

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт энергетики и электротехники  
(институт)

Кафедра «Промышленная электроника»

11.03.04 Электроника и нанoeлектроника  
(код и наименование направления подготовки, специальности)

Промышленная электроника  
направленность (профиль)

## БАКАЛАВРСКАЯ РАБОТА

на тему Кодовый замок с микроконтроллерным управлением

Студент(ка)	<u>И.К. Ермолаева</u> (И.О. Фамилия)	_____	(личная подпись)
Руководитель	<u>М.В. Позднов</u> (И.О. Фамилия)	_____	(личная подпись)
Консультанты	_____	_____	(личная подпись)
	_____	_____	(личная подпись)

**Допустить к защите**

Заведующий кафедрой к.т.н., доцент А.А. Шевцов  
(ученая степень, звание, И.О. Фамилия) \_\_\_\_\_  
(личная подпись)

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г.

Тольятти 2016

## **Аннотация**

Объем 83 с., 22 рис., 1 табл., 21 источников, 1 прил.

### **КОДОВЫЙ ЗАМОК, МИКРОКОНТРОЛЛЕР**

Объектом исследования является кодовый замок с микроконтроллерным управлением.

Цель работы: кодового дверного замка с микроконтроллерным управлением

Задачи работы:

1. обзор существующих решений
2. Выбор механический и электромеханической части замка
3. Разработка электрической части замка
4. Написание программы

Работа состоит из четырех глав, в которых решены упомянутые задачи.

Для написания программы использовалась программа Python.

Степень внедрения: произведена разработка и расчет устройства.

Данная работа может быть применена при разработке и изготовлении кодовых дверных замков.

## Содержание

Аннотация .....	2
Введение.....	4
1 Состояние вопроса .....	5
1.1 Анализ исходных данных и известных решений .....	6
1.1.1 Механические замки .....	6
1.1.2 Электромеханические замки .....	9
1.1.3 Электромагнитные замки .....	10
1.1.4 Резонансные электронные замки и ключи.....	12
1.1.5 Кодовый замок-звонок.....	13
1.1.6. Замок с магнитным ключом. ....	14
1.2 Формулировка актуальности, цели и задачи проекта.....	16
2 Механическая и электромеханическая части замка .....	22
3 Электрическая часть замка.....	25
3.1 Формирователь отпирающего импульса.....	25
3.2 Система идентификации.....	25
3.3 Вспомогательные схемы.....	27
3.3.1 Микроконтроллер.....	27
3.3.2 Блок питания .....	27
3.3.3 Кварцевый генератор .....	30
3.3.4 Блок световой и звуковой индикации .....	30
4 Программная часть.....	33
4.1 Общая блок-схема программы.....	33
4.1.1 Описание блок-схемы программы.....	33
4.1.2 Инструкция по использованию замка: .....	33
4.2 Подпрограмма идентификации кода .....	36
4.3 Подпрограмма отпираания замка .....	37
4.4 Подпрограмма задания кода.....	38
4.5 Вспомогательные подпрограммы .....	42
4.5.1 Подпрограмма таймера.....	42
4.5.3 Подпрограмма записи часов.....	44
4.5.4 Подпрограмма нажатия кнопок .....	45
4.5.5 Подпрограмма индикации .....	46
4.5.6 Подпрограмма основная .....	48
Заключение .....	49
Список используемой литературы .....	50
Приложение А .....	52

## Введение

Дверные замки играют важную роль в вопросах сохранности имущества и обеспечения внешней и внутренней безопасности.

Эволюция замков вобрала в себя массу достижений в части конструирования и дизайна. Существуют механические (сувальдные, цилиндровые), электрические (кодовые, с электронными ключами, картами) и различными приводами электромагнитными, электромеханическими (врезные, накладные).

Одной из систем, предназначенной для защиты от проникновения в помещение посторонних лиц, является система с замком электромеханического типа. Система основана на следующем принципе действия: каждому человеку (группе лиц) выдается определенный код, который будет относиться к одной из трех уровней доступа, с помощью кода человек может войти в дверь (двери, работающие по этой системе). По коду первого уровня можно зайти только в помещения, определенные под низший уровень и строго с 8.00 по 20.00, с сентября по июнь. По коду второго уровня можно зайти в помещения, определенные под низший и средний уровни строго с 8.00 по 20.00. По коду третьего уровня можно зайти в помещения, определенные под низший, средний и высший уровни, т.е. во все помещения, без ограничения. Человек набирает код на панели с функцией защиты от подсматривания. Можно добавлять и удалять коды. Для устранения естественных погрешностей по времени предусмотрено редактировать часов.

Эта система может пригодиться там, где необходимо обеспечить безопасность помещения. Это могут быть жилые, рабочие помещения. Применение этой системы приведет к дополнительным удобствам, например, существуют несколько лабораторий, офисных помещений и ряд сотрудников, которым необходимо попадать только в определенные помещения. Для этого каждому сотруднику выдается персональный код, который дает ему возможность открыть дверь только необходимых помещений, а в остальных ему будет отказано в доступе. Нет необходимости носить при себе много ключей, которые

можно перепутать, потерять, сломать. Проанализировав основной недостаток кодовых замков – можно подглядеть код, предполагается установить специальную панель, не позволяющую постороннему человеку увидеть набираемый код.

Создание этой системы на базе микропроцессора позволит получить безопасное, удобное, простое в обращении и несложное в изготовлении.

Данная работа посвящена разработке системы с замком электромеханического типа, предназначенной для защиты от проникновения в помещение посторонних лиц.

# 1 Состояние вопроса

## 1.1 Анализ исходных данных и известных решений

### 1.1.1 Механические замки

Самые распространенные и привычные в настоящее время замки – механические. Эволюция механических замков вобрала в себя массу достижений в части конструирования и дизайна.

Замки по способу установки делятся на накладные и врезные. Накладной замок устанавливается снаружи на полотне двери. Врезные замки устанавливаются в теле полотна двери. Накладные замки меньше ослабляют полотно двери, чем врезные и требуют меньше времени на установку.

По принципу внутреннего устройства привода механические замки подразделяются на две категории:

Первая категория: замки с непосредственным приводом запирающего стержня (сувальдные);

Вторая категория: замки с косвенным приводом запирающего стержня (цилиндровые).

В замках с непосредственным приводом (рисунок 1.1) (пример - большинство сейфовых) ключ при его вращении непосредственно воздействует на запирающую пластину. Бороздки ключа, строго определенной конфигурации, попадают в кодовые впадины механизма и, при несовпадении, препятствуют проворачиванию ключа. Такие замки относительно легко поддаются отпиранию отмычками.

В цилиндрических замках (рисунок 1.2) запирающая пластина беспрепятственно перемещается при вращении стержня-привода. Но вращение самого цилиндра (в который вставляется ключ) и жестко соединенного с ним стержня-привода, запрещено (при отсутствии ключа) рядом маленьких подпружиненных со стороны обоймы цилиндров (одинакового диаметра, но разной длины) которые проталкиваются миниатюрными пружинками из отверстий обоймы в по-

перечные отверстия цилиндра привода, тем самым стопоря вращательное движение цилиндра-привода в обойме. Когда ключ вставляется в торцовое отверстие цилиндра-привода

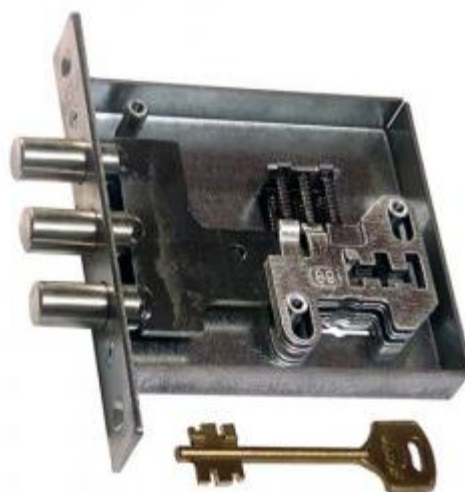


Рисунок 1.1 – Внутреннее устройство сувальдного замка.



Рисунок 1.2 – Внутреннее устройство цилиндрического замка.

(личину), то его выпуклые кулачки оказываются точно напротив каждого из стопорящих цилиндров, причем кулачок ключа утапливает "свой" стопорный цилиндр ровно настолько, чтоб тот не мешал поперечному движению (вращению) цилиндра-привода. Если выпуклость бороздки ключа будет немного меньше необходимого, то цилиндрок останется высунутым из обоймы и запретит проворачивание цилиндра-привода. Выпуклость кулачка ключа больше необходимого невозможна по причине того что такой ключ не войдет в отверстие (личину) замка.

Разновидность цилиндрических замков - замки с цилиндрами "пинового типа", в которых взаимодействующими частями являются разрезные наклонные штырьки (пины) расположенные в теле цилиндра в различных плоскостях и со-

ответствующие им проточки на ключе. Секретность замка определяется количеством используемых пинов и их взаимным расположением. Количество комбинаций ключей таких замков, доходит до нескольких миллиардов, что делает практически невозможным открытие замка с помощью отмычек. Отрицательными качествами цилиндрических замков является их зачастую слабое противодействие грубому механическому проворачиванию цилиндра-привода при котором стопорные цилиндрики срезаются и уже не препятствуют поворачиванию.

При запираении замок выдвигает один или несколько запирающих стержней, фиксирующих положение двери. Самой высокой устойчивостью против взлома обладают многоригельные замки. При запираении двери таким замком механизм замка выдвигает запирающие ригели (стальные стержни большой длины) в двух или четырех (горизонтальном и вертикальном) направлениях. Такое запираение двери, при достаточной ее прочности, обеспечивает высокую устойчивость против взлома.

**Механические кодовые** замки обладают одним неоспоримым преимуществом - невозможностью потери ключа. Однако, их использование ограничено по причинам достаточно легкого определения цифр набираемого кода. В простых механических кодовых замках последовательность набора цифр не имеет значения. А именно, набор, например, числа 23 совершенно равнозначен набору числа 32. Учитывая то, что код может состоять только из разных цифр, количество комбинаций набора предельно сокращается. Так, например, злоумышленник может нанести метки на кнопки наборного поля. При наборе кода на тех клавишах наборного поля, которые нажимались, метки стираются. Кроме того, код можно подсмотреть. Поэтому, при использовании кодовых замков, один из способов повышения степени защиты механического кодового замка является периодическая смена кода.

Механические кодовые замки могут использоваться совместно с другими устройствами для условного доступа в помещение.



Существуют две разновидности **электрических** замков: замки с электромеханическим приводом и электромагнитные замки.

### ***1.1.2 Электромеханические замки***

Электромеханические замки внешне представляют собой обычные механические замки с возможностью их открытия электрическим импульсом тока от источника напряжением 12 или 24 вольт, что происходит при нажатии удаленной кнопки.

Такой замок можно открыть тремя способами:

- электрическим импульсом подаваемым от удаленной кнопки;
- расположенной с внутренней стороны корпуса замка механической или электрической кнопкой;
- обычным механическим ключом.

Бывают электромеханические замки навесные (рисунок 1.3) и врезные (рисунки 1.4, 1.5).



Рисунок 1.3 – Навесной электромеханический замок.



Рисунок 1.4 – Врезной электромеханический замок.



Рисунок 1.5 – Врезной замок, установленный в дверь.

Электромеханические замки имеют открывающий механизм, управляемый небольшой электромагнитной защелкой. При закрывании двери запирающий ригель защелкивается и остается в высунутом и подпружиненном с небольшим усилием состоянии. В это же время автоматически взводится и фиксируется защелкой сильная пружина открывающего механизма. При открывании замка электромагнит освобождает защелку и пружина освободившись и преодолев сопротивление пружины запирающего ригеля, вытаскивает его. Такая конструкция замка обладает двумя недостатками:

- короткой длиной запирающего стержня запора;
- невозможностью повторного дистанционного закрывания замка.

Это значит, что после подачи открывающего сигнала дверь с замком такой конструкции останется в незапертом состоянии до тех пор, пока ее не откроют и не закроют снова.

### ***1.1.3 Электромагнитные замки***

Электромагнитные замки недороги и просты в установке. Такие замки весьма удобны на путях эвакуации при пожаре, теракте или стихийном бедствии. Их удерживающая сила не превышает 300-500 кг.

Электромагнитный замок (рисунок 1.6) представляет собой мощный электромагнит с плоской стальной пластиной. Для закрытия электромагнитного замка на его электромагнит подается постоянное напряжение. При закрывании двери, стальная пластина, закрепленная на ней, приближаясь к электромагниту, притягивается им и удерживается его магнитным полем.



Рисунок 1.6 - Электромагнитный замок.

В лучших конструкциях электромагнитных замков закрытое состояние двери поддерживается доводчиком двери, а на электромагнит подаются лишь зондирующие импульсы напряжения, чем резко снижается энергопотребление замка. При попытке оторвать дверь, на электромагнит контроллером замка подается полное удерживающее напряжение.

Открывание электромагнитного замка происходит при отключении питания. В комплекте с таким замком необходимо применять дверной доводчик.

Практически всех отрицательных качеств механических кодовых замков лишены большинство конструкций **электронных кодовых замков**.

В электронных кодовых замках много различных видов.

**Системы контроля доступа.** Под столь громким названием скрывается электронный кодовый замок, в котором вместо наборного поля использована пара: электронный ключ - считыватель. Электронный ключ (рисунок 1.7) представляет собой цилиндрическую головку, толщиной 6 мм и диаметром 20 мм с пластмассовым "хвостиком". В головке "зашифрован" магнитный код длиной несколько десятков байтов (байт - восемь двоичных разрядов). Одним из недостатков является возможность потерять ключ и следовательно невозможность открытия замка снаружи. Также их пластмассовый "хвостик" легко ломается и цилиндрическая головка выскакивает из него, что создает определенные неудобства по этому приходится менять ключи.



Рисунок 1.7 – Электронный ключ.

### 1.1.4 Резонансные электронные замки и ключи

Принцип действия этих электронных замков и ключей основан на явлении электронного резонанса. Известно, что существует два вида резонансных колебательных LC-контуров: последовательный и параллельный. Схема таких контуров и графики, иллюстрирующие зависимость общего сопротивления контура от частоты подводимого сигнала, показаны на рисунках 1.8 и 1.9.

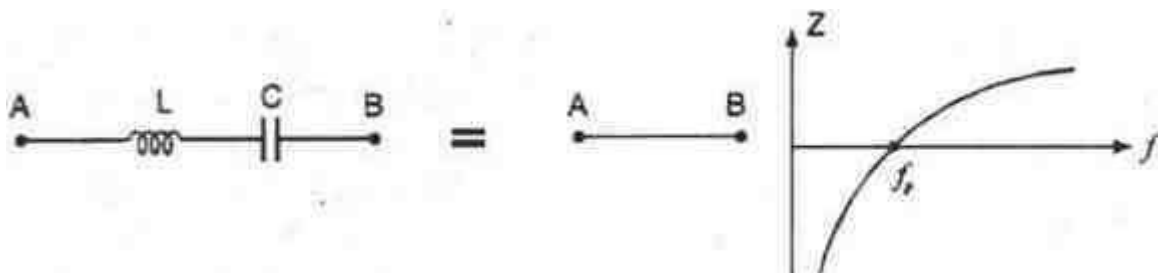


Рисунок 1.8 - Последовательный колебательный контур.

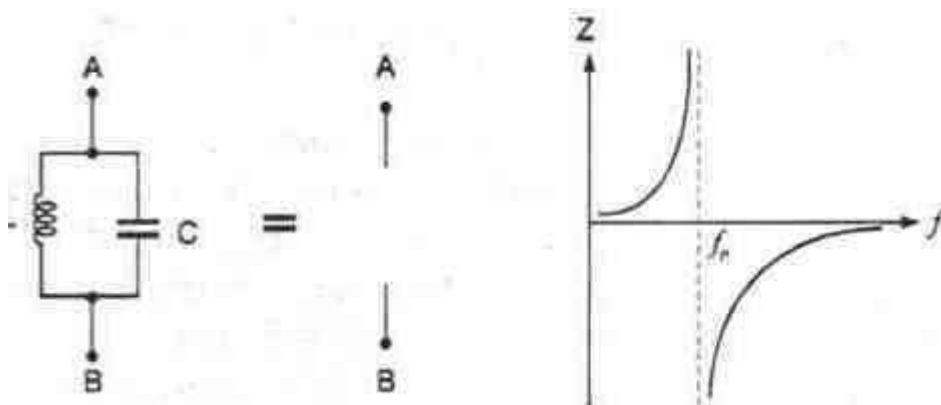


Рисунок 1.9 - Параллельный колебательный контур.

Из графиков, изображенных на этих рисунках, видно, что сопротивление  $Z$  последовательного LC-контура (рисунок 1.8) на частоте резонанса  $f_r$  стремится к нулю, а сопротивление параллельного (рисунок 1.9) LC-контура - к бесконечности. Иначе говоря, в момент резонанса последовательный контур подобен отрезку провода (точки A и B замкнуты), а параллельный - обрыву между точками A и B.

Резонансная частота контура может быть вычислена по известному соотношению (1.1):

$$f_p = 1/(2\pi\sqrt{LC}),, \quad (1.1)$$

где  $f_p$  - резонансная частота контура, Гц;  $L$  - индуктивность катушки, Гн;  $C$  - емкость конденсатора, Ф.

Минимальное или максимальное сопротивление контура проявляется тем ярче, чем меньше потери в нем.

Для замков, использующих резонансный метод, - LC-контур - ключом может быть катушка индуктивности  $L$  или конденсатор  $C$ , входящие в этот контур. Если функцию ключа выполняет конденсатор, то контурная катушка должна находиться внутри замка, а на внешнюю сторону двери должны быть выведены два контакта, для подключения конденсатора. При подключении конденсатора к катушке в образовавшемся контуре возникает резонанс, изменяющий его сопротивление.

Приборы с использованием резонансного метода хорошо работают только на тех частотах, на которых резонансные свойства контуров выражены наиболее ярко. Для резонансного замка наиболее приемлемым является диапазон частот 50-500 кГц. Параметры ключа зависят от температуры, по этому для гарантийного срабатывания ключа приходится расширять диапазон частот, принимаемый за верный и тем самым уменьшается число ключей. А также существует возможность изготовления отмычки, изменяющей свою резонансную частоту, т.е. взлом замка. Что является существенным недостатком.

