

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Шебзухова Татьяна Александровна

Должность: Директор Пятигорского института (филиал) Северо-Кавказского
федерального университета

Дата подписания: 27.05.2025 15:42:02

Уникальный программный ключ:

d74ce93cd40e39275c3ba2f58486412a1c8ef96f

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение
высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ

Колледж Пятигорского института (филиал) СКФУ

МДК 02.02. Программирование микроконтроллеров

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Специальности СПО

09.02.01 Компьютерные системы и комплексы

Квалификация: Техник по компьютерным системам

Пятигорск, 2025

Методические указания для лабораторных работ по дисциплине МДК.02.02
Программирование микроконтроллеров составлены в соответствии с требованиями
ФГОС СПО. Предназначены для студентов, обучающихся по специальности 09.02.01
Компьютерные системы и комплексы

В результате освоения учебной дисциплины обучающийся должен **уметь:**

- использовать программное обеспечение в профессиональной деятельности;
- использовать информационные ресурсы для поиска и хранения информации;
- обрабатывать текстовую и табличную информацию;
- использовать деловую графику и мультимедиа информацию;
- использовать технологии сбора, размещения, хранения, накопления, преобразования и передачи данных;
- обрабатывать текстовую и числовую информацию;
- применять мультимедийные технологии обработки и представления информации;
- обрабатывать информацию, используя средства пакетов прикладных программ.

знать:

- понятие информационных систем и информационных технологий, автоматизированной обработки информации;
- основные правила и методы работы с пакетами прикладных программ; □ возможности сетевых технологий работы с информацией;
- методы и средства сбора, обработки, хранения, передачи и накопления информации;
- принципы защиты информации от несанкционированного доступа;
- теоретические основы, виды и структуру баз данных;
- принципы классификации и кодирования информации;
- номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации;
- формат оформления результатов поиска информации; основы современных систем управления базами данных.

ЛАБОРАТОРНАЯ РАБОТА №1

ИЗУЧЕНИЕ ПРОЦЕДУРЫ ПРОГРАММИРОВАНИЯ МИКРОКОНТРОЛЛЕРА

Цель работы: закрепление и углубление лекционного материала по структуре и принципам функционирования микроконтроллера AT89S8252; изучение структуры и программного обеспечения программатора, приобретение практических навыков в программировании микроконтроллера AT89S8252

1 ОПИСАНИЕ ЛАБОРАТОРНОЙ УСТАНОВКИ

Рабочее место в лаборатории состоит из персонального компьютера (ПК) и специализированного лабораторного макета. Лабораторный макет содержит плату электронного блока (ЭБ) и блок питания (БП), которые заключены в стандартный корпус Z17. В корпусе лабораторного макета вырезано окно для наглядной демонстрации состава ЭБ.

Структурная схема макета представлена на рисунке 1.1. В его состав входят:

1. МК – микроконтроллер;
2. КОМ – аналоговый коммутатор;
3. АЦП – последовательный аналогово-цифровой преобразователь;
4. ЦАП – цифро-аналоговый преобразователь;
5. ПИ – последовательный интерфейс;
6. ЦОУ – цифровое отсчетное устройство;
7. СИ – система индикации;
8. ПУ – панель управления;
9. СУ – система управления блоками индикации и клавиатуры;
10. БП – блок питания; 11. Аналоговая часть;
12. Программатор.

Верхняя панель макета представлена на рисунке 1.2. На рисунке 1.2 выделены блоки, нумерация которых соответствует перечисленным блокам структурной схемы.

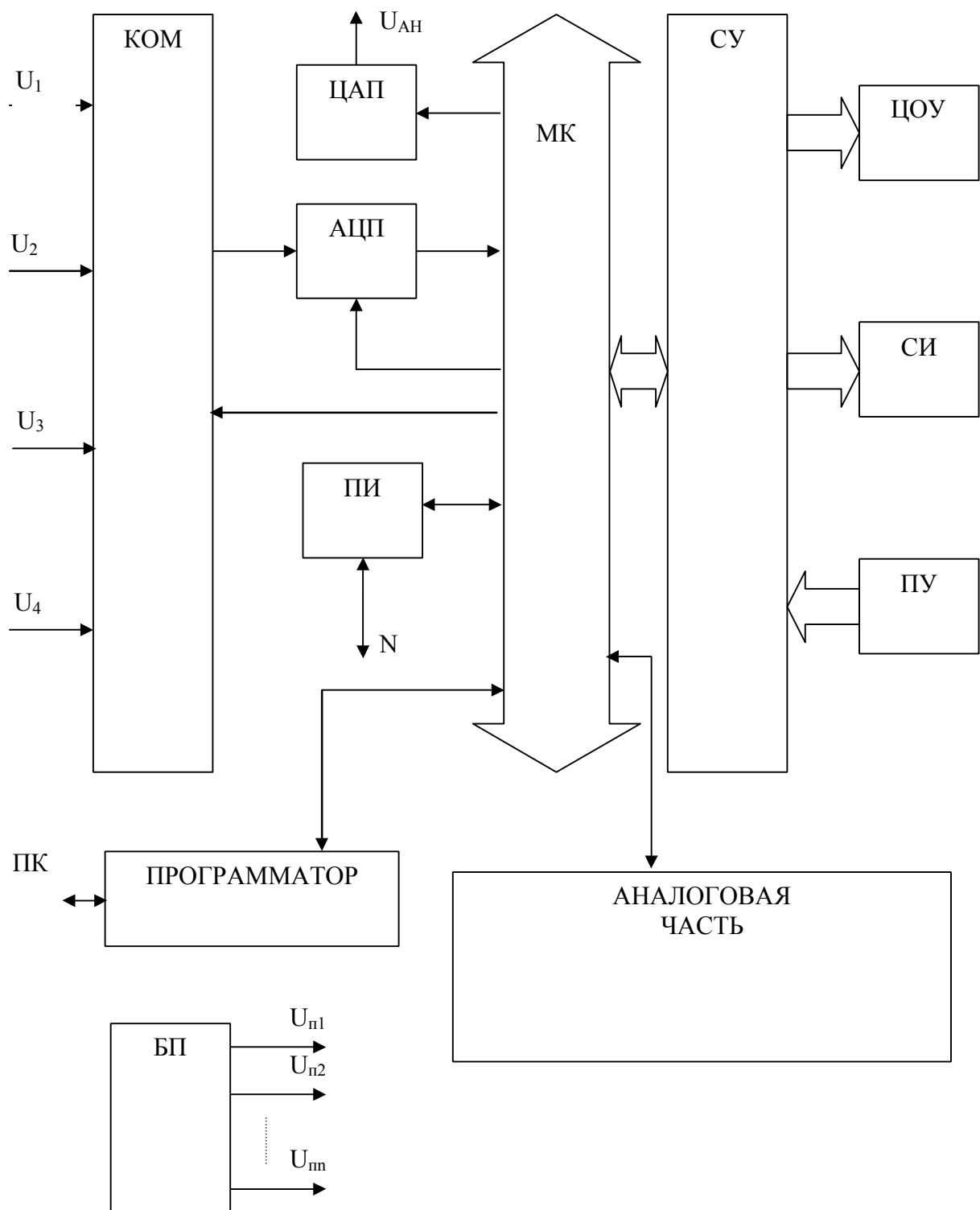


Рисунок 1.1 – Структурная схема лабораторного макета

Лабораторный макет содержит внешние разъемы для связи с ПК, 232- и 485- интерфейсов, а также разъем для подключения к сети.

