GOSUB 172
0137 IF #(966)>169 THEN BEEP 5,16:GOTO 165
0138 #(986)=#(960)
0139 #(967)=#(968)&#(960)&3
0140 IF #(967)=0 THEN BEEP 5,8:GOTO 165
0141 #(967)=#(968)&((#(960)/4)&3)
0142 #(975)=#(968)&((#(960)/16)&3)
0143 IF #(967)+#(975)=0 THEN BEEP 5,8:GOTO 165
0144 #(974)=RDEE{1023}:IF #(974)&1=0 THEN GOTO 148
0145 IF #(967)=0 THEN GOTO 148
0146 PORT 9,1:PORT 5,0:#(992)=#(992)+1
0147 PORT 3,1
0148 #(974)=RDEE{1023}:IF #(974)&2=0 THEN GOTO 152
0149 IF #(975)=0 THEN GOTO 152
0150 PORT 9,1:PORT 5,0:#(992)=#(992)+2
0151 PORT 4,1
0152 #(984)=#(992)
0153 IF #(992)&3=0 THEN GOTO 162
0154 AT 64:LPRINT "OPEN DOOR  "
0155 #(967)=RDEE{1020}
0156 FOR §=0 TO #(967)
0157 IF A%5=0 THEN BEEP 15,2
0158 AT 75:LPRINT %3,A
0159 PAUSE 100
0160 GOSUB 192
0161 NEXT §
0162 PORT 4,0:PORT 3,0
0163 PORT 9,0:PORT 5,1
0164 GOTO 166
0165 #(960)=0:#(984)=#(992)+40:PAUSE 300
0166 RETURN
0167 §=0
0168 FOR §=0 TO 169:REM ' READ DATABASE FIRST BUTE KEY
0169 #(975)=RDEE{A*6}:IF #(975)<>255 THEN B=B+1
0170 NEXT §
0171 RETURN
0172 FOR §=0 TO 169
0173 §=A*6
0174 RDPE *6,§(960),§
0175 §=SCOMP{ #(961),#(969),5}
0176 IF #(960)=255 THEN A=0
0177 IF §=1 THEN #(966)=§:EXFOR 180
0178 NEXT §

```



```

0179 #(966)=255
0180 RETURN
0181 FOR §=0 TO 169:REM ' READ DATABASE FIRST bute KEY
0182 #(975)=RDEE{A*6}:IF #(975)=255 THEN EXFOR 185
0183 NEXT §
0184 #(975)=255
0185 #(975)=§
0186 RETURN
0187 #(975)=255
0188 FOR §=0 TO 1019
0189 WREE #(975),A
0190 NEXT §
0191 RETURN
0192 IF #(976)=0 THEN GOTO 328
0193 CASE #(976),1,197,2,218,3,233,4,237,5,242
0194 CASE #(976),6,270,7,284,8,291,9,296,10,301
0195 CASE #(976),11,314,12,319,13,323,14,257
0196 GOTO 326
0197 FOR §=0 TO 169
0198 §=A*6
0199 RDPE *6,§(960),§
0200 §=SCOMP{#(979),#(961),5}
0201 IF #(960)=255 THEN A=0
0202 IF §=1 THEN #(966)=§:EXFOR 209
0203 NEXT §
0204 GOSUB 181
0205 §=#(975):IF §=255 THEN GOTO 213
0206 §=#(975)*6:WRPE *6,§(978),§
0207 #(976)=0:#(977)=2:AT 64:LPRINT "ADD KEY OK  "
0208 GOTO 328
0209 IF #(978)=#(960) THEN GOTO 215
0210 §=#(966)*6:WRPE *6,§(978),§
0211 #(976)=0:#(977)=3:AT 64:LPRINT "UPDATE KEY OK  "
0212 GOTO 328
0213 #(976)=0:#(977)=34:#(978)=0:AT 64
0214 LPRINT "NO FREE MEM  ":GOTO 328
0215 #(976)=0:#(977)=36:#(978)=0:AT 64
0216 LPRINT "THIS KEY PRESENT"
0217 GOTO 328
0218 FOR §=0 TO 169
0219 §=A*6
0220 RDPE *6,§(960),§
0221 §=SCOMP{#(979),#(961),5}
0222 IF #(960)=255 THEN A=0
0223 IF §=1 THEN #(966)=§:EXFOR 228
0224 NEXT §
0225 #(976)=0:#(977)=35:#(978)=0:AT 64
0226 LPRINT "NO THIS KEY  "
0227 GOTO 328
0228 #(975)=255:§=#(966)*6
0229 WREE #(975),§
0230 #(976)=0:#(977)=4:#(978)=0:AT 64
0231 LPRINT "DEL KEY OK  "
0232 GOTO 328
0233 GOSUB 187
0234 #(976)=0:#(977)=5:#(978)=0:AT 64
0235 LPRINT "DEL ALL KEY OK  "
0236 GOTO 328
0237 IF #(978)=0 THEN #(978)=RDEE{1020}:GOTO 239
0238 WREE #(978),1020
0239 #(976)=0:#(977)=6:AT 64
0240 LPRINT "SET TIME OK  "
0241 GOTO 328
0242 GOSUB 167
0243 #(978)=§
0244 #(979)=RDEE{1020}
0245 #(980)=RDEE{1021}
0246 #(980)=(!#(980))&1
0247 #(981)=RDEE{1022}
0248 #(981)=(!#(981))&3
0249 #(982)=RDEE{1023}
0250 #(982)=(!#(982))&3
0251 #(983)=0
0252 IF PORT*{3}=1 THEN #(983)=#(983)+1
0253 IF PORT*{4}=1 THEN #(983)=#(983)+2
0254 #(976)=0:#(977)=1:AT 64
0255 LPRINT "SEND STATUS OK  "
0256 GOTO 328
0257 IF #(978)=0 THEN GOTO 225
0258 §=0
0259 FOR §=0 TO 169:REM ' READ DATABASE FIRST bute KEY
0260 #(967)=A:REM ' сохраним адрес в базе
0261 #(975)=RDEE{A*6}:IF #(975)=255 THEN GOTO 263
0262 Б=Б+1:IF Б=#(978) THEN §--:EXFOR 265
0263 NEXT §
0264 GOTO 225
0265 §=#(967)*6:#(975)=255
0266 WREE #(975),§
0267 #(976)=0:#(977)=1:AT 64
0268 LPRINT "DEL KEY OK  "
0269 GOTO 328
0270 IF #(978)=0 THEN GOTO 225

```