### ***1.1.5 Кодовый замок-звонок***

У описываемого варианта кодового замка отсутствует традиционное наборное кнопочное поле. Код набирают одной кнопкой - последовательным нажатием и отпусканием. Она одновременно выполняет функцию кнопки дверного звонка, причем на время набора кода звонок отключается. При этом электронный блок замка формирует состоящее из нулей и единиц восьмиразрядное двоичное число. Введению единицы соответствует замыкание контактов кнопки в течение 1-2 с, а нуля - менее 1 с.

При первом нажатии на кнопку в квартире раздается звуковой сигнал, а после ее отпускания устройство блокирует подачу сигнала на несколько секунд - а это время необходимо приступить к набору кода. Если период между нажатием не превышает трех секунд, то набор кода происходит без включения сигнала звонка, т. к. каждое очередное размыкание контактов кнопки продлевает блокировку звукового сигнала на три секунды.

После набора кода необходимо еще раз нажать кнопку и удерживать ее в этом состоянии. Если код набран верно, через две секунды сработает электромагнит замка и дверь откроется.

Главным недостатком является сложность в наборе кода. Набор требует предельной концентрации внимания, т.к. при наборе кода кнопку нельзя удерживать нажатой более двух секунд, иначе замок окажется заблокированным и не будет реагировать на дальнейшие нажатия.

#### ***1.1.6. Замок с магнитным ключом.***

Работа этого замка основана на использовании постоянных магнитов и герконов (герметичных контактов). Это позволяет получить надежные с точки зрения устойчивости работы замки и ключи. Принципиальная схема одного из таких замков представлена на рисунке 1.10.

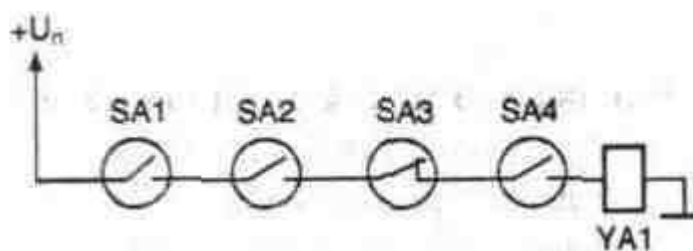


Рисунок 1.10 - Замок с магнитным ключом.

Замок состоит из четырех герконов SA1, SA2, SA3, SA4 и электромагнита YA1. Причем, герконы SA1, SA2, SA4 имеют нормально разомкнутые контакты, а геркон SA3 - нормально замкнутые. Герконы могут устанавливаться в произвольной последовательности, количество их может быть различным, что способствует повышению секретности замка.

Замок и ключ имеют следующую конструкцию (рисунок 1.11). Герконы SA1- SA4 устанавливаются вдоль отверстия, в которое вставляется ключ с постоянными магнитами. Расстояние между магнитами и герконами должно быть не более 5 мм. Расстояние между герконами должно быть равно расстоянию между

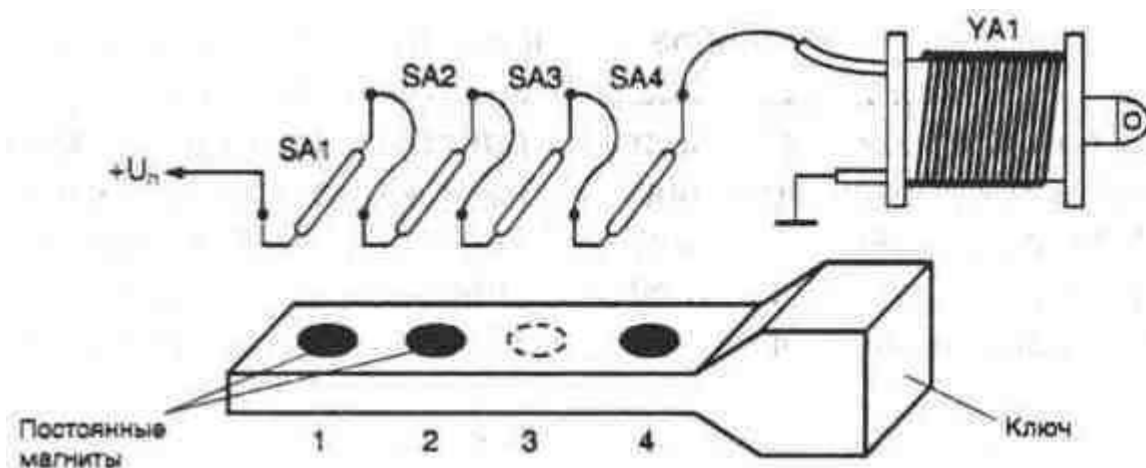


Рисунок 1.11 - Конструкция замка и ключа.

центрами магнитов ключа. После того как ключ будет вставлен в отверстие замка до упора, магниты 1, 2, 4 будут находиться напротив герконов SA1, SA2, SA4 соответственно, что приведет к замыканию контактов последних. Поскольку контакты геркона SA3 замкнуты (магнит на него не воздействует), через обмотку электромагнита YA1 начинает протекать ток, вызывая его срабатывание. Главным недостатком является сложность изготовления ключа.

Проанализировав исходные данные и известные технические решения, пришли к выводу, что наиболее лучшим является кодовый электромеханический замок, который имеет возможность питаться от сети напряжением 220 В, а также от аккумуляторных батарей напряжением 1,5 В. Запирающим замком будет узкий электромеханический замок для профильных дверей Abloy EL412 с управляемой внешней ручкой. Идентификацию будет осуществлять кодовая панель ЦИФРАЛ CD-96М.

В данной работе предполагается разработать систему с замком электромеханического типа. Одной из особенностей предполагается специальная мера по защите от подсматривания кода, заключающаяся в специальной конструк-

ции кодовой панели, ее нет в продаже. Панель необходимо изготовить, но наша система прекрасно будет функционировать и с простой кодовой панелью ЦИФРАЛ CD-96М. Задуманная панель (рисунок 1.12) осуществит идею невозможности подглядеть код, которая основана на том, что сами кнопки спрятаны за пластмассовую панельку, на которой изображены подписи кнопок (цифры). И посторонний человек не может визуальнo контролировать, какие кнопки нажимаются на оборотной стороне владельцем кода.

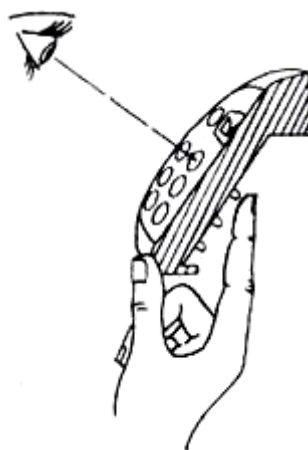


Рисунок 1.12 - Схематичная конструкция кодовой панели.

## 1.2 Формулировка актуальности, цели и задачи проекта

Замки и ключи были известны задолго до рождения Христа. Они часто упоминаются в Ветхом Завете и в мифах. В книге Нехемии, глава 3, указывается, что при ремонте старых врат Иерусалима - приблизительно в 445 до н. э. - "поставили их двери, и их замки, и их засовы". В то время замки изготавливали из дерева. Они были большими, их конструкция была примитивна, и все же принцип их работы был предтечей современных штифтовых замков.

Когда замочных и железных дел мастера достигали большого искусства в своем ремесле, их приглашали изготавливать замки и ключи для королевских дворов и для европейских церквей и соборов. Их замки славились изысканной, сложной и подробной орнаментацией - зачастую на религиозные сюжеты.

В Индии, в эпоху императора Аннама, сокровища запечатывали в боль-



шие деревянные чурбаны, которые помещали на небольшие островки или погружали в пруды, находившиеся на территории внутренних дворов дворца. Здесь они были защищены королевскими "ангелами-хранителями" - крокодилами, которых кормили впроголодь. Зайти в воду означало верную гибель. Со кровища можно было получить только единственным путем - усыпить или убить крокодилов.

На протяжении многих сотен лет для "запираания" дверей и укрепления стен использовали веревки, сделанные из камыша и волокна. Согласно легенде, завязанная узлом веревка стала символом безопасности. Искусно завязанным фригийским королем Гордием узлом, известным как гордиев узел, крепилось ярмо к оглобле его колесницы. Оракулы утверждали, что развязать его может только человек, которому суждено покорить Азию. Но, когда Александру Македонскому не удалось развязать гордиев узел, он перерубил его своим мечом. Так возникло выражение "разрубить гордиев узел", что значит предпринять смелое, решительное действие, когда достичь цели более мягкими мерами не удастся.

Медные и железные висячие замки, которые обнаруживают в Европе и на Дальнем Востоке, широко применялись римлянами и китайцами. Им оказывали особое предпочтение потому, что они были портативными. Они открывались ключами (рисунок 1.13), которые можно было поворачивать, ввинчивать и нажимать. Конструкция замка с нажимным ключом (рисунок 1.13) была проста: засов удерживался в запирающей позиции выступающей пружиной или пружинами. Чтобы открыть замок, нужно было прижать или распрямить пружины ключом, благодаря чему засов освобождался и отходил в сторону. Висячие замки этого типа наиболее часто используются на востоке в настоящее время. Узор на замке соответствовал традиции каждой страны, а фигурки часто изготавливались в форме животных - драконов, лошадей, собак, даже слонов и бегемотов. Пара висячих замков с нанесенными клинописью поздравительными надписями часто служила хорошим подарком.



Рисунок 1.13 Нажимной ключ.

Первые механические замки из дерева были, вероятно, созданы несколькими цивилизациями одновременно. Письменные источники свидетельствуют, что их применяли около 4000 лет назад в Египте. В этих деревянных замках, которые крепили вертикально к дверному косяку, находились подвижные штыри, или "пальцы", которые благодаря собственному весу опускались в отверстия в крестовине, или "засове", и запирали дверь. Замок открывался деревянным ключом с выступами или зубцами, которые поднимали ряд пальцев, освобождающих засов, так что его можно было отодвинуть. Этот метод запираения был предтечей современных штифтовых замков.

Первые цельнометаллические замки появились между 870 и 900 гг. благодаря английским ремесленникам. Они представляли собой обыкновенные железные засовы с выступами вокруг замочных скважин для предотвращения взлома.

Впервые неподвижные выступы в замке начали применяться римлянами, которые придумали их для того, чтобы чужой ключ не мог повернуться в замке. На выступах делали насечки и нарезали узоры. Система выступов существовала как основной замочный механизм на протяжении более тысячи лет. Первые висячие замки были "удобными" замками, так как их можно было по необходимости переносить и использовать. Они были издавна известны купцам, путешествующим по древним торговым путям в Азию и Европу.

В XVII веке в Европе были разработаны новые принципы запирающих механизмов. В ранних замках "Брама" использовался ряд кругообразно распо-

ложенных подвижных колец, что обеспечивало исключительную безопасность. "Брама" является старейшей компанией в мире по производству замков, которая и сегодня, спустя вот уже 200 лет, продолжает производить замки с их знаменитым механизмом.

Конструкции замков (рисунок 1.14) и ключей испытывали значительное влияние готической архитектуры: их орнаментация, даже во времена Возрождения, становилась все более изысканной. Признанных мастеров приглашали изготавливать замки для знати по всей Европе. Именно поэтому трудно установить, был ли какой-нибудь древний замок изготовлен в той же стране, где и применялся.



Рисунок 1.14 – Готический замок.

Висячие замки использовались столетиями для запираения заключенных и имущества. Они обычно делались из железа, бронзы или меди и имели грубую конструкцию. Однако внутренние запирающие механизмы были зачастую довольно просты и легки для вскрытия.

Массивный русский висячий замок (Рисунок 1.15), был тщательно выкован вручную в раннее царствование последнего царя, Николая II (1895-1918). Большое округлое кольцо сверху - это "рукоятка" ключа, на котором есть нарезка и который ввинчивается в замочную скважину, чтобы разблокировать запирающий механизм. Когда скоба становится в позицию запираения, ключ вынимается и вставляется затычка, чтобы казалось, что замочной скважины нет. Нарезная часть ключа ввинчивается затем в защитный футляр.



Рисунок 1.15 - Русский висячий замок.

Находчивые изобретатели заменяемыми бородками, особые механизмы вокруг замочных скважин для предотвращения взлома, колокольчики, которые прикреплялись к засову и звенели при касании засова, "секретные", или колечные, замки, работающие по тому же принципу, по которому работают современные кодовые замки на банковских сейфах, открывающиеся не ключом, а посредством набора определенной комбинации на циферблате.

Первые секретные замки появились на востоке. На них было от трех до семи колец с символами или буквами, которые, если их правильно выстроить, освобождали запор. Принцип работы наборных замков был схожим. Для того, чтобы открыть эти два типа замков, требовалось набрать определенные слова или цифры, которые знали только владелец или ответственное лицо. "Эврика" (рисунок 1.16), кодовый замок с пятью пальцами, защищенный от случайного набора кода. Этот замок запирает когда-то один из сейфов казначейства США. Его запатентовали в 1862 г. Доддз, Мэкил и Урбан из города Кантон, штат Огайо. Количество букв и цифр на замке делает возможным набор 1 073 741 824 комбинаций; чтобы все их перебрать, не прерываясь, потребовалось бы 2 042 года, 324 дня и 1 час.



Рисунок 1.16 – Замок "Эврика".

Проанализировав историю развития замков пришли к выводу, что наиболее лучшим для целей создания систем ограничения доступа является замок электромеханического типа.

Таким образом, целью данной работы является разработка кодового дверного замка с микроконтроллерным управлением

В ходе проектирования необходимо решить ряд задач: подобрать механическую и электромеханическую части замка. Разработать электрическую часть замка, а в частности формирователь отпирающего импульса, системы идентификации, вспомогательной схемы. А также необходимо составить общую блок-схему программы и написать ряд подпрограмм.

## **2 Механическая и электромеханическая части замка**

Проанализировав существующие варианты, выбрали ABLOY EL412. Замок используется, например, на внешних дверях частных домов и зданий, предприятий. Замок имеет двойной ригель повышенной надёжности для быстрой и легкой работы и предлагают экономичный способ электрического запирания, который намного выгоднее по сравнению с электро-защёлкой.

### **Свойства:**

- Переставляемый язычок позволяет устанавливать сторонность замка
- Быстрая расфиксация ригеля
- Возможность монтажа в раме. Можно ставить горизонтально в дверном полотне или раме.
- Широкий выбор различных бэксэтов (расстояние от передней планки до середины цилиндра): подходит для большинства профилей.

### **Работа:**

ABLOY EL412 – нормально закрытый тип замка для профильных дверей. Замки управляются электрически от разнообразных систем контроля доступа или устройств удалённого доступа таких как нажимные кнопки, наборные панели или таймеры и можно подключать к распашной автоматике дверей.

Ригель замка всегда автоматически фиксируется, если дверь закрыта.

Дверь можно открыть толкнув от себя или потянув на себя, когда электрическое управление включено. Когда электро-управление снято, замком можно управлять от ключа или поворотной кнопки.

### **Особенности работы:**

EL412 не рекомендуется ставить на двери с уплотнителем. В маятниковых дверях замок может быть закрыт только, когда дверь находится в закрытом положении. Замки нельзя устанавливать на противопожарные двери.

### **Схема подключения:**

Схема подключения изображена на рисунке 2.1.

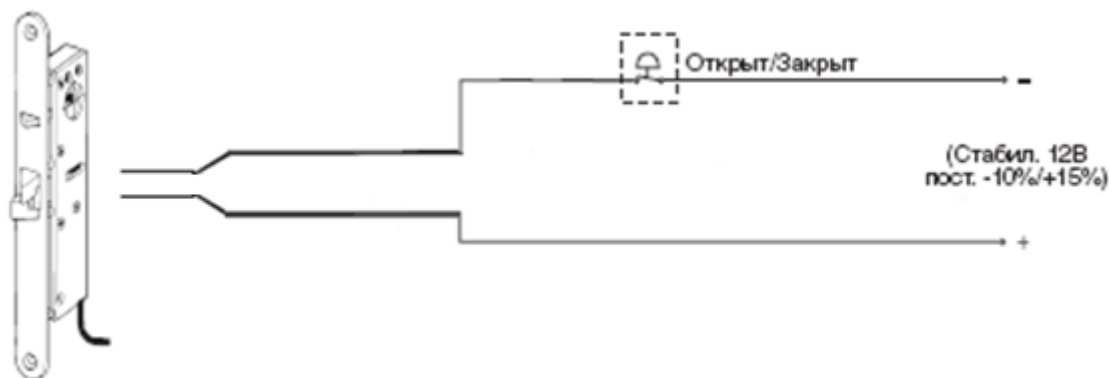


Рисунок 2.1 – Схема подключения замка ABLOY EL412.

Технические данные замка EL412 приведены в таблице 2.1.

Таблица 2.1 - Технические данные замка EL412

Технические параметры	Номинальные значения
Рабочее постоянное напряжение, В	12-10%/+15%
Обычный ток, А	0,48
Рабочая температура, °С	-20...+60
Выход ригеля, мм	14
Расстояние от перед планки до середины цилиндра, мм	25, 28, 30, 35
Передняя планка, мм	25

**Обработка:**

Хромированная передняя планка, хромированный корпус замка.

**Цилиндры:**

Скандинавский овальный тип; ABLOY® PRO, ABLOY® PROTEC, ASSA, RUKO, TrioVing и др.

**Размеры:**

Конструкция замка ABLOY EL412 изображены на рисунке 2.2.