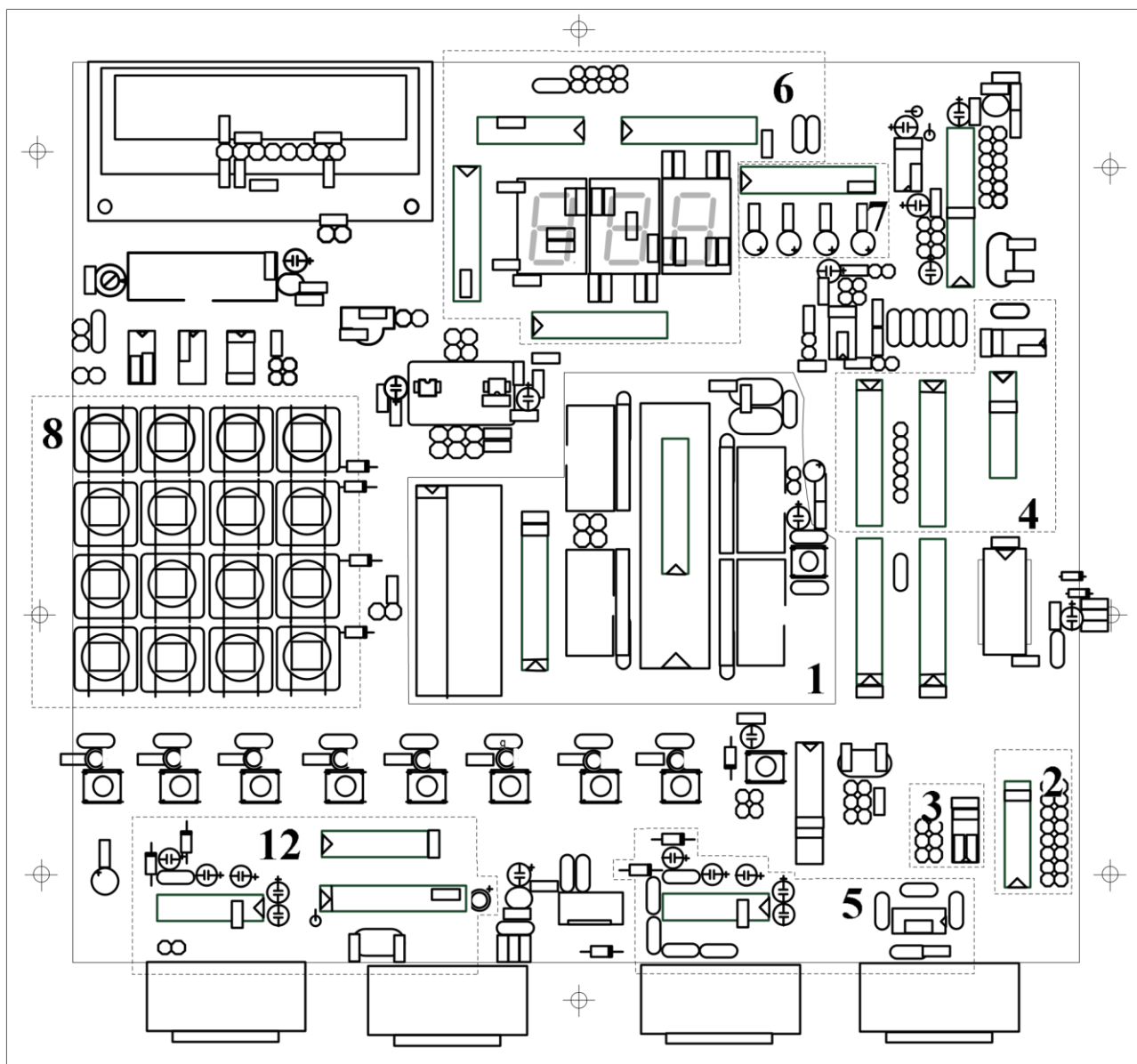


Рисунок 1.2 – Верхняя панель макета

2 АРХИТЕКТУРА И ПРОСТЕЙШИЕ ПРИНЦИПЫ ФУНКЦИОНИРОВАНИЯ МИКРОКОНТРОЛЛЕРА AT89S8252

Микроконтроллер семейства AT89 фирмы Atmel представляет собой восьмиразрядную однокристальную микроЭВМ с системой команд MCS-51 фирмы Intel. Микроконтроллеры изготавливаются по КМОП (CMOS) технологии и имеют статическую структуру. В состав семейства входят микроконтроллеры 15-ти типов.

МК семейства AT89 выпускаются для работы при разных значениях напряжения питания и тактовой частоты, определяемой частотой

подключенного к МК кварцевого резонатора. Диапазоны значений напряжения питания (V_{CC}) и тактовой частоты (F_{OSC}) у МК различных типов указаны в таблице 1.1. Ток потребления зависит от величины напряжения питания и тактовой частоты. В таблице 1.1 приведены значения тока потребления в рабочем режиме (I_{CC}) при максимальном значении напряжения питания и $F_{OSC}=12$ МГц.

Таблица 1.1 – Параметры МК

Тип МК	Vcc (В)	Fosc (МГц)	Icc (мА)	N
AT89C1051	2,7-6,0	0-24	15	20
AT89C1051U	2,7-6,0	0-24	15	20
AT89C2051	2,7-6,0	0-24	15	20
AT89C4051	2,7-6,0	0-24	15	20
AT89C51	4,0-6,0	0-24	20	40
AT89LV51	2,7-6,0	0-12	20	40
AT89C52	4,0-6,0	0-24	25	40
AT89C55	4,0-6,0	0-33	25	40
AT89S53	4,0-6,0	0-33	25	40
AT89LS53	2,7-6,0	0-12	25	40
AT89S8252	4,0-6,0	0-33	25	40
AT89LS8252	2,7-6,0	0-12	25	40

МК выпускаются в корпусах различных типов с разным числом выводов, при этом число выводов, используемых для подключения микроконтроллера к схеме устройства, может отличаться от числа выводов корпуса.

В данных лабораторных работах используется микроконтроллер AT89S8252. Условное обозначение МК и назначение его выводов представлены на рисунке 1.3.

МК содержит ряд выводов, которые можно разделить на вспомогательные, функциональные и порты. Вспомогательные выводы необходимы для подачи питания (V_{CC} , GND), подключения кварцевого резонатора (X_1 , X_2), необходимого для синхронизации действий в МК, подачи сигнала RES (сброс), необходимого для запуска МК, выводы для подключения программатора (MOSI, MISO, SCK). Порты МК (P_0 , P_1 , P_2 , P_3) представляют собой выводы, которые в зависимости от выполняемой программы, могут принимать значение логического "0" или логической "1". Функциональные выводы предназначены для перевода МК в режим приемо-

передатчика (RxD, TxD), в режим таймера-счетчика (T0, T1), в режим прерывания (INT0, INT1), в режим доступа к внешней памяти (WR, RD, EA, ALE, PSEN). Функциональные выводы МК могут работать и как обычный порт P₃.

RXD	10	P3.0/RXD	MCU	P0.0/AD0	39	AD0
TxD	11	P3.1/TxD		P0.1/AD1	38	AD1
/INT0	12	P3.2/ $\overline{\text{INT0}}$		P0.2/AD2	37	AD2
/INT1	13	P3.3/ $\overline{\text{INT1}}$		P0.3/AD3	36	AD3
T0	14	P3.4/T0		P0.4/AD4	35	AD4
T1	15	P3.5/T1		P0.5/AD5	34	AD5
/WR	16	P3.6/ $\overline{\text{WR}}$		P0.6/AD6	33	AD6
/RD	17	P3.7/ $\overline{\text{RD}}$		P0.7/AD7	32	AD7
T2	1	P1.0/T2		P2.0/A8	21	A8
T2EX	2	P1.1/T2EX		P2.1/A9	22	A9
P12	3	P1.2		P2.2/A10	23	A10
P13	4	P1.3		P2.3/A11	24	A11
/SS	5	P1.4/ $\overline{\text{SS}}$		P2.4/A12	25	A12
MOSI	6	P1.5/MOSI		P2.5/A13	26	A13
MISO	7	P1.6/MISO		P2.6/A14	27	A14
SCK	8	P1.7/SCK		P2.7/A15	28	A15
RES	9	RES		$\overline{\text{EA}}/\text{VPP}$	31	/EA
X1	19	XTAL1		ALE/ $\overline{\text{PRG}}$	30	ALE
X2	18	XTAL2		PSEN	29	/PSEN

Рисунок 1.3 – Назначение выводов МК AT89S8252

Структура МК представлена на рисунке 1.4.

Основой МК AT89S8252 является внутренняя шина данных, необходимая для передачи данных внутри МК. Кроме того, МК содержит следующие основные блоки:

1. АЛУ (арифметико-логическое устройство) предназначено для выполнения арифметических и логических операций);
2. T₁, T₂ – регистры для хранения операндов;
3. А – аккумулятор, регистр, предназначенный для аккумулярования результата выполненной операции;
4. В – расширитель аккумулятора;
5. Система управления – предназначена для приема и выдачи управляющих сигналов;
6. РК – регистр команд, представляющий собой формирователь выполняемых команд;

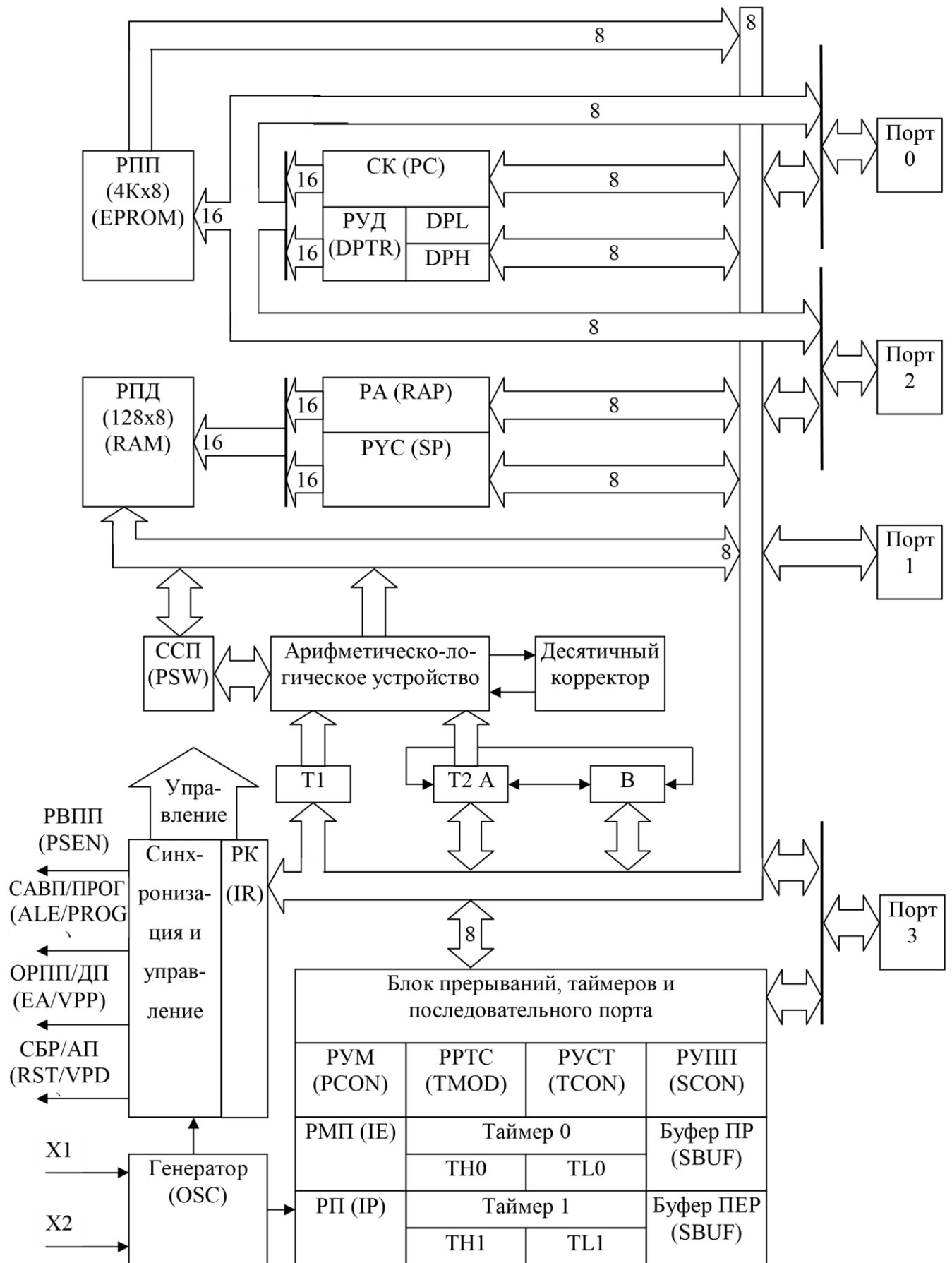


Рисунок 1.4 – Структура МК

7. Порт 0-3 – выходы МК, позволяющие устанавливать "0" или "1" в зависимости от требований выполняемой программы. Порт 3 способен выполнять специальные функции;
8. РП – регистр приоритета;
9. РПМ – регистр маски прерываний;
10. РCON – регистр управления мощностью;
11. TMOD – регистр таймера/счетчика;
12. TCON – регистр статуса таймера;
13. TH0, TL0 – байты нулевого таймера;
14. TH1, TL1 – байты первого таймера;
15. SCON – регистр приемопередатчика;
16. SBHF – буфер приемопередатчика;
17. РПД – регистр памяти данных;
18. РПП – регистр памяти программ;
19. РС – регистр команд;
20. РУД – регистр установки данных;
21. РА – регистр адреса; 22. SP – указатель стека.