```

0271 §=0
0272 FOR §=0 TO 169:REM ' READ DATABASE FIRST bute KEY
0273 #(967)=A:REM ' сохраним адрес в базе
0274 #(975)=RDEE{A*6}:IF #(975)=255 THEN GOTO 276
0275 Б=Б+1:IF Б=#(978) THEN §-:EXFOR 278
0276 NEXT §
0277 GOTO 225
0278 §=#(967)*6:RDPE *6,$(960),§
0279 #(978)=#(960):#(979)=#(961):#(980)=#(962)
0280 #(981)=#(963):#(982)=#(964):#(983)=#(965)
0281 #(976)=0:#(977)=1:AT 64
0282 LPRINT "SEND KEY OK  "
0283 GOTO 328
0284 #(975)=255
0285 IF #(978)=1 THEN #(975)=0
0286 IF #(978)=0 THEN #(975)=255
0287 WREE #(975),1021
0288 #(976)=0:#(977)=1:AT 64
0289 LPRINT "LOCK KBD ON/OFF  "
0290 GOTO 328
0291 #(978)=(1#(978))&3
0292 WREE #(978),1022
0293 #(976)=0:#(977)=1:AT 64
0294 LPRINT "LOCK READ ON/OFF"
0295 GOTO 328
0296 #(978)=(1#(978))&3
0297 WREE #(978),1023
0298 #(976)=0:#(977)=1:AT 64
0299 LPRINT "LOCK DOOR ON/OFF"
0300 GOTO 328
0301 #(976)=0:#(977)=1
0302 IF #(978)=0 THEN GOTO 328
0303 PORT 9,1
0304 IF #(978)&1=1 THEN PORT 3,1
0305 IF #(978)&2=2 THEN PORT 4,1
0306 AT 64:LPRINT "OPEN DOOR CMD  "
0307 #(967)=RDEE{1020}
0308 FOR §=0 TO #(967)
0309 IF A%5=0 THEN BEEP 15,2
0310 PAUSE 100
0311 NEXT §
0312 PORT 4,0:PORT 3,0:PORT 9,0
0313 GOTO 328
0314 #(976)=0:#(977)=1
0315 PORT 3,1
0316 PORT 4,1
0317 AT 64:LPRINT "OPEN ALL DOOR  "
0318 GOTO 328
0319 #(976)=0:#(977)=1
0320 PORT 4,0:PORT 3,0
0321 AT 64:LPRINT "CLOSE DOOR CMD  "
0322 GOTO 328
0323 AT 64:LPRINT "REBOOT      "
0324 #(976)=0:#(977)=1:PAUSE 500
0325 §^:CLR:GOTO 329:REM "***** REBOOT
0326 #(976)=0:#(977)=33:#(978)=0
0327 AT 64:LPRINT "ERROR CMD  "
0328 RETURN
0329 LINIT 1:BEEP 15,2
0330 LPRINT "SKD V2.0 Start "
0331 PORT 5,1:PORT 9,0
0332 GOSUB 167
0333 AT 64:LPRINT "KEY in base =",§:PAUSE 1500:BEEP 15,2
0334 CLS:LPRINT "READY":GOTO 335
0335 GOSUB 7
0336 GOSUB 341
0337 GOSUB 125
0338 GOSUB 192
0339 GOSUB 343
0340 GOTO 335
0341 CLS:LPRINT "READY"
0342 RETURN
0343 IF PORT{10}=1 THEN GOTO 356
0344 PORT 3,1
0345 PORT 4,1
0346 PORT 9,1
0347 #(986)=0:#(987)=0:#(988)=0
0348 #(989)=0:#(990)=0:#(991)=0
0349 #(960)=0:#(984)=83
0350 #(967)=RDEE{1020}
0351 FOR §=0 TO #(967)
0352 IF A%5=0 THEN BEEP 15,2
0353 PAUSE 100
0354 NEXT §
0355 PORT 4,0:PORT 3,0:PORT 9,0
0356 RETURN

```