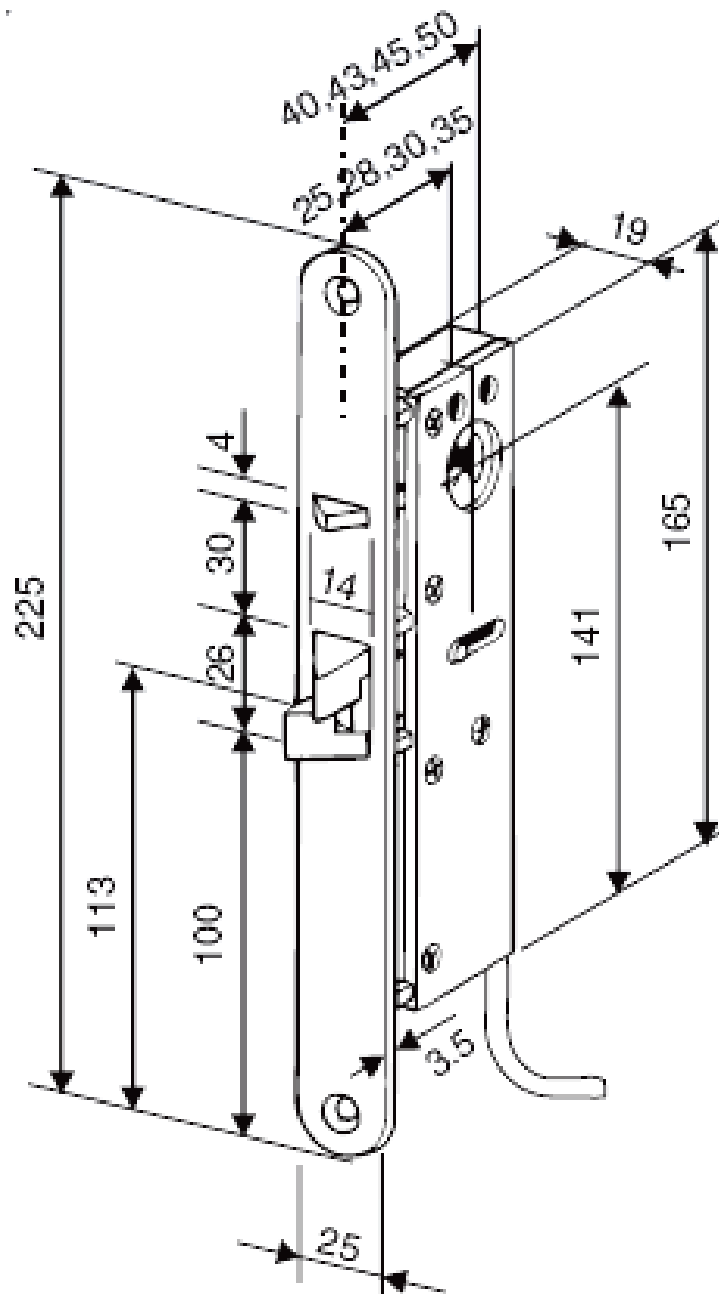


Рисунок 2.2 – Конструкция замка ABLOY EL412.



## 3 Электрическая часть замка

### 3.1 Формирователь отпирающего импульса

Принципиальная схема замка изображена на рисунке 3.1.

При подаче на бит порта P3.0 напряжения высокого уровня («лог 1»). Транзистор  $VT_1$  открывается и переходит в режим насыщения. Напряжение +14 В прикладывается к замку. Транзистор  $VT_1$  выбираем с большим коэффициентом усиления по току для возможности ограничения тока базы  $VT_1$ . Выбрали транзистор КТ972А, с коэффициентом усиления по току  $\beta_{VT_1}=750$ . Необходимо для открытия замка подать на него  $I_{кVT_1}=0,48$  А. Рассчитаем ток базы транзистора  $VT_1$ :

$$I_{бVT_1} = I_{кVT_1} / \beta = 0,48 / 750 = 0,64 \cdot 10^{-3} \text{ А.} \quad (3.1)$$

Рассчитаем резистор  $R_3$ :

$$R_3 = U_{пит} / I_{бVT_1} = 5 / 0,64 \cdot 10^{-3} = 8,2 \text{ кОм,} \quad (3.2)$$

где  $U_{пит}$  – напряжение питания.

Возьмем резистор  $R_3=3,3$  кОм (SMD-1206-3,3кОм). Выберем заведомо меньше, чтобы транзистор  $VT_1$  быстрее входил в режим насыщения. Замок закреплен при помощи разъемов DRB-25FA.

На плакате «Временные диаграммы» приведены временные диаграммы зарядки, разрядки аккумуляторных батарей и потребляемые токи в различных режимах работы.

### 3.2 Система идентификации

Выбрали кодовую панель ЦИФРАЛ CD – 96М. Габаритные и установочные размеры кодовой панели приведены на рисунке 3.2.

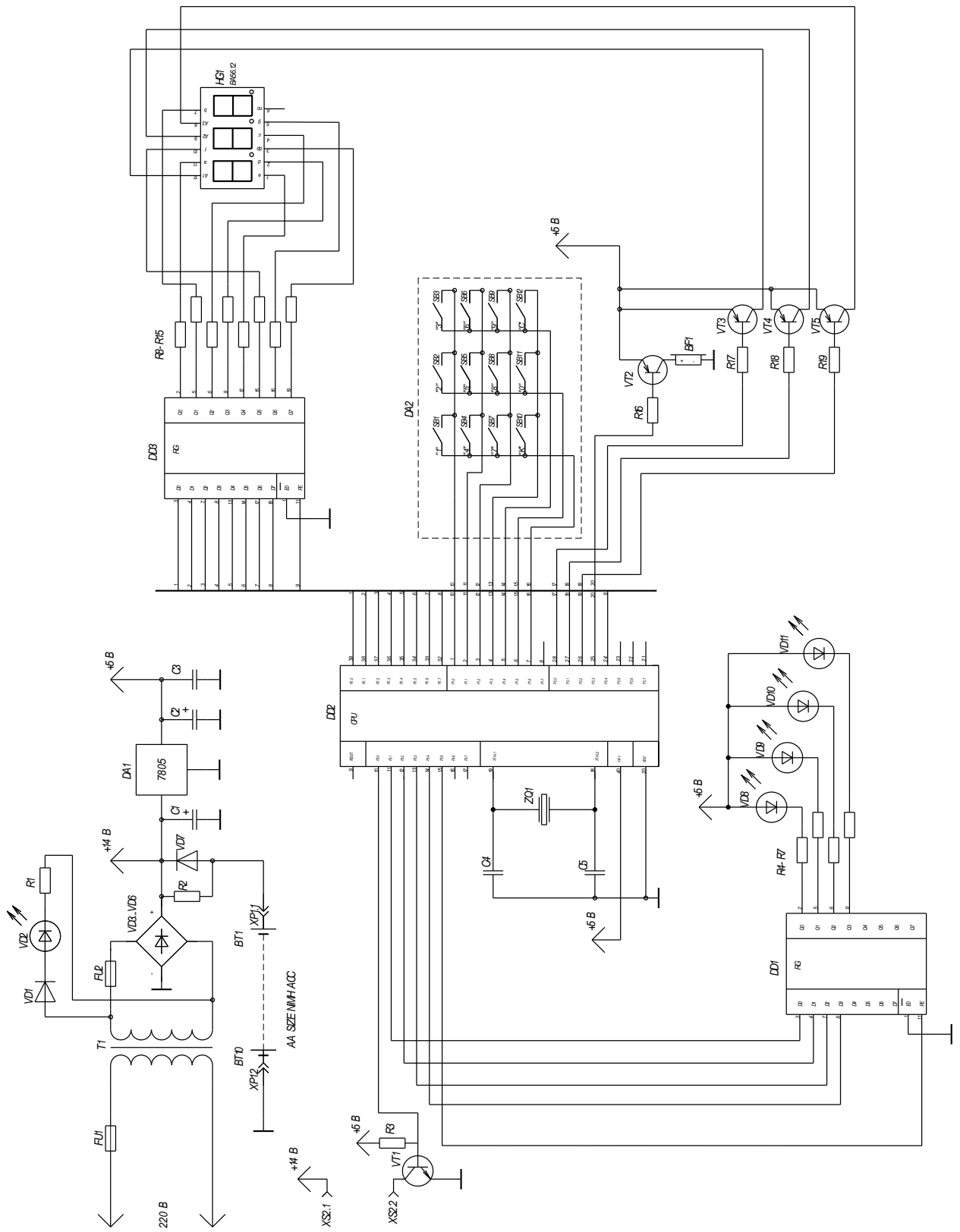


Рисунок 3.1 - Принципиальная схема замка.

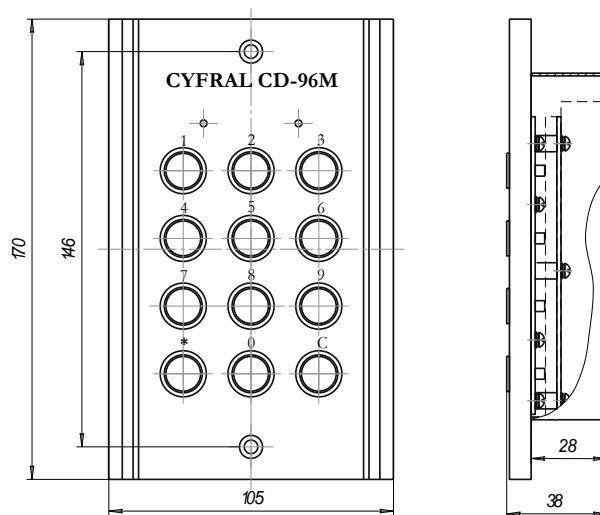


Рисунок 3.2 - Габаритные и установочные размеры панели кодовой ЦИФРАЛ CD-96М.

### 3.3 Вспомогательные схемы

#### 3.3.1 Микроконтроллер

Взяли микроконтроллер AT89S8252 подробно описан в [8].

#### 3.3.2 Блок питания

Сетевое напряжение 220 В понижается трансформатором напряжения ОСМ-0,1 220/5-12, напряжение первичной обмотки которого 220 В, а вторичной 12 В до 12 В и выпрямляется диодным мостом VD<sub>3</sub>-VD<sub>6</sub>. Диодный мост VD<sub>3</sub>-VD<sub>6</sub> выбирается по обратному напряжению (100В>14В) и току, где максимально возможный ток потребления схемы:

$$I_{\text{пот}} = I_{\text{зам}} + 2I_{\text{рег}} + I_{\text{МК}} + 4I_{\text{VD}} + I_{\text{HG}} + I_{\text{BF}} = \quad (3.3)$$

$$= 480 + 2 \cdot 20 + 25 + 4 \cdot 10 + 70 + 40 = 695 \text{ мА},$$

где  $I_{\text{зам}}$  – ток потребления замка, мА;

$I_{\text{рег}}$  – ток потребления регистров DD<sub>1</sub> и DD<sub>3</sub>, мА;

$I_{\text{МК}}$  – ток потребления микроконтроллера DD<sub>2</sub>, мА;

$I_{\text{VD}}$  – ток «горения» светодиодов VD<sub>8</sub>-VD<sub>11</sub>, мА;

$I_{\text{HG}}$  – ток работы семисегментного индикатора HG<sub>1</sub>;

$I_{\text{BF}}$  – ток работы пассивного звукового излучателя BF<sub>1</sub>.

Ток потребления схемы в «спящем» режиме:  $I_{\text{спящ}} = I_{\text{МК}} = 25 \text{ мА}$ ;

Ток потребления схемы в «рабочем» режиме:  $I_{\text{раб}} = I_{\text{МК}} + 2 I_{\text{рег}} + 4 I_{\text{VD}} + I_{\text{HG}} + I_{\text{BF}} = 25 + 2 \cdot 20 + 4 \cdot 10 + 70 + 40 = 215 \text{ мА}$ ;

Ток потребления схемы в режиме «открывания»:  $I_{\text{откр}} = I_{\text{МК}} + I_{\text{зам}} = 25 + 480 = 505 \text{ мА}$ .

Потребляемые токи в различных режимах работы изображены на рисунке 3.3.

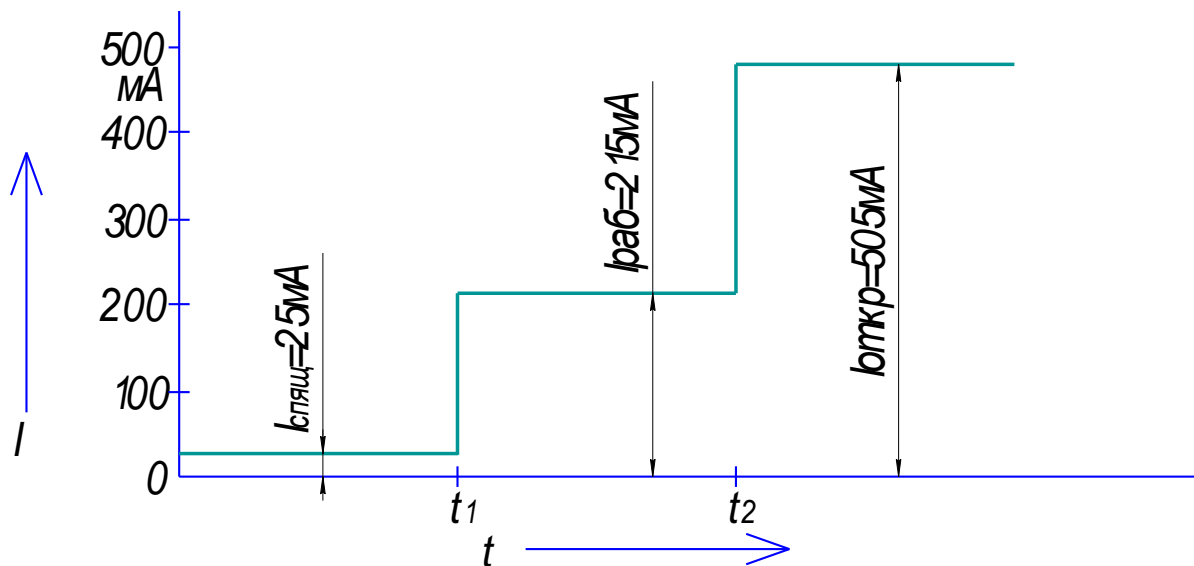


Рисунок 3.3 - Потребляемые токи в различных режимах работы.

Выбрали диодный мост (DB102) на 1 А. На выходе диодного моста напряжение будет равным:

$$U_{\text{VD3-V D6}} = U_{\text{T1-2}} \cdot \sqrt{2} - 2 \cdot U_{\text{падVD3-V D6}} = 12 \cdot \sqrt{2} - 2 \cdot 0,7 = 15,57 \text{ В}, \quad (3.4)$$

где  $U_{\text{T1-2}}$  — напряжение на вторичной обмотке трансформатора  $T_1$ , В;

$U_{\text{падVD3-V D6}}$  — падение напряжения на диодах диагонали диодного моста  $\text{VD}_3\text{-VD}_6$ .

Это напряжение подается на интегральный стабилизатор напряжения  $\text{DA}_1$ . Стабилизатор напряжения выберем таким образом, чтобы на его выходе было напряжение 5 В. Это будет  $\text{MC78L05}$ . Номинальные значения конденсаторов  $C_1$ ,  $C_2$ ,  $C_3$  возьмем из справочных данных на данный стабилизатор.  $C_1$ : К50-37-40В-10000мкФ,  $C_2$ : К50-37-50В-33мкФ,  $C_3$ : MURATA 0,1\*50В.

В качестве резервного источника питания используется батарея ВТ<sub>1</sub>-ВТ<sub>10</sub> из десяти NiMH аккумуляторов типоразмера АА емкостью E<sub>ВТ<sub>1</sub>-ВТ<sub>10</sub></sub>=800 мА·ч. При питании устройства от сети батарея аккумуляторов заряжается через резистор R<sub>2</sub> (SMD-1206-100Ом) током примерно:

$$I_{R2} = \frac{2}{3} \cdot \frac{\sqrt{2}U_{T1-2} - U_{BT1-BT10}}{R2} = \frac{2}{3} \cdot \frac{\sqrt{2} \cdot 12 - 12,5}{100} = 30 \text{ мА}, \quad (3.5)$$

где U<sub>ВТ<sub>1</sub>-ВТ<sub>10</sub></sub> – падение напряжения на батареях ВТ<sub>1</sub>-ВТ<sub>10</sub>.

В формуле (3.5) берем коэффициент 2/3, т.к. резистор R2 выбираем по действующему значению, а действующее значение равно 2/3 от амплитудного.

что составляет:

$$t_{зар} = \frac{E_{BT1-BT10}}{I_{R2}} = \frac{800}{13} = 61,5 \text{ ч}. \quad (3.6)$$

Время разрядки аккумуляторных батарей при отключенном питании:

- в «спящем» режиме:  $t_{разрспящ} = 10E_{BT1-BT10}/I_{спящ} = 10 \cdot 800/25 = 320 \text{ ч};$

- в «рабочем» режиме:  $t_{разрраб} = 10E_{BT1-BT10}/I_{раб} = 10 \cdot 800/215 = 37,2 \text{ ч};$

- в режиме «открывания»:  $t_{разроткр} = 10E_{BT1-BT10}/I_{откр} = 10 \cdot 800/505 = 15,8 \text{ ч}.$

Диод VD<sub>7</sub> шунтирует резистор R<sub>2</sub>, когда питание осуществляется от батареи. Диод VD<sub>7</sub> выбираем по максимальному току 800 мА и берем с запасом (1 А) 10BQ015. Наиболее важными характеристиками диодов Шоттки, определяющими их использование, являются низкое прямое падение напряжения и высокое быстродействие. Режим зарядки малым током называют капельным (trickle charge). В таком режиме аккумуляторы могут находиться сколь угодно долго, контроля конца процесса зарядки не требуется. Когда аккумуляторы оказываются полностью заряженными, забираемая ими от источника питания энергия превращается в тепло. Но поскольку ток зарядки очень маленький, выделяемое тепло рассеивается в окружающее пространство без заметного увеличения температуры аккумуляторов.

Аккумуляторы размещаются в стандартных пластмассовых держателях (ВН393А 10-АА), которые закреплены внутри корпуса рядом с устройством

при помощи разъемов DTD-0020. В принципе можно использовать и другие типы аккумуляторов, например 16 вольтовую кислотную необслуживаемую батарею, применяющуюся в охранных системах.

Предохранители FU<sub>1</sub> и FU<sub>2</sub> выбираем по току. FU<sub>2</sub> по I<sub>пот</sub>=695 мА, берем на 1 А, с запасом (NFCaM1). FU<sub>1</sub> находим по формуле:

$$I_{FU1} = I_{FU2} \cdot U_{T1-2} / U_{T1-1} = 1 \cdot 12 / 220 = 0,06 \text{ А}, \quad (3.7)$$

где U<sub>T1-1</sub> – напряжение на первичной обмотке трансформатора T<sub>1</sub>, В;

I<sub>FU2</sub> – номинальный ток предохранителя FU<sub>2</sub>, А.

Выбираем предохранитель FU<sub>1</sub> NFCaM0,16.

Блок VD<sub>1</sub>, VD<sub>2</sub>, R<sub>1</sub> нужен для индикации подачи питания от сети. Пусть светодиод VD<sub>2</sub> будет L-132-XIT. Ток свечения I<sub>прVD2</sub>=10 мА. Диод VD<sub>1</sub> (0C44N) выбираем, чтобы прямой ток через диод был больше 10 мА (30 мА) и максимальное обратное напряжение больше 12 В (15 В). Рассчитаем сопротивление резистора R<sub>1</sub>:

$$R_1 = U_{T1-2} - U_{\text{пад}VD1} - U_{\text{пад}VD2} / I_{\text{пр}VD2} = 12 - 0,7 - 1 / 0,01 \approx 1100 \hat{\Omega}, \quad (3.8)$$

где U<sub>падVD1</sub> – падение напряжения на диоде VD<sub>1</sub>;

U<sub>падVD2</sub> – падение напряжения на светодиоде VD<sub>2</sub>.