При включении питания "Сеть" (тумблер находится на передней панели макета), напряжение +5В поступает на вспомогательные выходы МК, после чего он запускается и автоматически начинает выполнять действия по программе, которая находится у него в памяти программ (РПП).

Написание программы для МК осуществляется на языке Си (в данных лабораторных работах) с использованием текстового редактора Me7. Компиляция программ производится с использованием пакета прикладных программ C51. Программирование МК осуществляется при помощи пакета программ, расположенных в папке ATB2\TM.

Первая строка любой программы на Си программы должна содержать:

```
#include <io51.h>
```

что обеспечивает подключение массивов данных, условных обозначений, переадресаций и т.д. пакета C51 к написанной программе.

В простейшем случае программирование МК сводится к программированию портов МК, а именно к выдаче или приему на выходы

портов логического "0" или логической "1". Для удобства пользования составляющим портов присваивают имена. Смотрите следующий пример:

```
#define CLK    P1.1  /* синхро-импульсы */
#define SDA    P1.2  /* данные в регистры */
#define D1     P1.7  /* 1-й адресный вход дешифратора */
#define D2     P1.6  /* 2-й адресный вход дешифратора */
#define SV     P1.4  /* выбор индикатора аварийной сигнализации */
#define ZV     P3.7  /* звонок */
#define ADC    P1.5  /* выбор АЦП */
```

Использование оператора **define** позволяет в программе для МК вместо обозначения P1.1 использовать CLK, вместо P1.2 – SDA и т.д.

Установка "0" или "1" на порт выполняется следующим образом:

```
ADC = 1;
SDA = 0;
CLK = 0;
RESET = 0; /* сброс регистров */
KEY=0;
Reg=0;
ZV=0;
D1=1;
D2=1;
SV=1;
```

После выполнения такой инициализации на P1.5 установится логическая "1", на P1.2 логический "0" и т.д. Установка на порты МК логических "0" и "1" позволяет выполнять управление периферийными устройствами, последовательную передачу/прием данных и др.

Перед программированием программа для МК, написанная на языке программирования Си, должна быть отредактирована и преобразована в файл с расширением .hex. Для этого файл с текстом программы необходимо сохранить под именем test.c51 и запустить на выполнение программу c.bat, которая находится в папке ATB2\LAB. Если в программе для МК есть ошибки c.bat указывает на них. Ошибки необходимо устранить и повторить запуск c.bat снова. Если ошибок нет c.bat выдает соответствующее сообщение и создает (обновляет) файл test.hex, информация которого должна быть записана в РПП МК при его программировании программатором.

3 ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРА AT89S8252

Для программирования МК используется программатор, находящийся в ЭБ, схема которого представлена на рисунке 1.5.

При программировании МК используется следующий алгоритм.

Полученный файл test.hex необходимо сохранить в папке ATB2\C51 под именем atb.hex. Для "прошивки" этого файла необходимо в среде FAR запустить на выполнение файл _ispatb.bat.

При проведении указанных действий МК будет "прошит" написанной программой, ЭБ автоматически выйдет из режима программирования и произойдет запуск МК на выполнение "прошитой" программы.

4 ЗАДАНИЕ И ПОРЯДОК ЕГО ВЫПОЛНЕНИЯ

4.1 Осуществить программирование микроконтроллера AT89S8252

1. Войти в текстовый редактор Me7.
2. Создать файл lab1.txt с сохранением его в папке ATB2\LAB. Для этого использовать опции **file** и **save as**.
3. Набрать в редакторе программу следующего вида:

```
#include <io51.h>
```

```
#define A8   P2.0  /* светодиод 1 */  
#define A9   P2.1  /* светодиод 2 */  
#define A10  P2.2  /* светодиод 3 */  
#define A11  P2.3  /* светодиод 4 */  
#define A12  P2.4  /* светодиод 5 */  
#define A13  P2.5  /* светодиод 6 */  
#define A14  P2.6  /* светодиод 7 */  
#define A15  P2.7  /* светодиод 8 */
```



```

A8 = 0;
A9 = 0;
A10 = 0;
A11 = 0;
A12=0;
A13=0;
A14=0;
A15=0;
}

/* Подпрограмма инициализации 2*/ void
init2(void)
{
    A8=1;
    A9=1;
    A10=1;
    A11=1;
    A12=1;
    A13=1;
    A14=1;
    A15=1;
}

/* Подпрограмма задержки 25 мкс */
void delay50(void)
{
    unsigned char i = 300;
    do {} while(i--);
}

/* ----- СТАРТ ----- */
/* Основная программа*/

void main(void)
{
    do
    {
        init1();          /* инициализация 1*/
        delay50();         /* задержка*/    init2();
        /* инициализация 2*/    delay50();    /*
задержка*/ while(1);      /* непрерывный
цикл работы */
    }
}

```

4. Сохранить набранную программу, используя опции **file** и **save**.
5. Сохранить набранную программу под именем test.txt.

6. Войти в среду Far и запустить на выполнение программу c.bat.
7. После выполнения программы c.bat выдает сообщения об ошибках (если такие есть) с указанием строки в программе для МК, где эти ошибки обнаружены. Необходимо исправить ошибки в программе для МК и повторить п.6. Повторять п.6 до тех пор пока c.bat не выдаст сообщения, что "ошибок 0", "замечаний 0".
8. После выполнения программы c.bat без ошибок и замечаний автоматически формируется (обновляется) файл test.hex.
9. Загрузить файл test.hex в программатор для дальнейшей прошивки МК. Для этого необходимо выполнить следующие действия:
 - 9.1. Сохранить файл test.hex под именем atb.hex в папке ATB2\C51, используя опции **file** и **save as**;
 - 9.2. Запустить на выполнение файл _ispatb.bat. Данное действие программирует микроконтроллер написанной программой. В случае если присутствует сообщение об ошибках необходимо обратиться к преподавателю.
10. Выполнить запуск программы МК. Убедиться в правильности функционирования МК согласно написанной программе.

4.2 Разработать и реализовать программу световой индикации

Для данного опыта схема соединения МК с периферийной частью представлена на рисунке 1.6. Согласно схеме на рисунке 1.6 каждый светодиод анодом подключен к источнику питания +5В, а катодом к порту МК. Светодиод будет подсвечен в том случае если на соответствующем порту МК будет присутствовать логический "0" (присутствует разность потенциалов на светодиоде) и не будет подсвечен если на соответствующем порту МК будет присутствовать логическая "1" (разность потенциалов на светодиоде отсутствует).

В данном опыте необходимо выполнить следующее:

1. Получить задание согласно вариантам (см. таблицу 1.2).
2. Перевести полученное число в двоичную систему счисления для определения тех светодиодов, которые необходимо подсветить. Логический "0" в числе в двоичной системе счисления соответствует неподсвеченному светодиоду. Логическая "1" соответствует подсвеченному светодиоду.

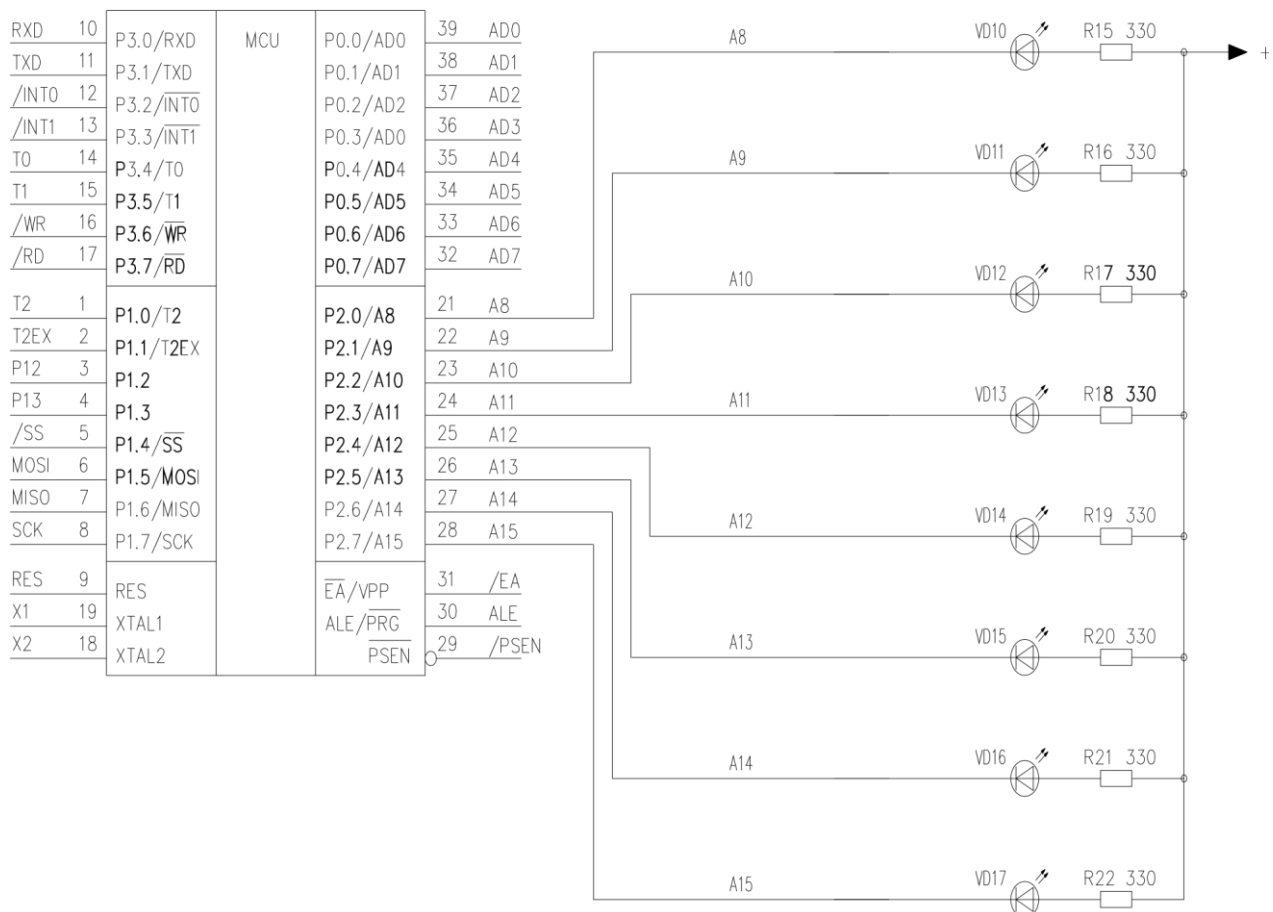


Рисунок 1.6 – Схема соединения светодиодов подсветки с МК

Таблица 1.2 – Варианты заданий к опыту 4.2

Номер варианта	Задание
1	100
2	44
3	26
4	67
5	103
6	52
7	29
8	35

3. Написать программу по аналогии с выполненной программой предыдущего опыта в Me7, где запрограммировать порты МК для подсветки светодиодов согласно варианту.