Возьмем резистор R<sub>1</sub>=1,1 кОм (SMD-1206-1,1кОм).

### **3.3.3 Кварцевый генератор**

На элементах C<sub>4</sub>, C<sub>5</sub> и ZQ<sub>1</sub> собран кварцевый генератор на частоту 12МГц.

Выбрали кварцевый резонатор ZQ<sub>1</sub> (НС-49S-12М), частотой 3,58 МГц. Номинальные значения конденсаторов C<sub>4</sub>, C<sub>5</sub> возьмем из справочных данных на данный резонатор. C<sub>4</sub>: TREC 30пФ\*50В-10%, C<sub>5</sub>: TREC 30пФ\*50В-10%.

### **3.3.4 Блок световой и звуковой индикации**

Блок световой и звуковой индикации состоит из семисегментного индикатора, светодиодов и пассивного звукового излучателя. Блоками управляют регистры и транзисторные ключи. Так, при подаче логической единицы на входы DD<sub>1</sub>(11) и DD<sub>3</sub>(11) разрешается трансляция данных P0.0 – P0.7 и P3.1 – P3.4 с микроконтроллера на выхода DD<sub>1</sub> и DD<sub>3</sub> соответственно. При поочередной

подаче на базу транзисторов  $VT_3 - VT_5$  логического нуля транзисторы переходят в режим насыщения и примерно +5 В передается на семисегментный индикатор ( $HG_1$ ), тем самым включая блоки индикатора. При подаче на базу транзистора  $VT_2$  логического нуля транзистор переходит в режим насыщения и примерно +5 В передается на пассивный звуковой излучатель ( $BF_1$ ), тем самым включая его. Для создания звукового сигнала на P2.3 подают поочередно логический ноль и логическую единицу. Частота смены сигнала задает частоту звука. При подаче логического нуля с выводов P3.1 – P3.4 микроконтроллера включаются соответствующие светодиоды. При подаче нуля на входы  $DD_1(11)$  и  $DD_3(11)$  происходит защёлкивание регистра и запрет трансляции данных.

В качестве регистров  $DD_1$  и  $DD_3$  выберем регистры 74ACT373PCN20A, напряжение питания у которого -0,5 - +7 В (имеется + 5 В). Ток питания 20 мА. А в качестве индикатора – красный семисегментный индикатор BA56-12SRWA, напряжение питания ( $U_{пит}$ ) которого 5 В и ток питания  $I_{питHG1}=70$  мА. Выбираем резисторы  $R_8 - R_{15}$ :

$$R_{8-15} = (U_{num} - U_{падHG1}) / I_{numHG1} = (5 - 1,85) / 10 \cdot 10^{-3} \approx 330 \text{ Ом}, \quad (3.9)$$

где  $U_{падHG1}$  – падение напряжения на одном сегменте индикатора  $HG_1$ .

Возьмем резистор  $R_{8-15}=330$  Ом (SMD-1206-330Ом).

Выбрали в качестве звукового излучателя - пассивный звуковой излучатель: НСМ1206А, у которого номинальное рабочее напряжение равно 6 В, максимальный ток 40 мА и частота 2400 Гц. Выбрали транзистор  $VT_2$  (КТ3107), с максимально допустимым током коллектора 70 мА > 40 мА, максимальным напряжением коллектор-эмиттер 45 В > 5 В и коэффициентом передачи тока  $\beta_{VT2}=70$ . Необходимо для включения пассивного звукового излучателя подать на него  $I_{кVT2}=40$  мА. Рассчитаем ток базы транзистора  $VT_2$ :

$$I_{бVT2} = I_{кVT2} / \beta_{VT2} = 40 \cdot 10^{-3} / 70 = 0,57 \cdot 10^{-3} \text{ А}, \quad (3.10)$$

Рассчитаем резистор  $R_{16}$ :

$$R_{16} = U_{num} / I_{бVT1} = 5 / 0,57 \cdot 10^{-3} \approx 9,1 \text{ кОм}, \quad (3.11)$$

Возьмем примерно на порядок меньше резистор  $R_{16}=3,7$  кОм (SMD-1206-3,7кОм), чтобы транзистор  $VT_2$  быстрее входил в режим насыщения.

Транзисторы  $VT_3 - VT_5$  выбираем с большим коэффициентом усиления по току для возможности ограничения тока базы. Выбрали транзистор КТ973А, с максимально допустимым током коллектора  $4A > 0,07A$  и коэффициентом усиления по току  $\beta_{VT_3-VT_5}=750$ . Рассчитаем ток базы транзисторов  $VT_3 - VT_5$ :

$$I_{\delta VT_3-VT_5} = I_{\kappa VT_3-VT_5} / \beta_{VT_3-VT_5} = 30 \cdot 10^{-3} / 750 = 0,04 \cdot 10^{-3} \text{ A}, \quad (3.12)$$

Рассчитаем резисторы  $R_{17}-R_{19}$ :

$$R_{17-19} = U_{num} / I_{\delta VT_3-VT_5} = 5 / 0,04 \cdot 10^{-3} = 125 \text{ кОм}, \quad (3.13)$$

Возьмем резисторы  $R_{17} - R_{19}=10$  кОм (SMD-1206-100кОм). Выберем заведомо намного меньше, чтобы транзисторы  $VT_3 - VT_5$  быстрее входили в режим насыщения.

Пусть светодиоды  $VD_8 - VD_{11}$  будут L-132-XIT. Ток свечения  $I_{npVD_8-VD_{11}}=10$  мА. Рассчитаем сопротивления резисторов  $R_4 - R_7$ :

$$R_{4-7} = (U_{num} - U_{на\delta VD_8-VD_{11}}) / I_{npVD_8-VD_{11}} = (5 - 1) / 0,01 \approx 430 \text{ Ом}, \quad (3.14)$$

где  $U_{падVD_8-VD_{11}}$  – падение напряжения на светодиодах  $VD_8 - VD_{11}$ .

Возьмем резисторы  $R_4 - R_7=510$  Ом (SMD-1206-430Ом).



## 4 Программная часть

Текст программы замка приведен в приложении А.

### 4.1 Общая блок-схема программы

#### 4.1.1 Описание блок-схемы программы

Общая блок-схемы программы приведена на рисунке 4.1.

Если кнопка «К» нажата, то переход на редактирование базы кодов, потом переход в начало, если кнопка «К» не нажата, то переход на проверку нажата ли кнопка «С». Если кнопка «С» не нажата, то запуск таймера, если нажата, то переход на чтение кода. Потом переход на проверку верен ли код, если код не верен, то переход на ошибку и возвращение в начало программы. Если код верен, то открытие двери и переход в начало программы.

#### 4.1.2 Инструкция по использованию замка:

##### **Ввод кода:**

- Нажать кнопку «С»;
- Ввести код.

##### **Редактирование кодов:**

- Нажать кнопку «К»;
- Ввести мастер-код;
- Ввести нужный код;
- Ввести уровень доступа (1-ый, 2-ой, 3-ий);
- Если записать код, то нажать кнопку «К», если удалить код, то нажать кнопку «С».

##### **Редактирование часов:**

- Нажать кнопку «К»;
- Ввести мастер-код часов;
- Загорается индикация «мин», ввести число минут;

- На дисплее загорается поциферно число минут, для подтверждения - нажатие кнопки «К», для опровержения – нажатие кнопки «С»;
- Загорается индикация «час», ввести число часов;
- На дисплее загорается поциферно число часов, для подтверждения - нажатие кнопки «К», для опровержения – нажатие кнопки «С»;
- Загорается индикация «день», ввести число дней;
- На дисплее загорается поциферно число дней, для подтверждения - нажатие кнопки «К», для опровержения – нажатие кнопки «С»;
- Загорается индикация «мес», ввести число месяцев;
- На дисплее загорается поциферно число месяцев, для подтверждения - нажатие кнопки «К», для опровержения – нажатие кнопки «С».

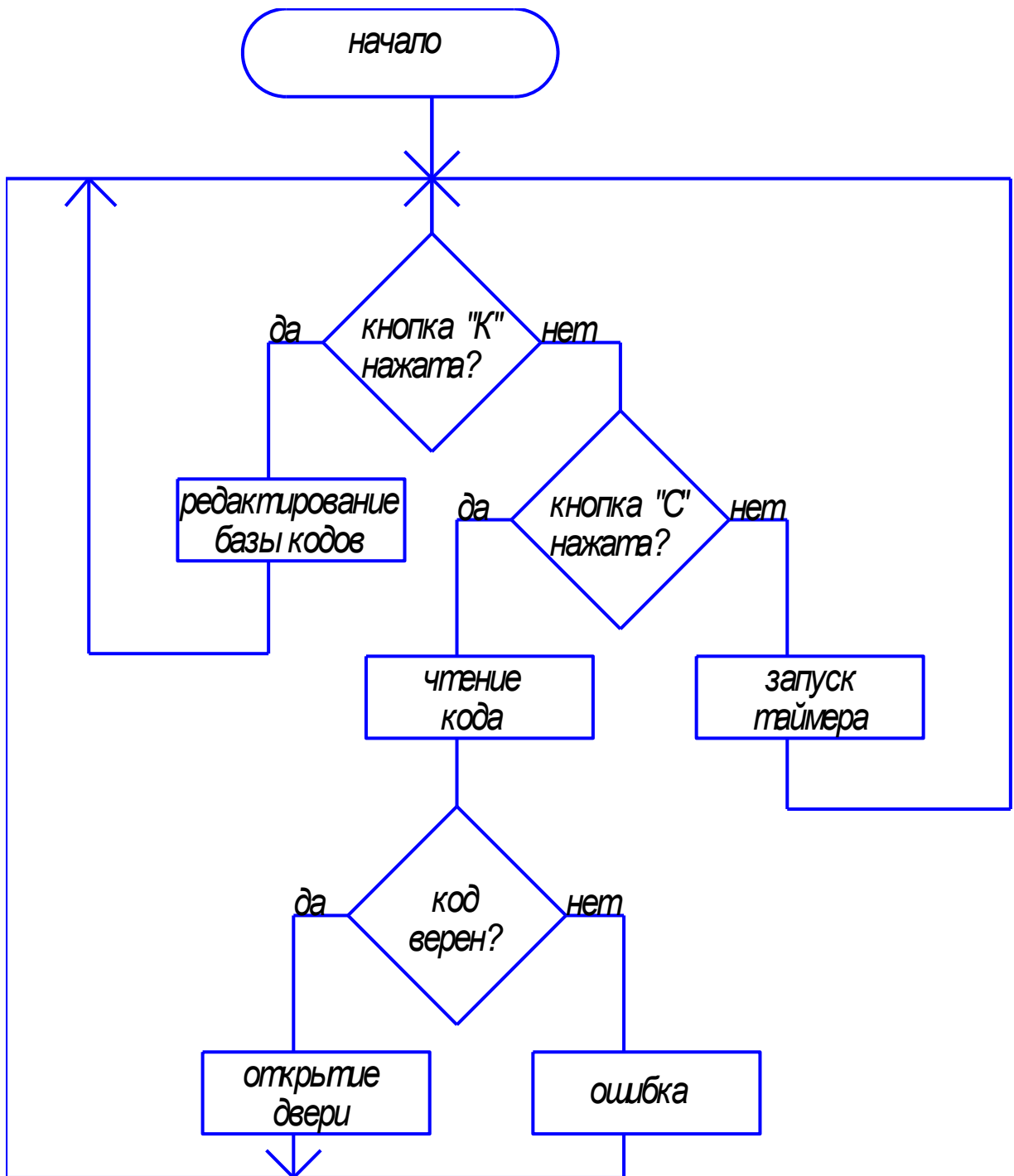


Рисунок 4.1 - Общая блок-схемы программы.

## 4.2 Подпрограмма идентификации кода

4.2.1 В R0 содержится адрес ячейки (порядковый номер цифры кода). Проверка, закончен ли ввод кода, если введено 6 цифр, чтение кода прекращается и идет его обработка. Настройка правильной работы всех кнопок (занесение «1»).

4.2.2 Занесение в 6-ой бит первого порта «0», если одна из кнопок («1», «4», «7», «K») нажата, то на соответствующем бите порта появится «0» и программа прыгнет на соответствующую метку.

4.2.3 Занесение в 5-ый бит первого порта «0», если одна из кнопок («2», «5», «8», «0») нажата, то на соответствующем бите порта появится «0» и программа прыгнет на соответствующую метку.

4.2.4 Занесение в 4-ый бит первого порта «0», если одна из кнопок («3», «6», «9», «C») нажата, то на соответствующем бите порта появится «0» и программа прыгнет на соответствующую метку.

4.2.5 Сброс. После отпускания кнопки «C», установка регистра на ввод первой цифры кода и возврат в начало подпрограммы.

4.2.6 Набор «1». После отпускания кнопки «1» повышение на 1 значения регистра R0 и возврат в начало подпрограммы.

4.2.7 Набор «2». После отпускания кнопки «2» повышение на 1 значения регистра R0 и возврат в начало подпрограммы.

4.2.8 Набор «3». После отпускания кнопки «3» повышение на 1 значения регистра R0 и возврат в начало подпрограммы.

4.2.9 Набор «4». После отпускания кнопки «4» повышение на 1 значения регистра R0 и возврат в начало подпрограммы.

4.2.10 Набор «5». После отпускания кнопки «5» повышение на 1 значения регистра R0 и возврат в начало подпрограммы.

4.2.11 Набор «6». После отпускания кнопки «6» повышение на 1 значения регистра R0 и возврат в начало подпрограммы.

4.2.12 Набор «7». После отпущения кнопки «7» повышение на 1 значения регистра R0 и возврат в начало подпрограммы.

4.2.13 Набор «8». После отпущения кнопки «8» повышение на 1 значения регистра R0 и возврат в начало подпрограммы.

4.2.14 Набор «9». После отпущения кнопки «9» повышение на 1 значения регистра R0 и возврат в начало подпрограммы.

4.2.15 Набор «0». После отпущения кнопки «0» повышение на 1 значения регистра R0 и возврат в начало подпрограммы.

4.2.16 Запись кода. Установка регистра на ввод первой цифры кода и вызов подпрограммы записи кода (zapcod).

### **4.3 Подпрограмма отпирания замка**

4.3.1 Занесение в аккумулятор первого числа paraud1 (может быть или верное число или неверное (условно приняли за «10»)). Потом сравнение его с первой цифрой kod, если не совпадает, то прыжок на метку err1 для проверки кода среднего доступа. Сравнение текущего месяца с «июнем» путем вычитания из аккумулятора числа «6», если аккумулятор меньше «6», то в бите переноса C «1», если нет, то «0». Если текущий месяц меньше числа «6», то прыжок на проверку доступа по часам. Сравнение текущего месяца с «сентябрем», если больше или равно, то прыжок на проверку по часам, иначе проверка 2-го уровня доступа. Сверка текущего часа с числом «8» путем вычитания из аккумулятора числа «8», если меньше, то проверка 2-го уровня доступа. Сверка с числом «20», если аккумулятор больше или равен, то прыжок на проверку 2-го уровня доступа. Подача «1» на порт P3.0 и замок открывается.

4.3.2 Занесение в аккумулятор первого числа paraud2 (может быть или верное число или неверное (условно приняли за «10»)). Потом сравнение его с первой цифрой kod, если не совпадает, то прыжок на метку err2 для проверки кода высшего доступа. Сверка текущего часа с числом «8» путем вычитания из

аккумулятора числа «8», если меньше, то проверка 3-го уровня доступа. Сверка с числом «20», если аккумулятор больше или равен, то прыжок на проверку 3-го уровня доступа Подача «1» на порт P3.0 и замок открывается.

4.3.3 Занесение в аккумулятор первого числа `paraud3` (может быть или верное число или неверное (условно приняли за «10»)). Потом сравнение его с первой цифрой `kod`, если не совпадает, то прыжок на метку `err3` и окончание подпрограммы. Если совпадает, то подача «1» на порт P3.0 и замок открывается.

#### **4.4 Подпрограмма задания кода**

4.4.1 Занесение значений мастер кода и кода часов в ячейки памяти. Установка начальных условий. Запись в регистр R0 адреса ячейки первой цифры введенного кода, в R1 адреса ячейки первой цифры правильного мастер кода.

4.4.2 Запись в аккумулятор первой цифры мастер кода. Запись в ячейку 0F0h первой цифры введенного кода. Сравниваем, если не равны, то прыжок на метку `terr`, если равны, то установка следующей цифры мастер кода и следующей цифры введенного кода. Проверка окончания кода, если не окончен, то прыжок на ZAK. Если окончен, то прыжок на ZK.

4.4.3 Установка начальных условий. Запись в регистр R0 адреса ячейки первой цифры введенного кода, в R1 адреса ячейки первой цифры правильного мастер кода часов.

4.4.4 Запись в аккумулятор первой цифры мастер кода часов. Запись в ячейку 0F0h первой цифры введенного кода. Сравниваем, если не равны, то прыжок на метку `terr`, если равны, то установка следующей цифры мастер кода и следующей цифры введенного кода. Проверка окончания кода, если не окончен, то прыжок на ZAK. Если окончен, то прыжок на ZC.

4.4.5 Опрос кнопок для выбора уровня доступа.

4.4.6 Вызов подпрограммы индикации цифры «1» (ind1). Опрос кнопок для выбора режима записи или режима стирания кода.

4.4.7 Установка начальных условий. Запись в регистр R0 адреса ячейки первой цифры введенного кода, в R1 адреса ячейки первой цифры правильного кода. Запись количества цифр до конца кода (для корректного переключения между кодами).

4.4.8 При поиске кода указатель R1 остановится на первой цифре последующего кода. Возвращение указателя R1 на последнюю цифру предыдущего кода путем снижения R1. Очищаем последовательно цифры кода путем подачи запрещающей комбинации (10). Выход из подпрограммы.

4.4.9 Проверка на совпадение введенного кода (@R1) путем вычитания. Установка следующей цифры введенного кода и следующей цифры правильного кода. Снижение количества цифр до конца кода. Запись разрешения на вход. При совпадении введенного и правильного кода прыжок на delk1.

4.4.10 Установка адреса начала следующего кода. Проверка последний ли код в уровне доступа.

4.4.11 Установка начальных условий. Запись в R1 адреса ячейки первой цифры правильного кода. Запись количества цифр до конца кода (для корректного переключения между кодами). Прыжок на метку spPas1.

4.4.12 Последовательное заполнение ячеек правильного кода (confirm) цифрами введенного кода (kod1, kod2, kod3, kod4, kod5, kod6). Выход из подпрограммы.

4.4.13 Сравнение путем вычитания значения правильного кода с цифрой 10. (Наличие цифры 10 говорит о том, что ячейка пуста). При совпадении правильной ячейки с цифрой 10 (ячейка пуста), прыжок на sorK1. При несовпадении переход на следующий пункт.

4.4.14 Увеличение указателя R1 на 6, выбирая следующий код. Если указатель R1 не превысил диапазон кодов 1-го уровня, то прыжок на spPas1. При

превышении – вызов подпрограммы индикации ошибки (Lcall inderr). Выход из подпрограммы.

4.4.15 Вызов подпрограммы индикации цифры «2» (ind2). Опрос кнопок для выбора режима записи или режима стирания кода.

4.4.16 Установка начальных условий. Запись в регистр R0 адреса ячейки первой цифры введенного кода, в R1 адреса ячейки первой цифры правильного кода. Запись количества цифр до конца кода (для корректного переключения между кодами).

4.4.17 При поиске кода указатель R1 остановится на первой цифре последующего кода. Возвращение указателя R1 на последнюю цифру предыдущего кода путем снижения R1. Последовательное очищение цифр кода путем подачи запрещающей комбинации (10). Выход из подпрограммы.

4.4.18 Проверка на совпадение введенного кода (@R1) путем вычитания. Установка следующей цифры введенного кода и следующей цифры правильного кода. Снижение количества цифр до конца кода. Запись разрешения на вход. При совпадении введенного и правильного кода прыжок на delk2.

4.4.19 Установка адреса начала следующего кода. Проверка последний ли код в уровне доступа.

4.4.20 Установка начальных условий. Запись в R1 адреса ячейки первой цифры правильного кода. Запись количества цифр до конца кода (для корректного переключения между кодами). Прыжок на метку spPas2.

4.4.21 Последовательное заполнение ячеек правильного кода (confirm) цифрами введенного кода (kod1, kod2, kod3, kod4, kod5, kod6). Выход из подпрограммы.

4.4.22 Сравнение путем вычитания значения правильного кода с цифрой «10». (Наличие цифры «10» говорит о том, что ячейка пуста). При совпадении правильной ячейки с цифрой «10» (ячейка пуста), прыжок на sorK2. При несовпадении переход на следующий пункт.



4.4.23 Увеличение указателя R1 на 6, выбирая следующий код. Если указатель R1 не превысил диапазон кодов 2-го уровня, то прыжок на `srPas2`. При превышении – вызов подпрограммы индикации ошибки (`Lcall inderr`). Выход из подпрограммы.

4.4.24 Вызов подпрограммы индикации цифры «3» (`ind3`). Опрос кнопок для выбора режима записи или режима стирания кода.

4.4.25 Установка начальных условий. Запись в регистр R0 адреса ячейки первой цифры введенного кода, в R1 адреса ячейки первой цифры правильного кода. Запись количества цифр до конца кода (для корректного переключения между кодами).

4.4.26 При поиске кода указатель R1 остановится на первой цифре последующего кода. Возвращение указателя R1 на последнюю цифру предыдущего кода путем снижения R1. Очищаем последовательно цифры кода путем подачи запрещающей комбинации (10). Выход из подпрограммы.

4.4.27 Проверка на совпадение введенного кода (`@R1`) путем вычитания. Установка следующей цифры введенного кода и следующей цифры правильного кода. Снижение количества цифр до конца кода. Запись разрешения на вход. При совпадении введенного и правильного кода прыжок на `delk3`.

4.4.28 Установка адреса начала следующего кода. Проверка последний ли код в уровне доступа.

4.4.29 Установка начальных условий. Запись в R1 адреса ячейки первой цифры правильного кода. Запись количества цифр до конца кода (для корректного переключения между кодами). Прыжок на метку `srPas3`.

4.4.30 Последовательное заполнение ячеек правильного кода (`confirm`) цифрами введенного кода (`kod1`, `kod2`, `kod3`, `kod4`, `kod5`, `kod6`). Выход из подпрограммы.

4.4.31 Сравнение путем вычитания значения правильного кода с цифрой 10. (Наличие цифры 10 говорит о том, что ячейка пуста). При совпадении пра-

вильной ячейки с цифрой 10 (ячейка пуста), прыжок на сорК3. При несовпадении переход на следующий пункт.

4.4.32 Увеличение указателя R1 на 6, выбирая следующий код. Если указатель R1 не превысил диапазон кодов 3-го уровня, то прыжок на срPas3. При превышении – вызов подпрограммы индикации ошибки (Lcall inderr). Выход из подпрограммы.

## **4.5 Вспомогательные подпрограммы**

### ***4.5.1 Подпрограмма таймера***

4.5.1.1 Присваивание названия адресам ячеек sek, min, chas, date, mes, sekmax, minmax, chasmax, datemax, mesmax... Это необходимо для наилучшего понимания и отладки программы.

4.5.1.2 Настройка таймера: разрешение прерывания, установка шестнадцатитбитного счетчика с младшим и старшим байтами

4.5.1.3 Установка максимального значения секунд, минут, часов. 13 mesmax взято максимальное число 13, т.к. счет идет с 1 и на 13 сбросится на 1.

4.5.1.4 Вызов «Основной подпрограммы». Запуск нулевого таймера. Прыжок на метку Zik1.

4.5.1.5 Обработка. Осуществление увеличения счетчика секунд 1 раз за 32 прерывания. После достижения 32 прерываний секунда увеличивается на 1, и ячейка 2DH обнуляется.

4.5.1.6 Необходимо для задания числа дней в месяце, т.к. в каждом их различное число. Текущий номер месяца заносится в аккумулятор и поэтапно снижается на 1 до тех пор пока не дойдет до 0. При каждом снижении задается число дней в месяце. При снижении аккумулятора до нуля перебор вариантов прекращается и присваивается последнее значение datemax, полученное при аккумуляторе равном нулю.

4.5.1.7 Собственно счетчик секунд, минут, часов, даты, месяца. Программа следит за отсчетом по одной все 60 секунд, далее счетчик секунд сбра-

сбрасывается на 0, и увеличивается счетчик минут на единицу, и возвращается на счетчик секунд. По достижении 60 минут увеличение счетчика часов на 1, и сброс минут на 0. При достижении 24 часов увеличение счетчика месяцев на 1. Месяца доходят до 13, число 13 не обрабатывается, и сразу сбрасывается на 1 месяц.

#### **4.5.2 Подпрограмма проверки кода**

4.5.2.1 Установка начальных условий. Запись в регистр R0 адреса ячейки первой цифры введенного кода, в R1 адреса ячейки первой цифры правильного кода. Запись количества цифр до конца кода (для корректного переключения между кодами).

4.5.2.2 Проверка окончания кода. Если не окончен, то прыжок на Pas1, а если окончен, то запрет 2-го и 3-го уровня доступа и на Ret (выход). Проверка на совпадение введенного кода (@R1) путем вычитания. Установка следующей цифры введенного кода и следующей цифры правильного кода. Снижение количества цифр до конца кода. Запись разрешения на вход. При совпадении введенного и правильного кода прыжок на ZPRK.

4.5.2.3 Установка адреса начала следующего кода. Проверка последний ли код в уровне доступа.

4.5.2.4 Проверка окончания кода. Если не окончен, то прыжок на Pas2, а если окончен, то запрет 3-го уровня доступа и на Ret (выход). Проверка на совпадение введенного кода (@R1) путем вычитания. Установка следующей цифры введенного кода и следующей цифры правильного кода. Снижение количества цифр до конца кода. Запись разрешения на вход. При совпадении введенного и правильного кода прыжок на ZPRK2.

4.5.2.5 Установка адреса начала следующего кода. Проверка последний ли код в уровне доступа.

4.5.2.6 Проверка окончания кода. Если не окончен, то прыжок на Pas3, а если окончен, то переход на Ret (выход). Проверка на совпадение введенного кода (@R1) путем вычитания. Установка следующей цифры введенного кода и

следующей цифры правильного кода. Снижение количества цифр до конца кода. Запись разрешения на вход. При совпадении введенного и правильного кода прыжок на ZPRK3.

4.5.2.7 Установка адреса начала следующего кода. Проверка последний ли код в уровне доступа.

### **4.5.3 Подпрограмма записи часов**

4.5.3.1 Разрешение трансляции данных на диоды VD8-VD11. Зажигание светодиода «мин». Вызов подпрограммы «нажатие кнопки» (после подпрограммы в аккумуляторе занесено число).

4.5.3.2 Перенос числа минут из аккумулятора в ячейку ed. Включение 3 блока световой индикации. Опрос кнопок «С» и «К». Если «С» нажата, то переход на метку ZCM1, если «К» нажата (подтверждение числа), то прыжок на метку ZCM2. При отсутствии нажатия прыжок на метку ZCM1X.

4.5.3.3 Вызов подпрограммы нажатия кнопки. Умножение аккумулятора на число 10. Включение 2 блока световой индикации. Опрос кнопок «С» и «К». Если «К» нажата, то прыжок на ZCC1, если «С» нажата – на ZCM2. Если не что не нажато – ZCM2X.

4.5.3.4 Сложение десятков и единиц. Получение числа минут. Гашение светодиода «мин» и зажигание светодиода «часы». Вызов подпрограммы «нажатие кнопки».

4.5.3.5 Перенос числа часов из аккумулятора в ячейку ed. Включение 3 блока световой индикации. Опрос кнопок «С» и «К». Если «С» нажата, то переход на метку ZCC1, если «К» нажата (подтверждение числа), то прыжок на метку ZCC2. При отсутствии нажатия прыжок на метку ZCC1X.

4.5.3.6 Вызов подпрограммы нажатия кнопки. Умножение аккумулятора на число 10. Включение 2 блока световой индикации. Опрос кнопок «С» и «К». Если «К» нажата, то прыжок на ZCD1, если «С» нажата – на ZCC2. Если не что не нажато – ZCC2X.

4.5.3.7 Сложение десятков и единиц. Получение числа часов. Гашение светодиода «час» и зажигание светодиода «дата». Вызов подпрограммы «нажатие кнопки».

4.5.3.8 Перенос числа дней из аккумулятора в ячейку ed. Включение 3 блока световой индикации. Опрос кнопок «С» и «К». Если «С» нажата, то переход на метку ZCD1, если «К» нажата (подтверждение числа), то прыжок на метку ZCD2. При отсутствии нажатия прыжок на метку ZCD1X.

4.5.3.9 Вызов подпрограммы нажатия кнопки. Умножение аккумулятора на число 10. Включение 2 блока световой индикации. Опрос кнопок «С» и «К». Если «К» нажата, то прыжок на ZCS1, если «С» нажата – на ZCD2. Если не что не нажато – ZCD2X.

4.5.3.10 Сложение десятков и единиц. Получение числа дней. Гашение светодиода «дата» и зажигание светодиода «мес». Вызов подпрограммы «нажатие кнопки».

4.5.3.11 Перенос числа месяцев из аккумулятора в ячейку ed. Включение 3 блока световой индикации. Опрос кнопок «С» и «К». Если «С» нажата, то переход на метку ZCS1, если «К» нажата (подтверждение числа), то прыжок на метку ZCS2. При отсутствии нажатия прыжок на метку ZCS1X.

4.5.3.12 Вызов подпрограммы нажатия кнопки. Умножение аккумулятора на число 10. Включение 2 блока световой индикации. Опрос кнопок «С» и «К». Если «К» нажата, то прыжок на ZCX1, если «С» нажата – на ZCS2. Если не что не нажато – ZCS2X.

4.5.3.13 Запрет трансляции данных на диоды VD8-VD11. Сложение десятков и единиц. Получение числа месяцев. Гашение светодиода «дата».

#### **4.5.4 Подпрограмма нажатия кнопок**

4.5.4.1 Опрос кнопок. Занесение в 6 бит первого порта «0», если одна из кнопок («1», «4», «7», «К») нажата, то на соответствующем бите порта появится «0» и программа прыгнет на соответствующую метку. Занесение в 5 бит первого порта «0», если одна из кнопок («2», «5», «8», «0») нажата, то на соответствующем бите порта появится «0» и программа прыгнет на соответствующую

метку. Занесение в 4 бит первого порта «0», если одна из кнопок («3», «6», «9», «С») нажата, то на соответствующем бите порта появится «0» и программа прыгнет на соответствующую метку.

4.5.4.2 Если кнопка «1» нажата. Занесение в аккумулятор числа «1». Ожидание отпускания кнопки и вызов подпрограммы индикации «1».

4.5.4.3 Если кнопка «2» нажата. Занесение в аккумулятор числа «2». Ожидание отпускания кнопки и вызов подпрограммы индикации «2».

4.5.4.4 Если кнопка «3» нажата. Занесение в аккумулятор числа «3». Ожидание отпускания кнопки и вызов подпрограммы индикации «3».

4.5.4.5 Если кнопка «4» нажата. Занесение в аккумулятор числа «4». Ожидание отпускания кнопки и вызов подпрограммы индикации «4».

4.5.4.6 Если кнопка «5» нажата. Занесение в аккумулятор числа «5». Ожидание отпускания кнопки и вызов подпрограммы индикации «5».

4.5.4.7 Если кнопка «6» нажата. Занесение в аккумулятор числа «6». Ожидание отпускания кнопки и вызов подпрограммы индикации «6».

4.5.4.8 Если кнопка «7» нажата. Занесение в аккумулятор числа «7». Ожидание отпускания кнопки и вызов подпрограммы индикации «7».

4.5.4.9 Если кнопка «8» нажата. Занесение в аккумулятор числа «8». Ожидание отпускания кнопки и вызов подпрограммы индикации «8».

4.5.4.10 Если кнопка «9» нажата. Занесение в аккумулятор числа «9». Ожидание отпускания кнопки и вызов подпрограммы индикации «9».

4.5.4.11 Если кнопка «0» нажата. Занесение в аккумулятор числа «0». Ожидание отпускания кнопки и вызов подпрограммы индикации «0».

#### ***4.5.5 Подпрограмма индикации***

4.5.5.1 Разрешение трансляции комбинации путем подачи на бит P2.4 «1». Подача соответствующей комбинации («0») на дисплей. Запрет трансляции комбинации путем подачи на бит P2.4 «0»

4.5.5.2 Разрешение трансляции комбинации путем подачи на бит P2.4 «1». Подача соответствующей комбинации («1») на дисплей. Запрет трансляции комбинации путем подачи на бит P2.4 «0».

4.5.5.3 Разрешение трансляции комбинации путем подачи на бит P2.4 «1». Подача соответствующей комбинации («2») на дисплей. Запрет трансляции комбинации путем подачи на бит P2.4 «0».

4.5.5.4 Разрешение трансляции комбинации путем подачи на бит P2.4 «1». Подача соответствующей комбинации («3») на дисплей. Запрет трансляции комбинации путем подачи на бит P2.4 «0».

4.5.5.5 Разрешение трансляции комбинации путем подачи на бит P2.4 «1». Подача соответствующей комбинации («4») на дисплей. Запрет трансляции комбинации путем подачи на бит P2.4 «0».

4.5.5.6 Разрешение трансляции комбинации путем подачи на бит P2.4 «1». Подача соответствующей комбинации («5») на дисплей. Запрет трансляции комбинации путем подачи на бит P2.4 «0».

4.5.5.7 Разрешение трансляции комбинации путем подачи на бит P2.4 «1». Подача соответствующей комбинации («6») на дисплей. Запрет трансляции комбинации путем подачи на бит P2.4 «0».

4.5.5.8 Разрешение трансляции комбинации путем подачи на бит P2.4 «1». Подача соответствующей комбинации («7») на дисплей. Запрет трансляции комбинации путем подачи на бит P2.4 «0».

4.5.5.9 Разрешение трансляции комбинации путем подачи на бит P2.4 «1». Подача соответствующей комбинации («8») на дисплей. Запрет трансляции комбинации путем подачи на бит P2.4 «0».

4.5.5.10 Разрешение трансляции комбинации путем подачи на бит P2.4 «1». Подача соответствующей комбинации («9») на дисплей. Запрет трансляции комбинации путем подачи на бит P2.4 «0».

4.5.5.11 Разрешение трансляции комбинации путем подачи на бит P2.4 «1». Установка комбинации «E». Зажигание первого сегмента индикатора (по-

дача «0»). Задержка. Отключение (подача «1») первого сегмента индикатора. Выставление комбинации «r», зажигание второго сегмента индикации. Выставление комбинации «r», зажигание третьего сегмента индикации. Запрет трансляции комбинации путем подачи на бит P2.4 «0». Вызов подпрограммы звонка. Если кнопка «С» нажата, то выход из подпрограммы индикации ошибки, если нет, то прыжок на метку Zvon.

4.5.5.12 Создание звуковых колебаний 255 раз, что равно 62мс.

#### **4.5.6 Подпрограмма основная**

4.5.6.1o Если кнопка «К» нажата, то прыжок на метку мастер кода (vizovMK). Если кнопка «С» нажата, то прыжок на метку чтения кода (vizovK). Если кнопки не нажаты, то прыжок на метку start.

4.5.6.2o Вызов подпрограммы чтения кода (CHTCOD). Вызов подпрограммы проверки кода (PRK), вызов подпрограммы открытия аудитории (checkaud1). Прыжок на метку start1.

4.5.6.3o Вызов подпрограммы чтения кода (CHTCOD), записи кода (ZAPCOD), прыжок на метку start.



## Заключение

В данной работе спроектирован кодовый замок с микроконтроллерным управлением, для жилых и рабочих помещений, питающейся от сети. Были разработаны блоки формирователя отпирающего импульса, системы идентификации, блока питания, блока световой и звуковой индикации; написаны подпрограммы: идентификации кода, отпираания задания кода, таймера, проверки кода, записи часов и индикации. Замок осуществлен на микроконтроллере AT89S8252.

Эта система может быть применена для обеспечения безопасности помещения. Это могут быть жилые, рабочие помещения. Применение этой системы приведет к дополнительным удобствам, например, существуют несколько лабораторий, офисных помещений и ряд сотрудников, которым необходимо попадать только в определенные помещения. Для этого каждому сотруднику выдается персональный код, который дает ему возможность открыть дверь только необходимых помещений, а в остальных ему будет отказано в доступе. Нет необходимости носить при себе много ключей, которые можно перепутать, потерять, поломать. Предполагается установить специальную панель, не позволяющую постороннему человеку увидеть набираемый код.