4. Сохранить набранную программу под именем test.txt.
5. Отладить написанную программу с использованием программы c.bat.
6. Запрограммировать МК программой файла test.hex, выполнив п.9 опыта 4.1.
7. Запустить МК на выполнение "прошитой" программы.

4.3 Разработать и реализовать программу кнопочной станции

Для данного опыта схема соединения МК с периферийной частью представлена на рисунке 1.7. Согласно схеме на рисунке 1.7, если все представленные порты МК запрограммировать в состояние логической "1", светодиоды гореть не будут. Но при нажатом состоянии кнопки на соответствующем светодиоде возникает разность потенциалов и он переходит в подсвеченное состояние.

Выполнить следующее:

1. Написать программу в Me7, где запрограммировать порты МК рисунка 1.6 в состояние логической "1".
2. Выполнить пункты 4 – 7 опыта 4.2.
3. Проверить работу написанной программы. При верном выполнении "прошитой" программы, при нажатии кнопки кнопочной станции подсвечивается светодиод над кнопкой.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Объясните назначение узлов структурной схемы микроконтроллера AT89S8252.
2. Объясните назначение выводов микроконтроллера AT89S8252.
3. Изложите методику компилирования программ с использованием пакета C51.
4. Изложите методику программирования микроконтроллера AT89S8252.

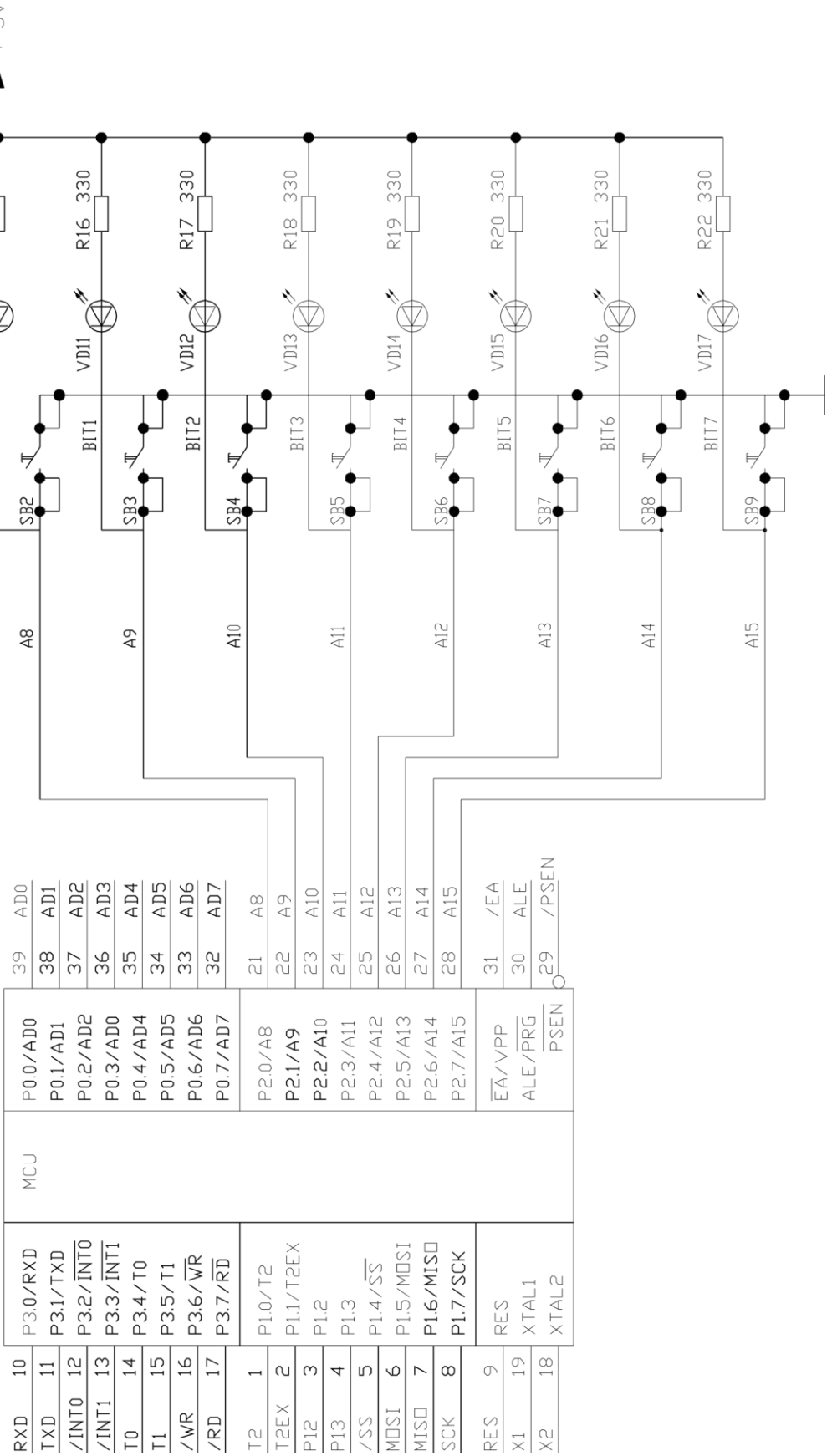


Рисунок 1.7 – Схема соединения кнопочной станции с МК

5. Объясните структуру и принципы построения программ для функционирования цифровых схем с использованием микроконтроллера AT89S8252.

6. Объясните аппаратную часть, которая использовалась при выполнении настоящей лабораторной работы.

7. Изложите задание и порядок выполнения работы.

СПИСОК ЛИТЕРАТУРЫ

1. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах. – М.: Энергоатомиздат, 1990. -224 с.

2. Шевкопляс Б.В. Микропроцессорные структуры. Инженерные решения: Справочник. – М.: Радио и связь, 1990. -512 с.

ЛАБОРАТОРНАЯ РАБОТА №2

ИЗУЧЕНИЕ ПРОЦЕДУР ПРОГРАММИРОВАНИЯ СВЕТОВОЙ ИНДИКАЦИИ

Цель работы: ознакомление с основными узлами схем световой индикации; изучение принципов использования дешифраторов и регистров сдвига в схемах индикации с МК; освоение методики и приобретение практических навыков при программировании МК в схемах световой индикации на светодиодах.

1 СХЕМОТЕХНИКА СИСТЕМЫ ИНДИКАЦИИ

1.1 Соединение МК с регистром сдвига

Соединение микроконтроллер–регистр сдвига позволяет преобразовать последовательный код информации, передающийся с МК, в параллельный. Пример схемы такого соединения представлен на рисунке 2.1. Организация такой передачи возможна с использованием четырех линий:

1. Линия Р – выбор кристалла;

2. Линия CLK – организация синхроимпульсов;
3. Линия RES – организация сброса;
4. Линия SDA – линия передачи информации.

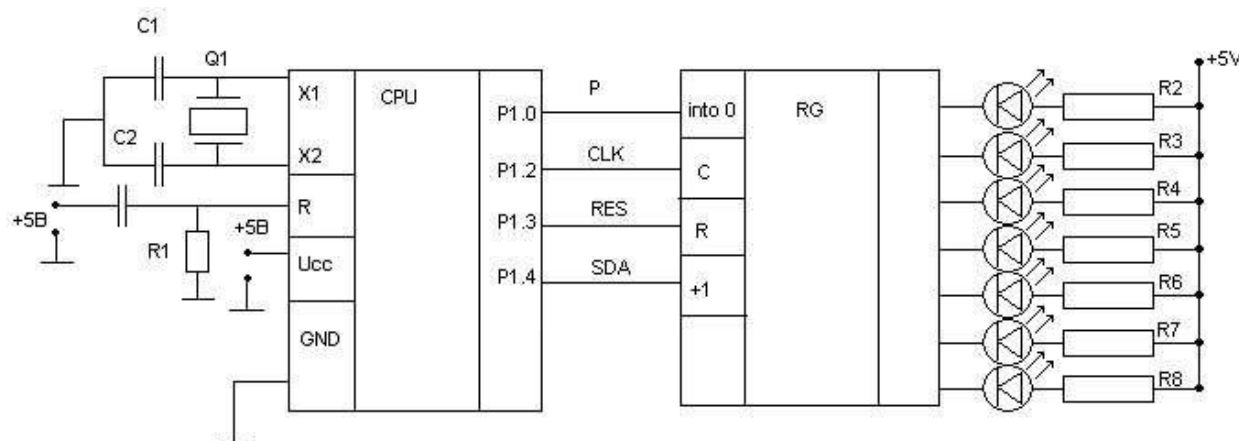


Рисунок 2.1 – Соединение микроконтроллер–регистр сдвига

С помощью подачи сигнала – сброс, единичного логического уровня по RES, МК переводит все выходы регистра в исходное (нулевое или единичное) состояние.

Выбор кристалла – это сигнал (P), который позволяет активизировать соответствующую микросхему для ее нормальной работы. Для активизации регистра на вход **into** регистра необходимо подать сигнал логической «1» по P.