## Список используемой литературы

1. <http://tech.freelook.msk.ru>. Научный электротехнический сайт «Свободный взгляд».
2. Прядилов А.В. Основы микропроцессорной техники: учебно-методическое пособие для студентов 210106 заочной формы обучения, 2007. – 82 с.
3. Бальян Р.Х. Трансформаторы для радио электроники. – М.: Сов. радио, 1971. 720с.
4. Кудряшов Б.П., Назаров Ю.В., Тарабрин Б.В., Куйбышев В.А.. Аналоговые интегральные микросхемы: справочник – М.: Радио и связь, 1981 г. - 160 с
5. Справочник по электронике для молодого рабочего: 4-е изд., переработанное и дополненное – М.: Высшая школа 1987 г. – 272с.
6. Андреев Ю.Н., Антонян А.И., Иванов Д.М. Резисторы: справочник; под редакцией Четвертакова И.И. – М.: Энергоатомиздат, 1981. – 352 стр.
7. Берзан В.П., Геликман Б.Ю., Гураевский М.Н. Электрические конденсаторы и установки: справочник; под редакцией Кучинского Г.С. – М.: Энергоатомиздат, 1987г. – 656 стр.
8. Справочное пособие: <http://www.atmel.com>
9. Однокристалльные микроЭВМ. - М.:МИКАП, 1994. – 400 с.:ил.
10. Сташин В.В. и др. Проектирование цифровых устройств на однокристалльных микроконтроллерах./В.В. Сташин, А.В. Урусов,О.Ф. Мологонцева. – М.: Энергоатомиздат, 1990. – 224 с.
11. Б.В. Тарабрин, С.В. Якубовский, Н.А. Барканов и др.: справочник по интегральным микросхемам. – М.: Энергия, 1980. – 816 с., ил.
12. Воронин, П.А. Силовые полупроводниковые ключи: семейства, характеристики, применение/ П.А. Воронин. - М.:Издательский дом “Додека-XXI”, 2001. – 384 с.

13. Зиновьев, Г.С. Основы силовой электроники: учеб.пособ./ Зиновьев, Г.С. - Новосибирск: Изд-во НГТУ, 2003. - 664с.
14. Смердов, В. Силовые модули на основе биполярных транзисторов с изолированным затвором / В. Смердов, А. Образцов // Электронные компоненты. - 2005 . - №1.
15. Флоренцев, С.Н. Состояние и тенденции развития силовых IGBT-модулей // Электротехника.- 2000.- №4.
16. Гуртов В. Твердотельная электроника : учеб. пособие для вузов / В. Гуртов. - 2-е, доп. изд. ; Гриф УМО. - Москва : Техносфера, 2007. - 407 с. : ил. - (Мир электроники).
17. Ямпурин Н. П. Электроника [Электронный ресурс] : учеб. пособие для студ. высш. учеб. заведений / Н. П. Ямпурин, А. В. Баранова, В. И. Обухов. - М. : Академия, 2011.
18. Подкин Ю. Г. Электротехника и электроника [Электронный ресурс]. В 2 т. Т. 2. Электроника : учеб. пособие для студ. высш. учеб. заведений / Ю. Г. Подкин, Т. Г. Чикуров, Ю. В. Данилов; под ред. Ю. Г. Подкина. - Гриф УМО. - М. : Академия, 2011.
19. Забродин Ю.С. Промышленная электроника. – М.: Высшая школа, 1982. – 496с.
20. Белов Г. А. Электроника и микроэлектроника : учеб. пособие для вузов / Г. А. Белов. - 2-е изд., испр. ; Гриф МО. - Чебоксары : Изд-во Чуваш. ун-та, 2001. - 377 с. : ил.
21. Миловзоров О. В. Электроника : учеб. для вузов / О. В. Миловзоров, И. Г. Панков. - Изд. 4-е, стер. ; Гриф МО. - М. : Высш. шк., 2008. - 288 с.

### Листинг программы

.include '89s8252.inc' ;подключение файла описания регистров процессора  
.ASEG ROM, CODE ;эта директива объявляет новый непереключаемый  
;сегмент в области памяти программ

#### **; Таймер:**

Mkod1 .equ 20H ;1)адрес ячейки Mkod1  
Skod1 .equ 26H ;адрес ячейки Skod1  
sek .equ 30H ;адрес ячейки sek  
min .equ 31H ;адрес ячейки min  
chas .equ 32H ;адрес ячейки chas  
date .equ 33H ;адрес ячейки date  
mes .equ 34H ;адрес ячейки mes  
sekmax .equ 35H ;адрес ячейки sekmax  
minmax .equ 36H ;адрес ячейки minmax  
chasmax .equ 37H ;адрес ячейки chasmax  
datemax .equ 38H ;адрес ячейки datemax  
mesmax .equ 39H ;адрес ячейки mesmax  
kod1 .equ 40H ;адрес ячейки kod1  
kod2 .equ 41H ;адрес ячейки kod2  
kod3 .equ 42H ;адрес ячейки kod3  
kod4 .equ 43H ;адрес ячейки kod4  
kod5 .equ 44H ;адрес ячейки kod5  
kod6 .equ 45H ;адрес ячейки kod6  
kod7 .equ 46H ;адрес ячейки kod7  
cont .equ 47H ;адрес ячейки cont  
aud1 .equ P3.0 ;адрес бита aud1  
aud2 .equ P3.0 ;адрес бита aud2  
aud3 .equ P3.0 ;адрес бита aud3  
confirm01 .equ 50H ;адрес ячейки confirm01

```

confirm11 .equ 0B4H ;адрес ячейки confirm11
confirm21 .equ 0CCH ;адрес ячейки confirm21
confirm31 .equ 0D8H ;адрес ячейки confirm31
ed .equ 0F0H ;адрес ячейки ed
;для получения частоты 32 Гц (31250мкс)
;прерывание срабатывает 32 раза в 1с
;(эквивалентно ст. и мл. байту от 65535-31250)
time .equ .NOT(31250)+1 ;присваивание аббревиатуре time, NOT (31250)+1
TimeH .equ .HIGH(Time) ;старший байт 16-битного счетчика
TimeL .equ .LOW(Time) ;младший байт 16-битного счетчика
.ORG 000H
jmp Start1 ;прыжок на метку Start1
.ORG 00BH ;прерывание от таймера T0
T0_emu:
jmp T0_obr ;прыжок на метку T0_obr
.ORG 030H ;2)место размещения программы
Start1:
mov TMOD,#00001001b ;настраивание таймера
; таймер T0 настраивается на режим 1 (16-и битный счетчик TH0,TL0),
;работающий таймером. 1 на INT0 - разрешает счет. Запуск по TR0=1.
setb EA ;разрешение всех прерываний
setb ET0 ;разрешение прерывания от T0
mov TH0,#TimeH ;занесение старшего байта в счетчик
mov TL0,#TimeL ;занесение младшего байта в счетчик
mov sekmax,#60 ;3)число секунд в минуте
mov minmax,#60 ;число минут в часе
mov Chasmax,#24 ;число часов в сутках
mov mesmax,#13 ;число месяцев в году
zikl: ;4)

```

### ; Основная

clr P1.6 ;1o)очистка бита P1.6

JnB P1.3,vizovMK ;если бит P1.3=0, то прыжок на метку vizovMK

Setb P1.6 ;установка бита P1.6 (подача «1»)

clr P1.4 ;очистка бита P1.4

JnB P1.3,vizovK ;если бит P1.3=0, то прыжок на метку vizovK

Setb P1.4 ;установка бита P1.4 (подача «1»)

Jmp start ;прыжок на метку start

vizovK: ;2o)

Lcall СНТКОД ;вызов подпрограммы СНТКОД

Lcall PRK ;вызов подпрограммы PRK

Lcall checkaud1 ;вызов подпрограммы checkaud1

Jmp start ;прыжок на метку start

VizovMK: ;3o)

Lcall СНТКОД ;вызов подпрограммы СНТКОД

Lcall Zapcod ;вызов подпрограммы Zapcod

Start:

setb TR0 ;запуск нулевого таймера

jmp Zik1 ;возврат к счету

T0\_obr: ;5)подпрограмма обработки прерывания от T0

mov TH0,#TimeH ;занесение старшего байта в счетчик

mov TL0,#TimeL ;занесение младшего байта в счетчик

setb TR0 ;запуск нулевого таймера

inc 2DH ;создание меандра частота (раз в прерывании увеличивается на 1)

mov A, 2DH ;меандр в АСС

jb ACC.5,metkasek ;создание частоты 32 раза меньше исходной или 1сек

;(5 бит, т.к.  $2^5=32$ )

setb TR0 ;запуск нулевого таймера

reti ;сброс всех меток прерывания (возможен вызов прерывания повторно)

metkasek:

```
mov 2DH,#0 ;запись в ячейку 2DH цифры 0
inc sek ;Sek – ячейка счета секунд повышение секунд
;Далее установка максимальной даты
mov A,mes ;6)занесение в аккумулятор месяца
mov datemax,#31 ;занесение в ячейку datemax числа 31
dec A ;уменьшение значения аккумулятора на 1
JZ endvibor ;прыжок на endvibor, если аккумулятор = 0
mov datemax,#28 ;занесение в ячейку datemax числа 28
dec A ;уменьшение значения аккумулятора на 1
JZ endvibor ;прыжок на endvibor, если аккумулятор = 0
mov datemax,#31 ;занесение в ячейку datemax числа 31
dec A ;уменьшение значения аккумулятора на 1
JZ endvibor ;прыжок на endvibor, если аккумулятор = 0
mov datemax,#30 ;занесение в ячейку datemax числа 30
dec A ;уменьшение значения аккумулятора на 1
JZ endvibor ;прыжок на endvibor, если аккумулятор = 0
mov datemax,#31 ;занесение в ячейку datemax числа 31
dec A ;уменьшение значения аккумулятора на 1
JZ endvibor ;прыжок на endvibor, если аккумулятор = 0
mov datemax,#30 ;занесение в ячейку datemax числа 30
dec A ;уменьшение значения аккумулятора на 1
JZ endvibor ;прыжок на endvibor, если аккумулятор = 0
mov datemax,#31 ;занесение в ячейку datemax числа 31
dec A ;уменьшение значения аккумулятора на 1
JZ endvibor ;прыжок на endvibor, если аккумулятор = 0
mov datemax,#31 ;занесение в ячейку datemax числа 31
dec A ;уменьшение значения аккумулятора на 1
JZ endvibor ;прыжок на endvibor, если аккумулятор = 0
```

```

mov datemax,#30 ;занесение в ячейку datemax числа 30
dec A ;уменьшение значения аккумулятора на 1
JZ endvibor ;прыжок на endvibor, если аккумулятор = 0
mov datemax,#31 ;занесение в ячейку datemax числа 31
dec A ;уменьшение значения аккумулятора на 1
JZ endvibor ;прыжок на endvibor, если аккумулятор = 0
mov datemax,#30 ;занесение в ячейку datemax числа 30
dec A ;уменьшение значения аккумулятора на 1
JZ endvibor ;прыжок на endvibor, если аккумулятор = 0
mov datemax,#31 ;занесение в ячейку datemax числа 31
endvibor:
mov A,sek ;7)занесение в аккумулятор sek
CJNE A,sekmax,zikl1 ;пока число секунд не превысит 60 (прошла минута)
;счет секунд
Mov sek,#0 ;сброс счетчика секунд на «0»
Inc min ;следующая минута
mov A,Min ;подготовка к проверке счетчика минут
CJNE A,minmax,zikl1 ;проверка счетчика минут пока менее 60 счет минут
Mov min,# 0 ;сброс счетчика минут на «0»
Inc Chas ;увеличение счетчика часов
mov A,Chas ;подготовка к проверке счетчика часов
CJNE A,chasmaх,zikl1 ;проверка счетчика часов пока менее 24 счет часов
Mov chas,#0 ;сброс счетчика часов на «0»
Inc date ;увеличение даты
mov A,date ;подготовка к проверке счетчика числа
CJNE A,datemax,zikl1 ;проверка счетчика числа
Mov date,#1 ;сброс числа на «1»
Inc mes ;увеличение месяца
mov A,mes ;подготовка к проверке счетчика месяца

```



CJNE A,mesmax,zik11 ;проверка счетчика месяца пока менее 12 счет

Mov mes,#1 ;сброс счетчика месяцев на «1»

Zik11:

reti ;сброс всех меток прерывания (возможен вызов прерывания повторно)

;конец создания часов

**; Чтение кода:**

CHTKOD: ;1)

CJNE R0,#Kod7, StartOn ;если R0≠#Kod7, то прыжок на StartOn

JMP endCH ;иначе прыжок на endCH

StartOn

Mov P1,#11111111B ;занесение в порт P1 всех «1»

CLR P1.6 ;2)очистка бита P1.6

JNB P1.0, CH1 ;если P1.0=0, то прыжок в CH1

JNB P1.1, CH4 ;если P1.1=0, то прыжок в CH4

JNB P1.2, CH7 ;если P1.2=0, то прыжок в CH7

JNB P1.3, CHK ;если P1.3=0, то прыжок в CHK

Setb P1.6 ;установка бита P1.6 (подача «1»)

CLR P1.5 ;3)очистка бита P1.5

JNB P1.0, CH2 ;если P1.0=0, то прыжок в CH2

JNB P1.1, CH5 ;если P1.1=0, то прыжок в CH5

JNB P1.2, CH8 ;если P1.2=0, то прыжок в CH8

JNB P1.3, CH0 ;если P1.3=0, то прыжок в CH0

Setb P1.5 ;установка бита P1.5 (подача «1»)

CLR P1.4 ;4) очистка бит P1.4

JNB P1.0, CH3 ;если P1.0=0, то прыжок в CH3

JNB P1.1, CH6 ;если P1.1=0, то прыжок в CH6

JNB P1.2, CH9 ;если P1.2=0, то прыжок в CH9

JNB P1.3, CHC ;если P1.3=0, то прыжок в CHC

Setb P1.4 ;установка бит P1.4 (подача «1»)

JMP СHTKOD ;прыжок СHTKOD

СНС: ;5)

Mov R0, #kod1 ;запись в регистр R0 адреса kod1

JNB P1.3, \$ ;прыжок на эту строчку пока не отпустят кнопку

Jmp СHTKOD ;прыжок на СHTKOD

СН1: ;6)

Mov @R0,#1 ;в адрес, указанный в R0, занесение «1»

JNB P1.0, \$ ;прыжок на эту строчку пока не отпустят кнопку

Inc R0 ;повышение на 1 значения регистра R0

Jmp СHTKOD ;прыжок на СHTKOD

СН2: ;7)

Mov @R0,#2 ;в адрес, указанный в R0, занесение «2»

JNB P1.0, \$ ;прыжок на эту строчку пока не отпустят кнопку

Inc R0 ;повышение на 1 значения регистра R0

Jmp СHTKOD ;прыжок на СHTKOD

СН3: ;8)

Mov @R0,#3 ;в адрес, указанный в R0, занесение «3»

JNB P1.0, \$ ;прыжок на эту строчку пока не отпустят кнопку

Inc R0 ;повышение на 1 значения регистра R0

Jmp СHTKOD ;прыжок на СHTKOD

СН4: ;9)

Mov @R0,#4 ;в адрес, указанный в R0, занесение «4»

JNB P1.1, \$ ;прыжок на эту строчку пока не отпустят кнопку

Inc R0 ;повышение на 1 значения регистра R0

Jmp СHTKOD ;прыжок на СHTKOD

СН5: ;10)

Mov @R0,#5 ;в адрес, указанный в R0, занесение «5»

JNB P1.1, \$ ;прыжок на эту строчку пока не отпустят кнопку

Inc R0 ;повышение на 1 значения регистра R0

Jmp СHTKOD ;прыжок на СHTKOD  
 CH6: ;11)  
 Mov @R0,#6 ;в адрес, указанный в R0, занесение «6»  
 JNB P1.1, \$ ;прыжок на эту строчку пока не отпустят кнопку  
 Inc R0 ;повышение на 1 значения регистра R0  
 Jmp СHTKOD ;прыжок на СHTKOD  
 CH7: ;12)  
 Mov @R0,#7 ;в адрес, указанный в R0, занесение «7»  
 JNB P1.2, \$ ;прыжок на эту строчку пока не отпустят кнопку  
 Inc R0 ;повышение на 1 значения регистра R0  
 Jmp СHTKOD ;прыжок на СHTKOD  
 CH8: 13)  
 Mov @R0,#8 ;в адрес, указанный в R0, занесение «8»  
 JNB P1.2, \$ ;прыжок на эту строчку пока не отпустят кнопку  
 Inc R0 ;повышение на 1 значения регистра R0  
 Jmp СHTKOD ;прыжок на СHTKOD  
 CH9: ;14)  
 Mov @R0,#9 ;в адрес, указанный в R0, занесение «9»  
 JNB P1.2, \$ ;прыжок на эту строчку пока не отпустят кнопку  
 Inc R0 ;повышение на 1 значения регистра R0  
 Jmp СHTKOD ;прыжок на СHTKOD  
 CH0: ;15)  
 Mov @R0,#0 ;в адрес, указанный в R0, занесение «0»  
 JNB P1.3, \$ ;прыжок на эту строчку пока не отпустят кнопку  
 Inc R0 ;повышение на 1 значения регистра R0  
 Jmp СHTKOD ;прыжок на СHTKOD  
 CHK: ;16)  
 Lcall ZAPCOD ;вызов подпрограммы ZAPCOD  
 Mov R0,#kod1 ;запись в регистр R0 адреса kod1

endCH:

ret

**; Отпирание:**

check aud1: ;1)

Mov A,paraud1 ;занесение в аккумулятор первого числа paraud1

CJNE A,kod1,err1 ;если ACC≠ kod1, то прыжок на err1

MOV A,mes ;занесение в аккумулятор значения месяца

Subb A,#6 ;вычитание из аккумулятора числа «6»

JB C,t2 ;если заем C =1, то прыжок по метке t2

T1:

MOV A,mes ;занесение в аккумулятор значения месяца

Subb A,#9 ;вычитание из аккумулятора числа «9»

JnB C,t2 ;если заем C =0, то прыжок по метке t2

JMP err1 ;прыжок на метку err1

T2:

MOV A,chas ;занесение в аккумулятор значения часа

Subb A,#8 ;вычитание из аккумулятора числа «8»

JB C,err1 ;прыжок на метку err1

MOV A,chas ;занесение в аккумулятор значения часа

Subb A,#20 ;вычитание из аккумулятора числа «20»