Сигнал CLK программным путем позволяет организовать последовательность релаксационных импульсов для последовательной передачи побитной информации. Сигнал SDA служит для передачи информации, синхронизируемой импульсами CLK. Временная диаграмма (см. рисунок 2.2) сигналов CLK и SDA показывает принцип передачи информации из МК в регистр сдвига.

Установка логических «0» или «1» на выходах соответствующих портов МК осуществляется программным путем. Логические «0» или «1», представляющие собой один бит информации, при передаче на регистр должны быть выставлены на порт SDA в момент присутствия логической «1» на выходе порта CLK МК.

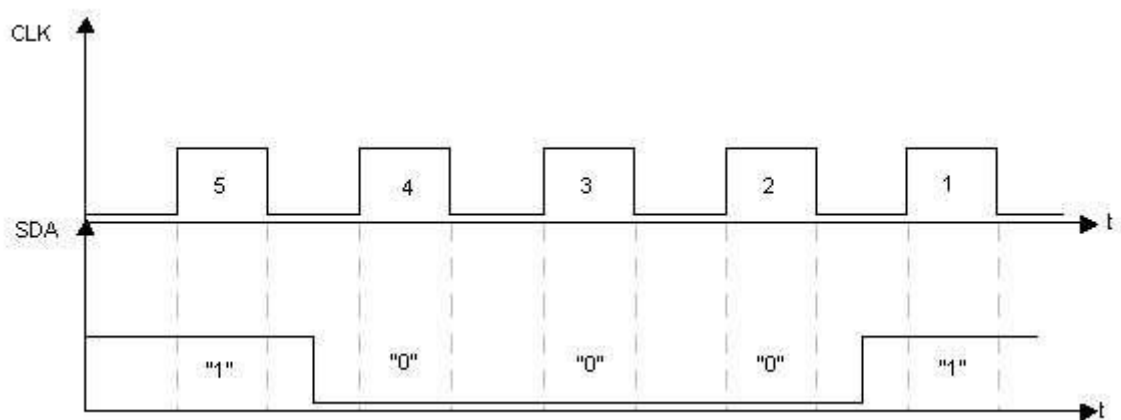


Рисунок 2.2 – Временная диаграмма последовательной передачи данных

1.2 Состав схемы индикации на светодиодах

На рисунке 2.3 представлена схема световой индикации на светодиодах, используемая в данной лабораторной работе. Данная схема состоит из МК, регистра сдвига K1533ИР24, четырех светодиодов типа АЛ307.

2 ПРОГРАММИРОВАНИЕ СХЕМЫ ИНДИКАЦИИ

Аппаратурная часть, при синтезе гибких логических схем на МК, является лишь составляющей полной картины логики функционирования МК. Вторая часть логики функционирования МК представляет программное обеспечение.

Программа для МК, осуществляющая последовательную передачу данных из МК в регистр сдвига для световой индикации должна содержать:

1. Строку подключений пакета данных условных обозначений и переадресаций к написанной программе:

```
#include <io51.h>
```

2. Строки переименований:

```
# define CLK      P1.0
# define SDA      P1.1
# define D1       P2.5
# define D2       P2.6
```



```

/* Подпрограмма инициализации */ void
init(void)
{
    SDA = 0;
    CLK = 0;
    D1=1;
    D2=1;
}

```

4. Подпрограмму задержки:

```

/* Подпрограмма задержки 10 мкс */
void delay10(void)
{
    unsigned char i = 10;
    do {} while(i--); }

```

5. Подпрограмму создания тактовых импульсов:

```

/* Подпрограмма создания тактовых импульсов */
void strob(void)
{
    CLK = 1;
    delay10();
    CLK = 0;
}

```

6. Подпрограмму записи слова в регистр:

```

/* Подпрограмма записи слова в регистр */
void outprog (char prog)
{
    char j,mask;
    mask=0x01;

    for (j=0; j<8; j++)
    {
        SDA = prog & mask ? 1:0;
        strob();
        mask <<= 1;
    }
}

```

7. Основную программу:

```

/* Основная программа*/

void main(void)

```

```

{
    char svindic;
    delay10();      /* задержка */
    svindic=0x99;
    init();         /* инициализация */
    delay10();      /* задержка */
    outprog(svindic); /* запуск подпрограммы записи слова в регистр */
    delay10();      /* задержка */
}

```

В основной программе переменная svindic содержит информацию 99H и определяет какие именно светодиоды должны быть подсвечены.

3 ЗАДАНИЕ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Осуществить программирование световой индикации

1. Изучить схему светодиодной индикации.
2. На основе фрагментов представленных в разделе 2 методических указаний настоящей лабораторной работы написать программу световой индикации на МК.
3. Отладить и запрограммировать МК написанной программой. Для этого использовать алгоритм действий, представленный в настоящих методических указаниях к лабораторной работе №1.
4. Изменить в написанной программе состояние переменной svindic на:

```
svindic=0x77;
```

и выполнить вновь п.2 и п.3 данного опыта.

3.2 Модернизировать световую индикацию

1. Изучить новое задание, в котором необходимо добиться мерцающего режима работы светодиодов схемы светодиодной индикации.
2. В написанную программу первого опыта внести следующие коррективы:

```
/* Основная программа */
```

```

void main(void)
{
    char svindic;

```

```

init();          /* инициализация */
delay10();       /* задержка*/ do
{
    svindic=0x00;
    outprog(svindic); /* запуск подпрограммы записи слова в регистр */
    delay10();        /* задержка*/    delay10();        /* задержка*/
    delay10();        /* задержка*/
    svindic=0xFF;    outprog(svindic); /* запуск подпрограммы записи
слова в регистр */    delay10();        /* задержка*/    delay10();
/* задержка*/    delay10();        /* задержка*/
    while(1);        /* непрерывный цикл работы */
}
}

```

3. Отладить и запрограммировать МК написанной программой.

4. Убедиться в правильности функционирования световой индикации.

3.3 Организовать бегущую световую строку

1. Изучить новое задание, в котором необходимо добиться режима работы светодиодов, когда последовательно в цикле подсвечиваются светодиоды под номерами:

1, 2, 3, 4, 1, 2, 3, 4, 1, 2, ... и т.д.

2. Написать соответствующую программу с использованием фрагментов выполненных программ лабораторных работ №1 и №2.

3. Отладить и запрограммировать МК написанной программой.

4. Убедиться в правильности функционирования световой индикации.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. В чем заключается логика работы регистра сдвига?
2. Какие разновидности регистров Вам известны?
3. Объясните аппаратное обеспечение последовательной передачи данных из МК в регистр сдвига.
4. Объясните программное обеспечение последовательной передачи данных из МК в регистр сдвига.
5. В чем заключаются преимущества алгоритма последовательной передачи данных по сравнению с параллельной?

СПИСОК ЛИТЕРАТУРЫ

1. Гутников В.С. Интегральная электроника в измерительных устройствах. - Л.: Энергия, 1980. - С. 202-206.
2. Алексенко А.Г., Шагурин И.И. Микросхемотехника / Под ред. И.П.Степаненко. - М.: Радио и связь, 1982. - С. 150-174.
3. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах. – М.: Энергоатомиздат, 1990. -224 с.

ЛАБОРАТОРНАЯ РАБОТА №3

ИЗУЧЕНИЕ СРЕДСТВ ПРОГРАММИРОВАНИЯ ЦИФРОВОГО ОТСЧЕТНОГО УСТРОЙСТВА

Цель работы: ознакомление с основными узлами схем цифрового отсчетного устройства (ЦОУ); изучение принципов использования дешифраторов и регистров сдвига в схемах ЦОУ с МК; освоение методики и приобретение практических навыков при программировании МК в схемах ЦОУ на семисегментных индикаторах.

1 СХЕМОТЕХНИКА СХЕМЫ ЦОУ

1.1 Управление цифровым отсчетным устройством Схема управления ЦОУ представлена на рисунке 3.1.

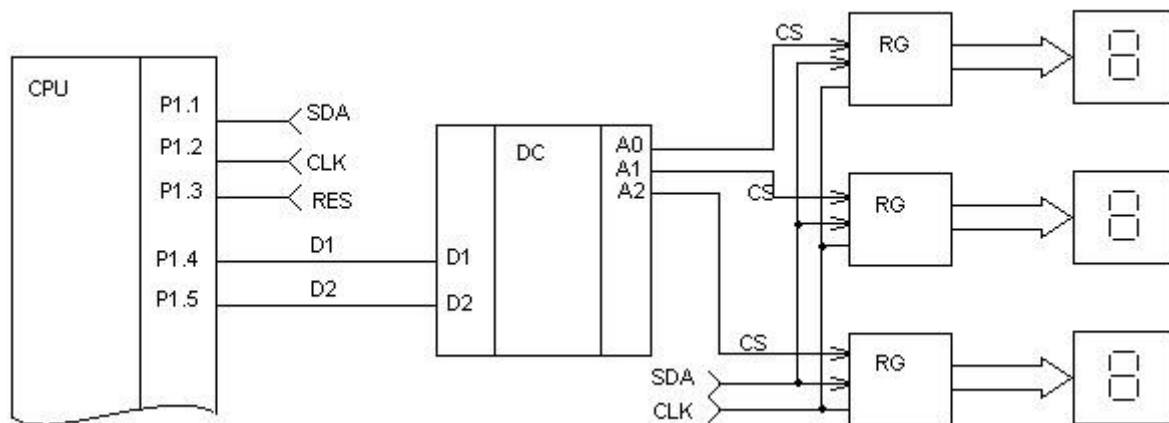


Рисунок 3.1 – Схема включения ЦОУ на семисегментных индикаторах

Схема включения ЦОУ, представленная на рисунке 3.1, предполагает наличие дешифратора для размножения сигналов выбора кристалла CS, регистров сдвига, семисегментных индикаторов.

Дешифратор в представленной схеме получает код с МК по линиям D₁ и D₂ из портов P1.4 и P1.5 МК и согласно таблице истинности (см. таблицу 3.1) выставляет на свои выходы A₀, A₁, A₂ соответствующие сигналы. Логические сигналы A₀, A₁, A₂ являются сигналами активизации CS регистров сдвига. При наличии сигнала CS нулевого уровня на определенный регистр, именно этот регистр активизируется и начинает реагировать на входные сигналы SDA и CLK, поступающие из МК. Регистры, на входе которых сигнал CS имеет уровень логической «1» являются неактивными и сигналы SDA и CLK игнорируют.