JnB C,err1 ;прыжок на метку err1

Setb aud1;установка бита P3.0 (подача «1»)

err1: ;2)

check aud2: ; аналогичная процедура для второй двери

Mov A,paraud2 ;занесение в аккумулятор первого числа paraud2

CJNE A,kod1,err2 ;если ACC≠ kod1, то прыжок на err2

MOV A,chas ;занесение в аккумулятор значения часа

Subb A,#8 ;вычитание из аккумулятора числа «8»

JB C,err2 ;прыжок на метку err1

MOV A,chas ;занесение в аккумулятор значения часа  
 Subb A,#20 ;вычитание из аккумулятора числа «20»  
 JnB C,err2 ;прыжок на метку err1  
 Setb aud2;установка бита P3.0 (подача «1»)  
 err2: ;3)  
 check aud3: ;аналогичная процедура для третьей двери  
 Mov A,paraud31 ;занесение в аккумулятор первого числа paraud3  
 CJNE A,kod1,err3 ;если ACC≠ kod1, то прыжок на err3  
 Setb aud3;установка бита P3.0 (подача «1»)  
 err3:  
 ret

### **; Проверка кода:**

PRK: ;1)  
 MOV R0,#kod1 ;в R0 запись адреса первой цифры введенного кода  
 Mov R1,#confirm01 ;в R1 запись адреса первой цифры правильного кода  
 Mov cont,#6 ;запись количества цифр до конца кода  
 ZPRK: ;2)  
 CJNE R0,#kod7,pas1 ;если R0≠ адресу kod7, то прыжок на pas1  
 MOV paraud21,#10 ;запись цифры «10» в paraud21  
 MOV paraud31,#10 ;запись цифры «10» в paraud31  
 Ret  
 Pas1:  
 Mov A,@R0 ;занесение в аккумулятор содержимого ячейки, адрес которой  
 ;указан в R0  
 Subb A,@R1 ;вычитание из аккумулятора содержимого ячейки, адрес которой  
 ;указан в R1  
 Inc R0 ;повышение на 1 значения регистра R0  
 Inc R1 ;повышение на 1 значения регистра R1  
 Dec cont ;уменьшение на 1 количество цифр до конца кода

MOV paraud11,kod1 ;доступ в первого уровня  
 JZ ZPRK ;если аккумулятор=0, то прыжок на ZPRK  
 Mov A,cont ;3)запись в аккумулятор количество цифр до конца кода  
 Add A,R1 ;сложение содержимого аккумулятора с содержимым регистра R1  
 Mov R1,A ;запись в регистр R1 значение аккумулятора  
 MOV R0,#kod1 ;запись в регистр R0 адреса первой цифры введенного кода  
 Mov cont,#6 ;запись количества цифр до конца кода  
 CJNE R1,#confirm11,ZPRK ;если значение регистра R1≠ адресу confirm11  
 ;(начало ячеек записи кода второго уровня), то прыжок на ZPRK  
 ZPRK2: ;4)  
 CJNE R0,#kod7,pas2 ;если R0≠ адресу kod7, то прыжок на pas2  
 MOV paraud31,#10 ;запись цифры 10 в paraud31  
 Ret  
 Pas2:  
 Mov A,@R0 ; занесение в аккумулятор содержимого ячейки, адрес которой  
 ;указан в R0  
 Subb A,@R1 ; вычитание из аккумулятора содержимого ячейки, адрес которой  
 ;указан в R1  
 Inc R0 ;повышение на 1 значения регистра R0  
 Inc R1 ;повышение на 1 значения регистра R1  
 Dec cont ;уменьшение на 1 количество цифр до конца кода  
 MOV paraud11,kod1 ;доступ в первого уровня  
 MOV paraud21,kod1 ;доступ во второго уровня  
 JZ ZPRK2 ;если аккумулятор=0, то прыжок на ZPRK2  
 Mov A,cont ;5) запись в аккумулятор количество цифр до конца кода  
 Add A,R1 ;сложение содержимого аккумулятора с содержимым регистра R1  
 Mov R1,A ;запись в регистр R1 значение аккумулятора  
 MOV R0,#kod1 ;запись в регистр R0 адреса первой цифры введенного кода  
 Mov cont,#6 ;запись количества цифр до конца кода

CJNE R1,#confirm21,ZPRK2 ;если значение регистра R1≠ адресу confirm21  
 ;(начало ячеек записи кода второго уровня), то прыжок на ZPRK2  
 ZPRK3: ;6)  
 CJNE R0,#kod7,pas3 ;если R0≠ адресу kod7, то прыжок на pas3  
 Ret  
 Pas3:  
 Mov A,@R0; занесение в аккумулятор содержимого ячейки, адрес которой  
 ;указан в R0  
 Subb A,@R1; вычитание из аккумулятора содержимого ячейки, адрес которой  
 ;указан в R1  
 Inc R0 ;повышение на 1 значения регистра R0  
 Inc R1 ;повышение на 1 значения регистра R1  
 Dec cont ;уменьшение на 1 количество цифр до конца кода  
 MOV paraud11,kod1 ;доступ первого уровня  
 MOV paraud21,kod1 ;доступ в второго уровня  
 MOV paraud31,kod1 ;доступ в третьего уровня  
 JZ ZPRK3 ;если аккумулятор=0, то прыжок на ZPRK3  
 Mov A,cont ; 7) запись в аккумулятор количество цифр до конца кода  
 Add A,R1 ;сложение содержимого аккумулятора с содержимым регистра R1  
 Mov R1,A ;запись в регистр R1 значение аккумулятора  
 MOV R0,#kod1 ;запись в регистр R0 адреса первой цифры введенного кода  
 Mov cont,#6 ;запись количества цифр до конца кода  
 CJNE R1,#confirm31,ZPRK3 ;если значение регистра R1≠ адресу confirm31  
 ;(начало ячеек записи кода второго уровня), то прыжок на ZPRK3  
 err: ;ошибка запрещенной является комбинация которую  
 ;нельзя набрать на клавишах например 10  
 MOV paraud11,#10 ;запись цифры «10» в paraud11  
 MOV paraud21,#10 ;запись цифры «10» в paraud21  
 MOV paraud31,#10 ;запись цифры «10» в paraud31

Ret

**; Запись кода:**

ZAPCOD: ;1)

Mov 20H,#1 ;запись цифры «1» в ячейку 20H

Mov 21H,#7 ;запись цифры «7» в ячейку 21H

Mov 22H,#0 ;запись цифры «0» в ячейку 22H

Mov 23H,#1 ;запись цифры «1» в ячейку 23H

Mov 24H,#8 ;запись цифры «8» в ячейку 24H

Mov 25H,#7 ;запись цифры «7» в ячейку 25H

Mov 26H,#0 ;запись цифры «0» в ячейку 26H

Mov 27H,#2 ;запись цифры «2» в ячейку 27H

Mov 28H,#0 ;запись цифры «0» в ячейку 28H

Mov 29H,#6 ;запись цифры «6» в ячейку 29H

Mov 2AH,#0 ;запись цифры «0» в ячейку 2AH

Mov 2BH,#9 ;запись цифры «9» в ячейку 2BH

Mov R1,#Mkod1 ;запись в регистр R1 адреса первого числа мастер кода

Mov R0,#kod1 ;запись в регистр R0 адреса первого числа введенного кода

ZAK: ;2)

Mov A,@R1 ; занесение в аккумулятор содержимого ячейки, адрес которой  
;указан в R1

Mov 0F0h,@R0 ; занесение в ячейку 0F0h содержимого ячейки, адрес которой  
;указан в R0

CJNE A,0F0h,merr ;если ACC≠ 0F0h, то прыжок на merr

Inc R0 ;повышение на 1 значения регистра R0

Inc R1 ;повышение на 1 значения регистра R1

CJNE R0,#kod7,ZAK ;если R0≠адресу kod7, то прыжок на ZAK

JMP ZK ;прыжок на ZK

merr: ;3)

Mov R1,#Skod1 ;запись в регистр R1 адреса первого числа кода смены часов



Mov R0,#kod1 ;запись в регистр R0 адреса первого числа введенного кода  
 ZAC: ;4)  
 Mov A,@R1 ; занесение в аккумулятор содержимого ячейки, адрес которой  
 ;указан в R1  
 Mov 0F0h,@R0 ; занесение в ячейку 0F0h содержимого ячейки, адрес которой  
 ;указан в R0  
 CJNE A,0F0h,cerr ;если ACC≠ 0F0h, то прыжок на cerr  
 Inc R0 ;повышение на 1 значения регистра R0  
 Inc R1 ;повышение на 1 значения регистра R1  
 CJNE R0,#kod7,ZAC ;если R0≠адресу kod7, то прыжок на ZAC  
 JMP ZC ;прыжок на ZC  
 Cerr:  
 ret  
 ZK: ;5)  
 Lcall CHTkod ;вызов подпрограммы CHTkod  
 ZS1:  
 CLR P1.6 ;очищение бита P1.6  
 JNB P1.0,ZAP1 ;если P1.0=0, то прыжок на ZAP1  
 Setb P1.6 ;установка бита P1.6 (подача «1»)  
 CLR P1.5 ;очистка бита P1.5  
 JNB P1.0,ZAP2 ;если P1.0=0, то прыжок на ZAP2  
 Setb P1.5 ;установка бита P1.5 (подача «1»)  
 CLR P1.4 ;очистка бита P1.4  
 JNB P1.0,ZAP3 ;если P1.0=0, то прыжок на ZAP3  
 Setb P1.4 ;установка бита P1.4 (подача «1»)  
 Jmp ZS1 ;прыжок на ZS1  
 ZAP1: ;6)  
 Lcall ind1 ;вызов подпрограммы ind1  
 CLR P1.4 ;очистка бита P1.4

JNB P1.3,delkod ;если P1.3=0, то прыжок на delkod  
 Setb P1.4 ;установка бита P1.4 (подача «1»)  
 CLR P1.6 ;очистка бита P1.6  
 JNB P1.3, сорykod ;если P1.3=0, то прыжок на сорykod  
 Setb1.6 ;установка бита P1.6 (подача «1»)  
 Jmp zap1 ;прыжок на zap1  
 Delkod: ;7)  
 Mov Cont,#6 ;запись количества цифр до конца кода  
 MOV R0,#kod1 ;в R0 запись адреса первой цифры введенного кода  
 Mov R1,#confirm01 ;в R1 запись адреса первой цифры правильного кода  
 delK1:  
 CJNE R0,#kod7,dpas1 ;если R0≠адресу kod7, то прыжок на dpas1  
 Dec R1 ;8) уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ; занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ; занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ; занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ; занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ; занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ; занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Ret  
 dPas1: ;9)  
 Mov A,@R0; занесение в аккумулятор содержимого ячейки, адрес которой  
 ;указан в R0

Subb A,@R1; вычитание из аккумулятора содержимого ячейки, адрес которой  
;указан в R1

Inc R0 ;повышение на 1 значения регистра R0

Inc R1 ;повышение на 1 значения регистра R1

Dec cont ;уменьшение на 1 количество цифр до конца кода

JZ delk1 ;если аккумулятор=0, то прыжок на delk1

Mov A,cont ;10) запись в аккумулятор количество цифр до конца кода

Add A,R1 ;сложение содержимого аккумулятора с содержимым регистра R1

Mov R1,A ;запись в регистр R1 значение аккумулятора

MOV R0,#kod1 ;запись в регистр R0 адреса первой цифры введенного кода

Mov cont,#6 ;запись количества цифр до конца кода

CJNE R1,#confirm11, delK1 ;если значение регистра R1≠ адресу confirm11  
;(начало ячеек записи кода второго уровня), то прыжок на delK1

Lcall inderr ;вызов подпрограммы inderr

Copycode: ;11)

Mov Cont,#6 ;запись количества цифр до конца кода

Mov R1,#confirm01 ;в R1 запись адрес первой цифры правильного кода

Jmp cpPas1 ;прыжок на cpPas1

Сop K1:

Mov @R1,kod1 ;12) в адрес, указанный в R1, запись 1-ого числа введенного  
;кода

Inc R1 ;повышение на 1 значения регистра R1

Mov @R1,kod2 ;в адрес, указанный в R1 запись 2-ого числа введенного кода

Inc R1 ;повышение на 1 значения регистра R1

Mov @R1,kod3 ;в адрес, указанный в R1 запись 3-ого числа введенного кода

Inc R1 ;повышение на 1 значения регистра R1

Mov @R1,kod4 ;в адрес, указанный в R1 запись 4-ого числа введенного кода

Inc R1 ;повышение на 1 значения регистра R1

Mov @R1,kod5 ;в адрес, указанный в R1 запись 5-ого числа введенного кода

Inc R1 ;повышение на 1 значения регистра R1  
 Mov @R1,kod6 ;в адрес, указанный в R1 запись 6-ого числа введенного кода  
 Ret  
 crPas1: ;13)  
 Mov A,@R1; занесение в аккумулятор содержимого ячейки, адрес которой  
 ;указан в R1  
 Subb A,#10 ; вычитание из аккумулятора числа «10»  
 JZ sorp1 ;если аккумулятор=0, то прыжок на sorp1  
 Mov A,#6 ; 14) запись в аккумулятор число «6»  
 Add A,R1 ;сложение содержимого аккумулятора с содержимым регистра R1  
 Mov R1,A ;запись содержимого из аккумулятора в регистр R1  
 CJNE R1,confirm11, crPas1 ;если значение регистра R1≠ адресу confirm11  
 ;(начало ячеек записи кода второго уровня), то прыжок на crPas1  
 Lcall inderr ;вызов подпрограммы inderr  
 ret  
 ZAP2: ;15)  
 Lcall ind2 ;вызов подпрограммы ind2  
 CLR P1.4 ;очистка бита P1.4  
 JNB P1.3,delkod2 ;если P1.3=0, то прыжок на delkod2  
 Setb P1.4 ;установка бита P1.4 (подача «1»)  
 CLR P1.6 ;очистка бита P1.6  
 JNB P1.3, сорукod2 ;если P1.3=0, то прыжок на сорукod2  
 Setb1.6 ;установка бита P1.6 (подача «1»)  
 Jmp zap2 ;прыжок на zap2  
 Delkod2: ;16)  
 Mov Cont,#6 ;запись количества цифр до конца кода  
 MOV R0,#kod1 ; запись в R0 адреса первой цифры введенного кода  
 Mov R1,#confirm11 ; запись в R1 адреса первой цифры правильного кода  
 delK2:

CJNE R0,#kod7,dpas2 ;если R0≠адресу kod7, то прыжок на dpas2  
 Dec R1 ;17) уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»  
 Ret  
 dPas2: ;18)  
 Mov A,@R0; занесение в аккумулятор содержимого ячейки, адрес которой  
 ;указан в R0  
 Subb A,@R1; вычитание из аккумулятора содержимого ячейки, адрес которой  
 ;указан в R1  
 Inc R0 ;повышение на 1 значения регистра R0  
 Inc R1 ;повышение на 1 значения регистра R1  
 Dec cont ;уменьшение на 1 количество цифр до конца кода  
 JZ delk2 ;если аккумулятор=0, то прыжок на delk2  
 Mov A,cont ; 19)запись в аккумулятор количество цифр до конца кода  
 Add A,R1 ;сложение содержимого аккумулятора с содержимым регистра R1  
 Mov R1,A ;запись в регистр R1 значение аккумулятора  
 MOV R0,#kod1 ;запись в регистр R0 адреса первой цифры введенного кода  
 Mov cont,#6 ;запись количества цифр до конца кода

CJNE R1,#confirm21, delK2 ;если значение регистра R1≠ адресу confirm21  
 ;(начало ячеек записи кода второго уровня), то прыжок на delK2  
 Lcall inderr ;вызов подпрограммы inderr  
 Copycode: ;20)  
 Mov Cont,#6 ;запись количества цифр до конца кода  
 Mov R1,#confirm11 ;в R1 запись адрес первой цифры правильного кода  
 Jmp cpPas2 ;прыжок на cpPas2  
 Cop K2:  
 Mov @R1,kod1 ;21) в адрес, указанный в R1 запись 1-ого числа введенного  
 ;кода  
 Inc R1 ;повышение на 1 значения регистра R1  
 Mov @R1,kod2 ;в адрес, указанный в R1 запись 2-ого числа введенного кода  
 Inc R1 ;повышение на 1 значения регистра R1  
 Mov @R1,kod3 ;в адрес, указанный в R1 запись 3-ого числа введенного кода  
 Inc R1 ;повышение на 1 значения регистра R1  
 Mov @R1,kod4 ;в адрес, указанный в R1 запись 4-ого числа введенного кода  
 Inc R1 ;повышение на 1 значения регистра R1  
 Mov @R1,kod5 ;в адрес, указанный в R1 запись 5-ого числа введенного кода  
 Inc R1 ;повышение на 1 значения регистра R1  
 Mov @R1,kod6 ;в адрес, указанный в R1 запись 6-ого числа введенного кода  
 Ret  
 cpPas2: ;22)  
 Mov A,@R1; занесение в аккумулятор содержимого ячейки, адрес которой  
 ;указан в R1  
 Subb A,#10 ; вычитание из аккумулятора число «10»  
 JZ sorK2 ;если аккумулятор=0, то прыжок на sorK1  
 Mov A,#6 ;23) запись в аккумулятор число «6»  
 Add A,R1 ;сложение содержимого аккумулятора с содержимым регистра R1  
 Mov R1,A ;запись содержимого из аккумулятора в регистр R1

CJNE R1,confirm21,срPas2 ;если значение регистра R1≠ адресу confirm21  
 ;(начало ячеек записи кода третьего уровня), то прыжок на срPas2  
 Lcall inderr ;вызов подпрограммы inderr  
 ret  
 ZAP3: ;24)  
 Lcall ind3 ;вызов подпрограммы ind3  
 CLR P1.4 ;очистка бита P1.4  
 JNB P1.3,delkod3 ;если P1.3=0, то прыжок на delkod3  
 Setb P1.4 ;установка бита P1.4 (подача «1»)  
 CLR P1.6 ;очистка бита P1.6  
 JNB P1.3,сорукod3 ;если P1.3=0, то прыжок на сорукod3  
 Setb1.6 ;установка бита P1.6 (подача «1»)  
 Jmp zap3 ;прыжок на zap3  
 Delkod3: ;25)  
 Mov Cont,#6 ;запись количества цифр до конца кода  
 MOV R0,#kod1 ;в R0 запись адрес первой цифры введенного кода  
 Mov R1,#confirm21 ;в R1 запись адрес первой цифры правильного кода  
 delK3:  
 CJNE R0,#kod7,dpas3 ;если R0≠адресу kod7, то прыжок на dpas3  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;26) уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»  
 Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»