Таким образом, для загрузки в регистр сдвига информации размерностью в один байт для ее последующего отображения на ЦОУ, необходимо подать на входы D₁ и D₂ дешифратора соответствующий код (см. таблицу 3.1) для выбора конкретного регистра сдвига и загрузить в выбранный регистр сдвига информацию по линиям SDA и CLK, так как это выполнялось в предыдущей лабораторной работе.

1.2 Индицирующая аппаратура

Сдвиговый регистр K1533ИР24 обеспечивает непосредственное управление светодиодными сегментными индикаторами, например типа SA-56-11GWA. В таблице 3.2 показано каким образом нужно соединять выводы микросхемы регистра и этих индикаторов. В указанной таблице приведены номера выводов индикатора и микросхемы сдвигового регистра, соответствующих одним и тем же сегментам десятичной цифры a-g. Сегмент g старшего разряда индикатора используется для индикации знака “минус”.

Таблица 3.1 – Таблица истинности дешифратора K1533ИД7

Вход					Выход							
S ₁	S ₂	D ₁	D ₂	D ₃	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
X	L	X	X	X	H	H	H	H	H	H	H	H
H	X	X	X	X	H	H	H	H	H	H	H	H

L	L	L	L	L	L	H	H	H	H	H	H	H
L	L	H	L	L	H	L	H	H	H	H	H	H
L	L	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	L	H	H	H	L	H	H	H	H
L	L	L	L	H	H	H	H	H	L	H	H	H
L	L	H	L	H	H	H	H	H	H	L	H	H
L	L	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L

Для индикации посредством сдвигового регистра К1533ИР24 также подходят индикаторы АЛС 321Б, АЛС 324Б и АЛС 338Б, не имеющие никаких существенных отличий от SA-56-11GWA, кроме цоколевки выводов.

Таблица 3.2 – Таблица связи регистра сдвига с индикатором

f e	a g b cd .	Сегмент цифры	Номера выводов, соответствующие различным сегментам цифры	
			Индикатор SA-56- 11GWA	Микросхема К1533ИР24
		a	7	7
		b	6	13
		c	4	6
		d	2	14
		e	1	5
		f	9	15
		g	10	4
		.	5	16

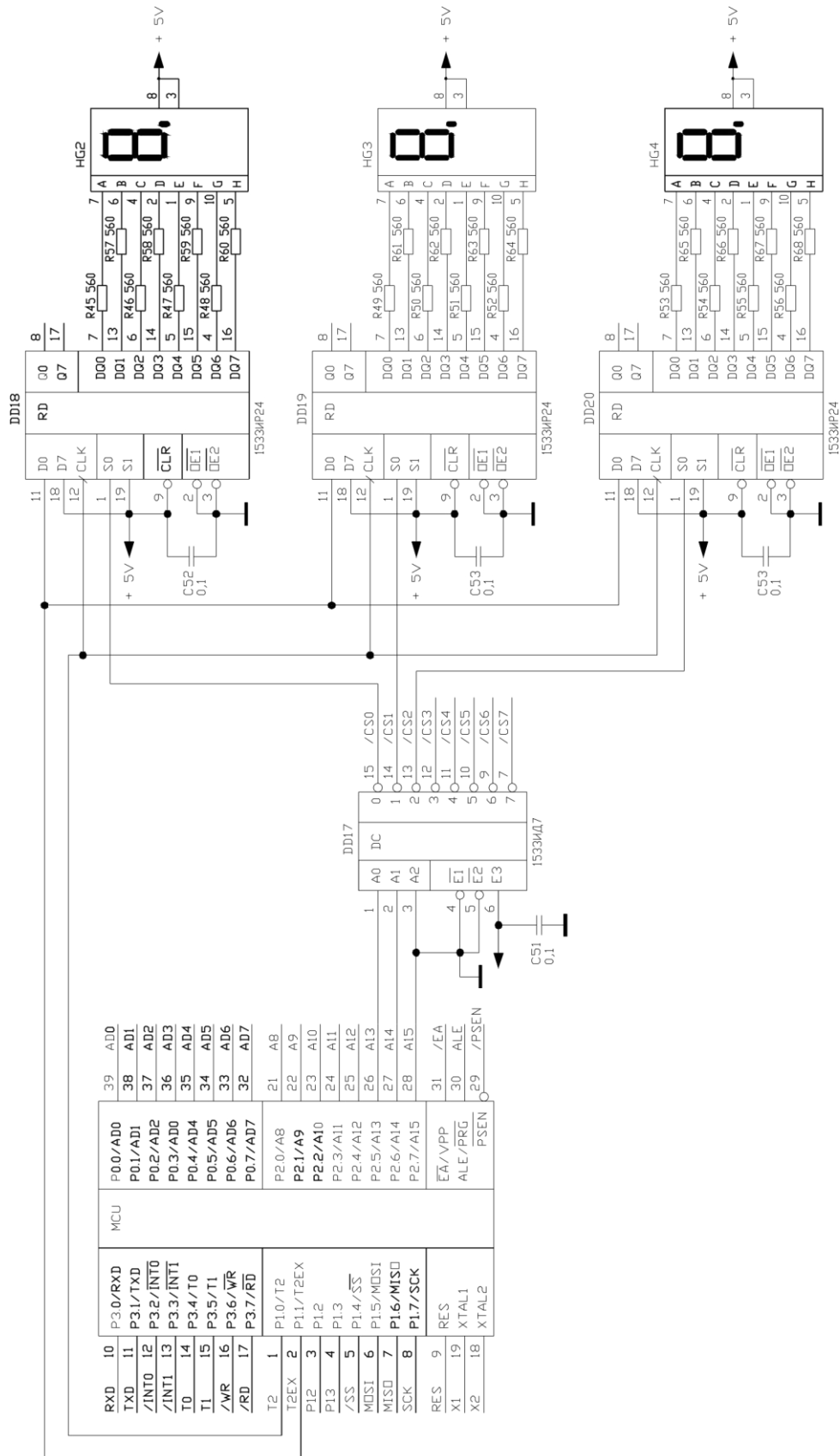


Рисунок 3.2 – Схема цифровой индикации на ЦОУ

1.3 Соединение МК с регистром сдвига

На рисунке 3.2 представлена схема индикации на трехразрядном семисегментном ЦОУ, используемая в данной лабораторной работе. Данная

схема состоит из МК, дешифратора K1533ИД7, регистра сдвига K1533ИР24, трех светоиндикаторов SA-56-11GWA.

2 ПРОГРАММИРОВАНИЕ СХЕМЫ ИНДИКАЦИИ

Структура программы для создания индикации на ЦОУ с использованием МК должна иметь такой же вид как и в программах предыдущих лабораторных работ данного курса и изложенных в настоящих методических указаниях. Т.е. должна содержать:

1. Строку подключений пакета данных условных обозначений и переадресаций к написанной программе,
2. Строки переименований,
3. Подпрограмму инициализации,
4. Подпрограмму задержки,
5. Подпрограмму создания тактовых импульсов, 6. Подпрограмму записи слова в регистр,
7. Основную программу.

Однако, для осуществление работы МК с выводом информации на трехразрядное семисегментное ЦОУ необходимы еще следующие программные узлы:

1. Задание массива данных для цифровой индикации:

```
char als_code[3];    /* массив из 3-х элементов для индикации */
```

2. Задание массива кодов индикаторов:

```
const char NUM_CODE[] = {0x03,0x9F,0x25,0x0D,  
                          0x99,0x49,0x41,0x1F,  
                          0x01,0x09}; /* цифры 0-9 индикатора */
```

3. Подпрограмма индикации результата на ЦОУ:

```
/* Подпрограмма индикации результата */
```

```
void outals(char _code[])  
{  
    char i,j,mask;  
    for (i=0;i<3;i++)  
    {  
        if (i==0)  
            { D1=0; D2=0; }  
    }
```

```

    if (i==1)
        { D1=1; D2=0; }
    if (i==2)
        { D1=0; D2=1; }
mask = 0x01;
delay10();
for(j=0;j<8;j++)
{
    SDA = _code[i] & mask ? 1:0;
    strob();
    mask <<= 1;
}
}
D1=1;
D2=1;
}

```

4. Подпрограмма перевода числа в двоично-десятичную систему:

```

/* Подпрограмма перевода в двоично-десятичную систему */ void
decode(unsigned int number)
{
    als_code[1] = 0xFF;
als_code[0] = 0xFF;
    als_code[2] = NUM_CODE[number % 10];
    number /= 10;
    if(!number) return;
    als_code[1] = NUM_CODE[number % 10];
    number /= 10;
    if(!number) return;
    als_code[0] = NUM_CODE[number % 10];
}

```

5. Основная программа для осуществления цифровой индикации может иметь вид:

```

/* Основная программа проверки индикации */ void
main(void)
{
    int ust;
    init();          /* инициализация */
delay10(); delay10(); delay10();
delay10(); ust=70; /* задание
переменной */
    decode(ust);     /* преобразование переменной в двоично-десятичную систему */
    outals(als_code); /* высвечивание параметра на ЦОУ */ }

```

3 ЗАДАНИЕ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Осуществить программирование цифровой индикации

1. Изучить схему цифровой индикации.
2. На основе фрагментов представленных в разделах 2 методических указаний лабораторных работ №2 и №3 написать программу цифровой индикации на МК.
3. Отладить и запрограммировать МК написанной программой. Для этого использовать алгоритм действий, представленный в настоящих методических указаниях к лабораторной работе №1.
4. Изменить в написанной программе состояние переменной `ust` на:

```
ust=150; /* задание переменной */
```

и выполнить вновь п.3 данного опыта.

3.2 Отладить цифровую индикацию

В предыдущем опыте на индикаторах подсвечивались сегменты, которые не образовывали цифру десятичной системы счисления. Такое явление произошло по причине того, что массив кодов индикаторов оказался неверным.

Массив кодов индикаторов составляется формированием таблицы 3.3.

Первый столбец таблицы 3.3 соответствует цифре, которая должна высветиться на табло. Последний столбец соответствует коду индикатора, который должен быть записан в массив кодов индикатора программы для МК. Промежуточные столбцы определяют подсвеченное или не подсвеченное состояние сегмента в индикаторе по правилу: «1» – не подсвеченное состояние сегмента индикатора, «0» – подсвеченное состояние. Промежуточные столбцы определяют один байт информации в двоичной системе счисления, из которых, построчно, определяется код индикатора соответствующей цифры в шестнадцатиричной системе счисления и представляется в последнем столбце таблицы 3.3.

Таблица 3.3 – Составление кодов цифровых индикаторов

	a	b	c	d	e	f	g	,	
	16	4	15	5	14	6	13	7	
0	0	0	0	0	0	0	1	1	03H
1	1	0	0	1	1	1	1	1	9FH
2	0	0	1	0	0	1	0	1	25H
3	0	0	0	0	1	1	0	1	0DH
4	1	0	0	1	1	0	0	1	99H
5	0	1	0	0	1	0	0	1	49H
6	0	1	0	0	0	0	0	1	41H
7	0	0	0	1	1	1	1	1	1FH
8	0	0	0	0	0	0	0	1	01H
9	0	0	0	0	1	0	0	1	09H

Первые две строки таблицы 3.3 определяют связи выводов регистра сдвига и выводов индикаторов. При этом первая строка таблицы 3.3 соответствует выводу индикатора, вторая строка соответствует выводам регистра сдвига.

1. По аналогии с таблицей 3.3 составить таблицу кодов цифрового индикатора с использованием данных таблицы 3.2.
2. Определить массив кодов индикатора.
3. В написанную программу первого опыта внести откорректированный массив кодов индикатора.
4. Отладить и запрограммировать МК написанной программой.
5. Убедиться в правильности функционирования цифровой индикации.
6. Если массив кодов индикаторов окажется неверным вновь, повторить п.1 – п.5 данного опыта.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Объясните совокупность связей в принципиальной схеме цифровой индикации.
2. Какую функцию выполняет дешифратор?
3. Каким образом дешифратор активизирует работу регистров сдвига для приема информации с выдачей на ЦОУ?

4. Объясните подпрограмму перевода числа в двоично-десятичную систему.
5. Объясните подпрограмму индикации числа на ЦОУ.
6. Каким образом можно изменить аппаратурную часть схемы цифровой индикации для того, чтобы избавиться от дешифратора?

СПИСОК ЛИТЕРАТУРЫ

1. Гутников В. С. Интегральная электроника в измерительных устройствах. - Л.: Энергия, 1980. - С.202-206.
2. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах. – М.: Энергоатомиздат, 1990. -224 с.

ЛАБОРАТОРНАЯ РАБОТА №4

ИЗУЧЕНИЕ ПРОЦЕДУР ПРОГРАММИРОВАНИЯ КЛАВИАТУРЫ

Цель работы: закрепление и углубление лекционного материала по использованию клавиатуры в схемах с МК. Приобретение навыков по программированию клавиатуры в схемах с МК.

1 ОБЩИЕ СВЕДЕНИЯ О ВОЗМОЖНЫХ ВАРИАНТАХ ПОДКЛЮЧЕНИЯ КЛАВИАТУРЫ К МИКРОКОНТРОЛЛЕРУ

1.1 Режим кнопочной станции

Простейшая схема включения кнопочной станции к МК представлена на рисунке 4.1.

Схемотехнически схема выполнена таким образом, что на каждый порт P заводится один вывод ключа (кнопки), а на другой вывод подается общий потенциал. При этом подвешенный в воздухе вывод порта у работающего МК, содержит логическую «1» (разомкнутое состояние ключа). Усиление логической «1» возможно специальной подпиткой вывода

порта от источника питания. Такая подпитка выполнена на порту P1.5 (см. рисунок 4.1).

Если ключ (кнопку) замкнуть, то на соответствующий порт будет подведен общий потенциал, что соответствует логическому «0».

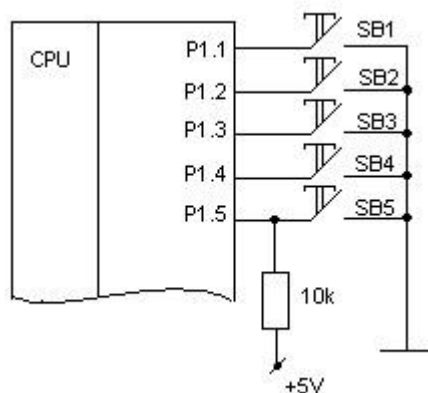


Рисунок 4.1 – Схема подключения кнопочной станции к МК

Для нормального функционирования схемы на рисунке 4.1 программа МК должна содержать цикл опроса состояния портов МК. Смотрите пример фрагмента такой программы:

```
if (!SB1) { decode(tok);    /* выбор тока для индикации */
als_code[1]&=0xFE;
}
if (!SB2) { decode(napr);   /* выбор напряжения для индикации */
}
if (!SB3) { decode(temp);   /* выбор температуры для индикации */
}
if (!SB4) { decode(ust);    /* выбор уставки для индикации */
}
```

В приведенном фрагменте программы при выполнении условия (!SBi) отрабатывается подпрограмма **decode**, если же условие не выполняется – МК отрабатывает следующую строку программы.

Необходимо отметить, что условие (!SBi) будет истинно (условие выполняется), если на этот порт будет подан общий потенциал. В противном случае условие (!SBi) будет ложно (условие не выполняется).

1.2 Режим клавиатуры

На рисунке 4.2 представлена схема соединения МК с клавиатурой.

В режиме клавиатуры выходы ключей (кнопок) объединяются по строкам и столбцам, а объединенные выходы заводятся на порты МК.

Алгоритм опроса клавиатуры в этом случае может быть следующим:

1. Порты P2.0-P2.7 устанавливаются в единичное состояние.
2. На порт P2.4 устанавливается логический «0».
3. В цикле опрашиваются порты P2.0-P2.3. Если на каком-либо порту P2.0-P2.3 присутствует логический «0», это означает, что нажата кнопка первого ряда клавиатуры. При этом:

-если логический «0» присутствует на порту P2.0 то нажата кнопка SB10,

-если «0» на P2.1 – нажата SB14,

-если «0» на P2.2 – нажата SB18,

-если «0» на P2.3 – нажата SB22, (см. рисунок 4.2).

Если на портах P2.0-P2.3 присутствуют логические «1», это говорит об отсутствии замыкания кнопок первого ряда клавиатуры.

4. Далее на порт P2.5 устанавливается логический «0». В цикле опрашиваются порты P2.0-P2.3. Если на каком-либо порту P2.0-P2.3 присутствует логический «0», это означает, что нажата кнопка второго ряда клавиатуры. При этом:

-если логический «0» присутствует на порту P2.0 то нажата кнопка SB11,

-если «0» на P2.1 – нажата SB15,

-если «0» на P2.2 – нажата SB19, -

если «0» на P2.3 – нажата SB23.

Отсутствие логических «0» на портах P2.0-P2.3 указывает на отсутствие замыкания кнопок второго ряда клавиатуры.

5. По аналогии опрашиваются кнопки третьего и четвертого рядов клавиатуры.

Подпрограмма опроса клавиатуры представлена ниже:

```
/* Подпрограмма опроса клавиатуры */
unsigned int key (void) {
    unsigned int tmpkey=0;
    unsigned char keyscan=0x80;
    unsigned char k;    for
(k=0; k<4; k++)      { P2= !
keyscan;    tmpkey <<=
4;    tmpkey |= P2 &
0x0F;    keyscan >>= 1;
delay 10();    delay 10();
}    return ! tmpkey; }
```

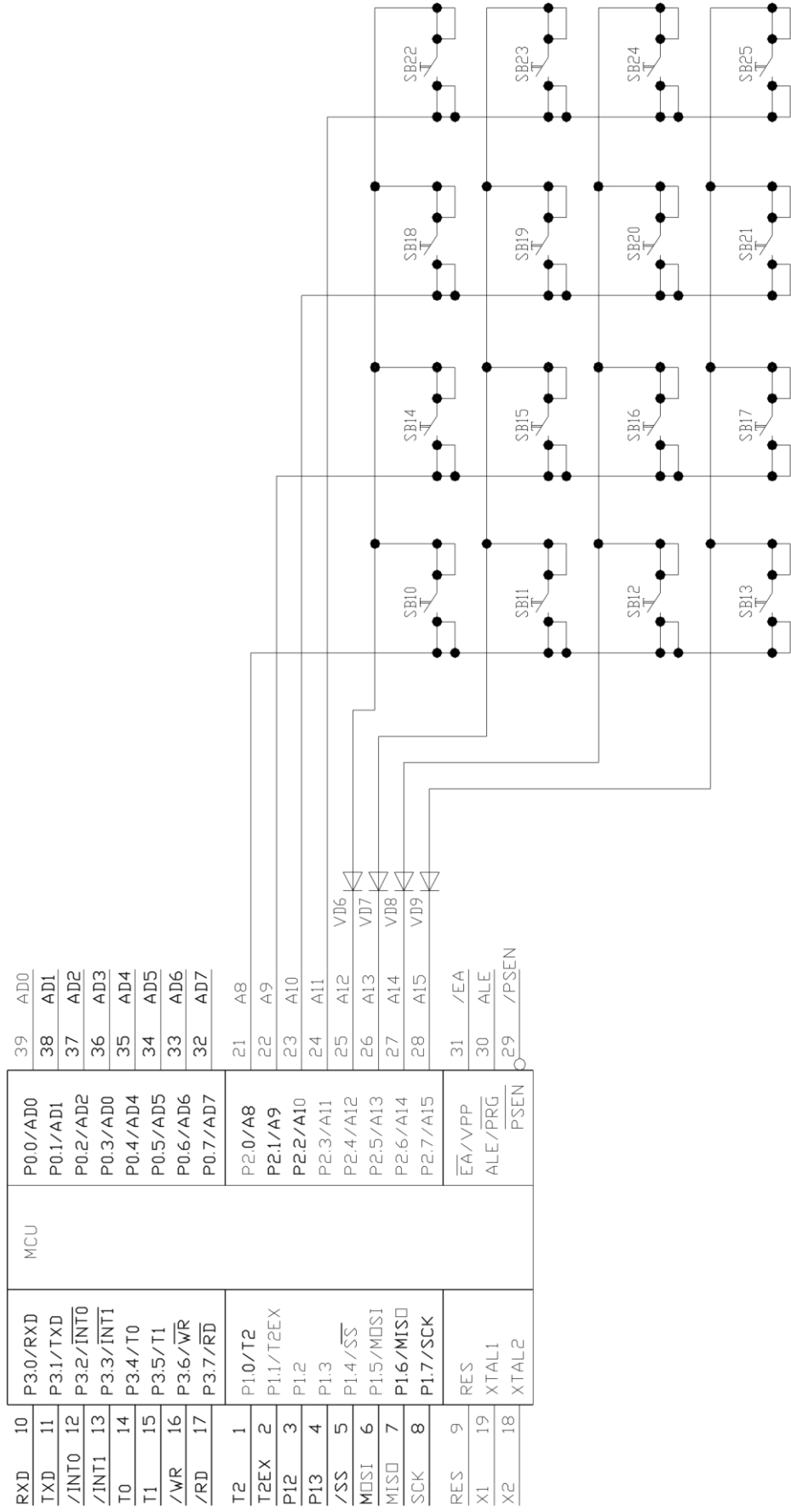


Рисунок 4.2 – Схема подключения клавиатуры к МК

2. ЗАДАНИЕ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

2.1 Осуществить программирование клавиатуры

1. Изучить схему подключения клавиатуры к МК.
2. Принять задание, в котором осуществить подсвечивание соответствующего светодиода при нажатии на кнопку согласно варианту (см. таблицу 4.1):