Dec R1 ;уменьшение на 1 значения регистра R1  
 Mov @R1,#10 ;занесение в адрес, указанный в R1, цифры «10»  
 Ret  
 dPas3: ;27)  
 Mov A,@R0 ;занесение в аккумулятор содержимого ячейки, адрес которой  
 ;указан в R0  
 Subb A,@R1 ; вычитание из аккумулятора содержимого ячейки, адрес которой  
 ;указан в R1  
 Inc R0 ;повышение на 1 значения регистра R0  
 Inc R1 ;повышение на 1 значения регистра R1  
 Dec cont ;уменьшение на 1 количество цифр до конца кода  
 JZ delk3 ;если аккумулятор=0, то прыжок на delk1  
 Mov A,cont ;28) запись в аккумулятор количество цифр до конца кода  
 Add A,R1 ;сложение содержимого аккумулятора с содержимым регистра R1  
 Mov R1,A ;запись в регистр R1 значение аккумулятора  
 MOV R0,#kod1 ;запись в регистр R0 адреса первой цифры введенного кода  
 Mov cont,#6 ;запись количества цифр до конца кода  
 CJNE R1,#confirm31, delK3 ;если значение регистра R1≠ адресу confirm31  
 ;(начало ячеек записи кода второго уровня), то прыжок на delK3  
 Lcall inderr ;вызов подпрограммы inderr  
 Copycode3: ;29)  
 Mov Cont,#6 ;запись количества цифр до конца кода  
 Mov R1,#confirm21 ;в R1 запись адрес первой цифры правильного кода  
 Jmp cpPas3 ;прыжок на cpPas3  
 Cop K3:  
 Mov @R1,kod1 ;30)в адрес, указанный в R1, запись 1-ого числа введенного  
 ;кода  
 Inc R1 ;повышение на 1 значения регистра R1  
 Mov @R1,kod2 ;в адрес, указанный в R1, запись 2-ого числа введенного кода



Inc R1 ;повышение на 1 значения регистра R1  
 Mov @R1,kod3 ;в адрес, указанный в R1, запись 3-ого числа введенного кода  
 Inc R1 ;повышение на 1 значения регистра R1  
 Mov @R1,kod4 ;в адрес, указанный в R1, запись 4-ого числа введенного кода  
 Inc R1 ;повышение на 1 значения регистра R1  
 Mov @R1,kod5 ;в адрес, указанный в R1, запись 5-ого числа введенного кода  
 Inc R1 ;повышение на 1 значения регистра R1  
 Mov @R1,kod6 ;в адрес, указанный в R1 запись 6-ого числа введенного кода  
 Ret  
 cpPas3: ;31)  
 Mov A,@R1; занесение в аккумулятор содержимого ячейки, адрес которой  
 ;указан в R1  
 Subb A,#10 ;вычитание из аккумулятора числа «10»  
 JZ cprk3 ;если аккумулятор=0, то прыжок на cprk3  
 Mov A,#6 ;32) запись в аккумулятор числа «6»  
 Add A,R1 ;сложение содержимого аккумулятора с содержимым регистра R1  
 Mov R1,A ;запись содержимого из аккумулятора в регистр R1  
 CJNE R1,confirm31, cpPas3 ;если значение регистра R1≠ адресу confirm31  
 ;(конец ячеек записи кода третьего уровня), то прыжок на cpPas3  
 Lcall inderr ;вызов подпрограммы inderr  
 Ret

### **; Запись часов**

ZC:  
 zcm1: ;1)  
 Setb P3.5 ;установка бита P3.5 (подача «1»)  
 clr P3.1 ;очистка бита P3.1  
 Lcall Najatie ;вызов подпрограммы Najatie  
 Zcm1x:  
 Mov ed,A ;2) занесение в ячейку ed содержимого аккумулятора

CLR P2.2 ;очистка бита P2.2  
Setb P2.2 ;установка бита P2.2 (подача «1»)  
CLR P1.6 ;очистка бита P1.6  
JNB P1.3, zcm2 ;если P1.3=0, то прыжок на zcm2  
Setb P1.6 ;установка бита P1.6 (подача «1»)  
CLR P1.4 ;очистка бита P1.4  
JNB P1.3, zcm1 ;если P1.3=0, то прыжок на zcm2  
Setb P1.4 ;установка бита P1.4 (подача «1»)  
Jmp zcm1x ;прыжок на метку zcm1x  
zcm2: ;3)  
Lcall Najatie ;вызов подпрограммы Najatie  
Mul A,#10 ;умножение аккумулятора на число «10»  
Zcm2x:  
CLR P2.1 ;очистка бита P2.1  
Setb P2.1 ;установка бита P2.1 (подача «1»)  
CLR P1.4 ;очистка бита P1.4  
JNB P1.3, zcm2 ;если P1.3=0, то прыжок на zcm2  
Setb P1.4 ;установка бита P1.4 (подача «1»)  
CLR P1.6 ;очистка бита P1.6  
JNB P1.3, zcc1 ;если P1.3=0, то прыжок на zcc1  
Setb P1.6 ;установка бита P1.6 (подача «1»)  
Jmp zcm2x ;прыжок на метку zcm2x  
zcc1: ;4)  
Add A,ed ;сложение содержимого аккумулятора с содержимым ячейки ed  
Mov min,A ;занесение в ячейку min содержимого аккумулятора  
setb P3.1 ;установка бита P3.1 (подача «1»)  
clr P3.2 ;очистка бита P3.2  
Lcall Najatie ;вызов подпрограммы Najatie  
Zcc1x: ;5)

Mov ed,A ;занесение в ячейку ed содержимого аккумулятора  
 CLR P2.2 ;очистка бита P2.2  
 Setb P2.2 ;установка бита P2.2 (подача «1»)

CLR P1.6 ;очистка бита P1.6  
 JNB P1.3, zcc2 ;если P1.3=0, то прыжок на zcc2  
 Setb P1.6 ;установка бита P1.6 (подача «1»)

CLR P1.4 ;очистка бита P1.4  
 JNB P1.3, zcc1 ;если P1.3=0, то прыжок на zcc1  
 Setb P1.4 ;установка бита P1.4 (подача «1»)

Jmp zcc1x ;прыжок на zcc1x  
 zcc2: ;6)

Lcall Najatie ;вызов подпрограммы Najatie

Mul A,#10 ;умножение аккумулятора на число «10»

Zcc2x:

CLR P2.1 ;очистка бита P2.1  
 Setb P2.1 ;установка бита P2.1 (подача «1»)

CLR P1.6 ;очистка бита P1.6  
 JNB P1.3, zcd1 ;если P1.3=0, то прыжок на zcd1  
 Setb P1.6 ;установка бита P1.6 (подача «1»)

CLR P1.4 ;очистка бита P1.4  
 JNB P1.3, zcc2 ;если P1.3=0, то прыжок на zcc2  
 Setb P1.4 ;установка бита P1.4 (подача «1»)

Jmp zcc2x ;прыжок на zcc2x  
 zcd1: ;7)

Add A,ed ;сложение содержимого аккумулятора с содержимым ячейки ed

Mov Chas,A ;занесение в ячейку ed содержимого аккумулятора

setb P3.2 ;установка бита P1.6 (подача «1»)

clr P3.3 ;очистка бита P1.6

Lcall Najatie ;вызов подпрограммы Najatie

Zcd1x: ;8)

Mov ed,A ;занесение в ячейку ed содержимого аккумулятора

CLR P2.2 ;очистка бита P2.2

Setb P2.2 ;установка бита P2.2 (подача «1»)

CLR P1.6 ;очистка бита P1.6

JNB P1.3, zcd2 ;если P1.3=0, то прыжок на zcd2

Setb P1.6 ;установка бита P1.6 (подача «1»)

CLR P1.4 ;очистка бита P1.4

JNB P1.3, zcd1 ;если P1.3=0, то прыжок на zcd1

Setb P1.4 ;установка бита P1.4 (подача «1»)

Jmp zcd1x ;прыжок на zcd1x

zcd2: ;9)

Lcall Najatie ;вызов подпрограммы Najatie

Mul A,#10 ;умножение аккумулятора на число «10»

Zcd2x:

CLR P2.1 ;очистка бита P2.1

Setb P2.1 ;установка бита P2.1 (подача «1»)

CLR P1.6 ;очистка бита P1.6

JNB P1.3, zcs1 ;если P1.3=0, то прыжок на zcs1

Setb P1.6 ;установка бита P1.6 (подача «1»)

CLR P1.4 ;очистка бита P1.4

JNB P1.3, zcd2 ;если P1.3=0, то прыжок на zcd2

Setb P1.4 ;установка бита P1.4 (подача «1»)

Jmp zcd2x ;прыжок на zcd2x

zcs1: ;10)

Add A,ed ;сложение содержимого аккумулятора с содержимым ячейки ed

Mov date,A ;занесение в ячейку ed содержимого аккумулятора

setb P3.3 ;установка бита P3.3 (подача «1»)

clr P3.4 ;очистка бита P3.4

```

Lcall Najatie ;вызов подпрограммы Najatie
Zcs1x: ;11)
Mov ed,A ;занесение в ячейку ed содержимого аккумулятора
CLR P2.2 ;очистка бита P2.2
Setb P2.2 ;установка бита P2.2 (подача «1»)
CLR P1.6 ;очистка бита P1.6
JNB P1.3, zcs2 ;если P1.3=0, то прыжок на zcs2
Setb P1.6 ;установка бита P1.6 (подача «1»)
CLR P1.4 ;очистка бита P1.6
JNB P1.3, zcs1 ;если P1.3=0, то прыжок на zcs1
Setb P1.4 ;установка бита P1.4 (подача «1»)
Jmp zcs1x ;прыжок на zcs1x
zcs2: ;12)
Lcall Najatie ;вызов подпрограммы Najatie
Mul A,#10 ;умножение аккумулятора на число «10»
Zcs2x:
CLR P2.1 ;очистка бита P2.1
Setb P2.1 ;установка бита P2.1 (подача «1»)
CLR P1.6 ;очистка бита P1.6
JNB P1.3, zcx1 ;если P1.3=0, то прыжок на zcx1
Setb P1.6 ;установка бита P1.6 (подача «1»)
CLR P1.4 ;очистка бита P1.4
JNB P1.3, zcs2;если P1.3=0, то прыжок на zcx2
Setb P1.4 ;установка бита P1.4 (подача «1»)
Jmp zcs2x ;прыжок на zcs2x
Zcx1: ;13)
Add A,ed ;сложение содержимого аккумулятора с содержимым ячейки ed
Mov mes,A ;занесение в ячейку ed содержимого аккумулятора
setb P3.4 ;установка бита P3.4 (подача «1»)

```

CLR P3.5 ;очистка бита P3.5

ret

### **; Нажатие**

Najatie:

Mov P1,#11111111B ;1) занесение в порт P1 всех «1»

CLR P1.6 ;очистка бита P1.6

JNB P1.0, NCH1 ;если бит P1.0=0, то прыжок на NCH1

JNB P1.1, NCH4 ;если бит P1.1=0, то прыжок на NCH4

JNB P1.2, NCH7 ;если бит P1.2=0, то прыжок на NCH7

Setb P1.6 ;установка бита P1.6 (подача «1»)

CLR P1.5 ;очистка бита P1.5

JNB P1.0,NCH2 ;если бит P1.0=0, то прыжок на NCH2

JNB P1.1,NCH5 ;если бит P1.1=0, то прыжок на NCH5

JNB P1.2,NCH8 ;если бит P1.2=0, то прыжок на NCH8

JNB P1.3,NCH0 ;если бит P1.3=0, то прыжок на NCH0

Setb P1.5 ;установка бита P1.5 (подача «1»)

CLR P1.4 ;очистка бита P1.4

JNB P1.0,NCH3 ;если бит P1.0=0, то прыжок на NCH3

JNB P1.1,NCH6 ;если бит P1.1=0, то прыжок на NCH6

JNB P1.2,NCH9 ;если бит P1.2=0, то прыжок на NCH9

Setb P1.4 ;установка бита P1.4 (подача «1»)

JMP Najatie ;прыжок на метку Najatie

ret

NCH1: ;2)

Mov A,#1 ;занесение в аккумулятор числа «1»

JNB P1.0, \$ ;прыжок на эту строчку пока не отпустят кнопку

Lcall Ind1 ;вызов подпрограммы Ind1

ret

NCH2: ;3)

```
Mov A,#2 ;занесение в аккумулятор числа «2»
JNB P1.0, $ ;прыжок на эту строчку пока не отпустят кнопку
Lcall Ind2 ;вызов подпрограммы Ind
ret
NCH3: ;4)
Mov A,#3 ;занесение в аккумулятор числа «3»
JNB P1.0, $ ;прыжок на эту строчку пока не отпустят кнопку
Lcall Ind3 ;вызов подпрограммы Ind3
ret
NCH4: ;5)
Mov A,#4 ;занесение в аккумулятор числа «4»
JNB P1.1, $ ;прыжок на эту строчку пока не отпустят кнопку
Lcall Ind4 ;вызов подпрограммы Ind4
ret
NCH5: ;6)
Mov A,#5 ;занесение в аккумулятор числа «5»
JNB P1.1, $ ;прыжок на эту строчку пока не отпустят кнопку
Lcall Ind5 ;вызов подпрограммы Ind5
ret
NCH6: ;7)
Mov A,#6 ;занесение в аккумулятор числа «6»
JNB P1.1, $ ;прыжок на эту строчку пока не отпустят кнопку
Lcall Ind6 ;вызов подпрограммы Ind6
ret
NCH7: ;8)
Mov A,#7 ;занесение в аккумулятор числа «7»
JNB P1.2, $ ;прыжок на эту строчку пока не отпустят кнопку
Lcall Ind7 ;вызов подпрограммы Ind7
ret
```

NCH8: ;9)

Mov A,#8 ;занесение в аккумулятор числа «8»

JNB P1.2, \$ ;прыжок на эту строчку пока не отпустят кнопку

Lcall Ind8 ;вызов подпрограммы Ind8

ret

NCH9: ;10)

Mov A,#9 ;занесение в аккумулятор числа «9»

JNB P1.2, \$ ;прыжок на эту строчку пока не отпустят кнопку

Lcall Ind9 ;вызов подпрограммы Ind9

ret

NCH0: ;11)

Mov A,#0 ;занесение в аккумулятор числа «0»

JNB P1.3, \$ ;прыжок на эту строчку пока не отпустят кнопку

Lcall Ind0 ;вызов подпрограммы Ind0

ret

### **; Индикация**

Ind0: ;1)

Setb P2.4 ;установка бита P2.4 (подача «1»)

Mov P0,#11000000B ;занесение в порт P0 соответствующей комбинации

clr P2.4 ;очистка бита P2.4

ret

Ind1: ;2)

Setb P2.4 ;установка бита P2.4 (подача «1»)

Mov P0,#11111001B ;занесение в порт P0 соответствующей комбинации

clr P2.4 ;очистка бита P2.4

ret

Ind2: ;3)

Setb P2.4 ;установка бита P2.4 (подача «1»)

Mov P0,#10100100B ;занесение в порт P0 соответствующей комбинации



clr P2.4 ;очистка бита P2.4

ret

Ind3: ;4)

Setb P2.4 ;установка бита P2.4 (подача «1»)

Mov P0,#10110000B ;занесение в порт P0 соответствующей комбинации

clr P2.4 ;очистка бита P2.4

ret

Ind4: ;5)

Setb P2.4 ;установка бита P2.4 (подача «1»)

Mov P0,#10011001B ;занесение в порт P0 соответствующей комбинации

clr P2.4 ;очистка бита P2.4

ret

Ind5: ;6)

Setb P2.4 ;установка бита P2.4 (подача «1»)

Mov P0,#10010010B ;занесение в порт P0 соответствующей комбинации

clr P2.4 ;очистка бита P2.4

ret

Ind6: ;7)

Setb P2.4 ;установка бита P2.4 (подача «1»)

Mov P0,#10000010B ;занесение в порт P0 соответствующей комбинации

clr P2.4 ;очистка бита P2.4

ret

Ind7: ;8)

Setb P2.4 ;установка бита P2.4 (подача «1»)

Mov P0,#11111000B ;занесение в порт P0 соответствующей комбинации

clr P2.4 ;очистка бита P2.4

ret

Ind8: ;9)

Setb P2.4 ;установка бита P2.4 (подача «1»)

```

Mov P0,#10000000B ;занесение в порт P0 соответствующей комбинации
clr P2.4 ;очистка бита P2.4
ret
Ind9: ;10)
Setb P2.4 ;установка бита P2.4 (подача «1»)
Mov P0,#10010000B ;занесение в порт P0 соответствующей комбинации
clr P2.4 ;очистка бита P2.4
ret
Inderr: ;11)
MOV 0FAh,#11111111B ;занесение в ячейку 0FAh всех «1»
Setb P2.4 ;установка бита P2.4 (подача «1»)
Mov P0,#10000110B ;занесение в порт P0 соответствующей комбинации
Clr p2.0 ;очистка бита P2.0
DJNZ 0FAh, $ ;задержка на ячейке 0FAh
setb P2.0 ;установка бита P2.0 (подача «1»)
Mov P0,#10101111B ;занесение в порт P0 соответствующей комбинации
Clr P2.1 ;очистка бита P2.1
DJNZ 0FAh, $ ;задержка на ячейке 0FAh
setb P2.1 ;установка бита P2.1 (подача «1»)
Mov P0,#10101111B ;занесение в порт P0 соответствующей комбинации
Clr P2.2 ;очистка бита P2.2
DJNZ 0FAh, $ ;задержка на ячейке 0FAh
setb P2.2 ;установка бита P2.2 (подача «1»)
clr P2.4 ;очистка бита P2.4
Lcall Zvon ;вызов подпрограммы Zvon
JB P1.3,inderr ; если P1.3=1, то прыжок в inderr
ret
Zvon: ;12)
Setb P2.3 ;установка бита P2.3 (подача «1»)

```

DJNZ 0FAh, \$ ;задержка на ячейке 0FAh

Dec 0fch ;уменьшение значения ячейки 0fch на 1

clr P2.3 ;очистка бита P2.3

DJNZ 0FAh, \$ ;задержка на ячейке 0FAh

Mov A, 0FCh ; запись в аккумулятор значения ячейки 0FCh

Jnz zvon ;если аккумулятор $\neq$ 0, то прыжок на метку zvon

ret

Jmp start1 ;прыжок на метку start

End.