Таблица 4.1 – Варианты для программирования клавиатуры

№ варианта	№ кнопки
1	SB10
2	SB14
3	SB18
4	SB22
5	SB11
6	SB15

3. Для написания программы МК необходимо использовать следующие фрагменты:

- строку подключений пакета данных условных обозначений и переадресаций к написанной программе,
- строки переименований,
- подпрограмму инициализации,
- подпрограмму задержки,
- подпрограмму создания тактовых импульсов,
- подпрограмму записи слова в регистр, -
- подпрограмму опроса клавиатуры, -основную программу.

4. Для осуществления процедуры подсветки использовать основную программу следующего вида:

```
/* Основная программа*/

void main(void)
{
    char svindic;
    init();          /* инициализация */
    delay10();       /* задержка*/ do
    {
```

```

    svindic= key();
    delay10();      /* задержка*/    outprog(svindic);    /* запуск
подпрограммы записи слова в регистр */
}
while(1);          /* непрерывный цикл работы */
}

```

5. Для данного опыта необходимо в подпрограмме опроса клавиатуры переменную **tmpkey** объявить как **char**.

6. Отладить и запрограммировать МК написанной программой.

7. Убедиться в правильности функционирования опроса клавиатуры.

2.2 Осуществить индикацию номера нажатой кнопки клавиатуры

1. Для написания программы МК необходимо модифицировать программу предыдущего опыта добавлением следующих фрагментов:

- задание массива данных для цифровой индикации,
- задание массива кодов индикаторов,
- подпрограмма индикации результата на ЦОУ,
- подпрограмма перевода числа в двоично-десятичную систему

2. Типовая основная программа для осуществления цифровой индикации может иметь вид:

```

/* Основная программа*/

void main(void)
{
    int ust;
    init();      /* инициализация */
    delay10();   /* задержка*/ do
    {
        ust = key();  decode(ust);    /* преобразование переменной в двоично-
десятичную систему */    delay10();    /* задержка*/    outals(als_code);
/* высвечивание параметра на ЦОУ */    delay10();    /* задержка*/
    }
    while(1);    /* непрерывный цикл работы */
}

```

3. Отладить и запрограммировать МК написанной программой.

4. Убедиться в правильности функционирования программы.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назовите рассмотренные режимы работы панели управления.
2. Какие преимущества и недостатки режима клавиатуры по сравнению с режимом кнопочной станции при подключении панели управления в цепь МК?
3. Объясните совокупность связей в принципиальной схеме подключения клавиатуры к МК.
4. Объясните алгоритм опроса клавиатуры микроконтроллером в схеме на рисунке 4.2.
5. Какие функции может выполнять клавиатура в схемах с МК?

СПИСОК ЛИТЕРАТУРЫ

1. Гутников В. С. Интегральная электроника в измерительных устройствах. - Л.: Энергия, 1980. - С.202-206.
2. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах. – М.: Энергоатомиздат, 1990. -224 с.

ЛАБОРАТОРНАЯ РАБОТА №5

ИССЛЕДОВАНИЕ ИЗМЕРИТЕЛЬНОГО КАНАЛА

Цель работы: закрепление и углубление лекционного материала по конструированию измерительных каналов в схемах с использованием микроконтроллера. Приобретение навыков работы с аналогово-цифровыми преобразователями и аналоговыми коммутаторами с передачей информации последовательным кодом.

1 ВКЛЮЧЕНИЕ АНАЛОГОВОГО КОММУТАТОРА В ЦЕПЬ МИКРОКОНТРОЛЛЕРА

Аналоговый коммутатор представляет собой схемотехническое устройство, предназначенное для пропускания аналогового измерительного сигнала выбранного канала с одного из входов на единый выход, в зависимости от набора управляющих воздействий. Структурно коммутатор состоит из аналоговой и цифровой частей. Аналоговая часть предназначена для пропускания аналогового электрического сигнала и содержит набор ключей (как правило, герконовых). Цифровая часть содержит схему управления, основным элементом которой является дешифратор.

ЭБ лабораторного макета, используемого на лабораторных занятиях, содержит коммутатор К561КП2.

1.1 Схема управления коммутатором

Микроконтроллер способен управлять коммутатором либо параллельным кодом с использованием n -го количества портов (см. рисунок 5.1), либо последовательным кодом с использованием регистра сдвига (см. рисунок 5.2).

Управление микроконтроллером при параллельном соединении происходит за счёт установления нужного уровня управляющего сигнала на портах МК. Программирование коммутатора последовательным кодом фактически сводится к программированию регистра сдвига, выводы которого соединены с управляющими выводами коммутатора (см. соответствующие сведения в лабораторной работе №2 настоящих методических указаний).

Выбор канала для прохождения аналогового сигнала в коммутаторах определяется таблицей истинности. Пример такой таблицы истинности представлен таблицей 5.1.

Таблица 5.1 – Таблица истинности коммутатора

A	B	C	Канал
0	0	0	I
1	0	0	II
0	1	0	III
1	1	0	IV

В таблице 5.1 внесены следующие обозначения:

A, B, C – цифровые (управляющие) входы коммутатора,

CS – выбор кристалла (вывод для активизации микросхемы ("1" – активное состояние)).

Согласно приведенной таблице истинности сигнал первого канала будет выведен на выход коммутатора при комбинации: CS=1; A=0; B=0; C=0. Сигнал второго канала будет выведен на выход коммутатора если: CS=1; A=1; B=0; C=0 (см. таблицу 5.1) и т.д.

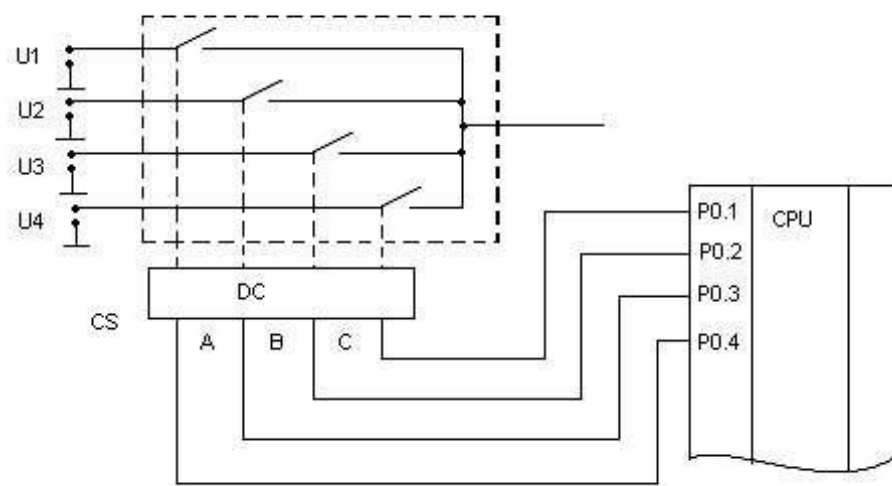


Рисунок 5.1 – Схема включения коммутатора в цепь МК с параллельным управлением

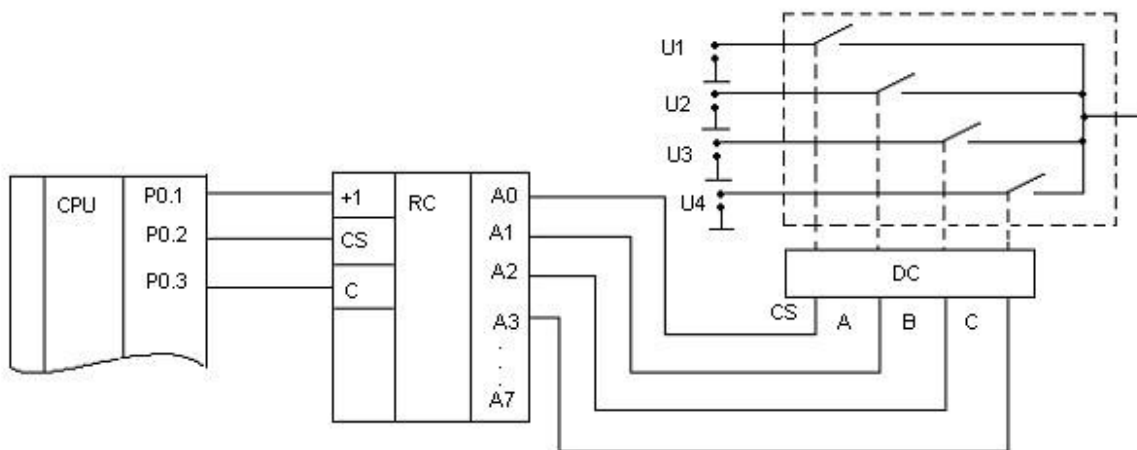


Рисунок 5.2 – Схема включения коммутатора в цепь МК с последовательным управлением

2 ВКЛЮЧЕНИЕ АНАЛОГОВО-ЦИФРОВОГО ПРЕОБРАЗОВАТЕЛЯ В ЦЕПЬ МИКРОКОНТРОЛЛЕРА

2.1 Схема включения АЦП в цепь МК

Схема включения АЦП в микроконтроллерную цепь представлена на рисунке 5.3.

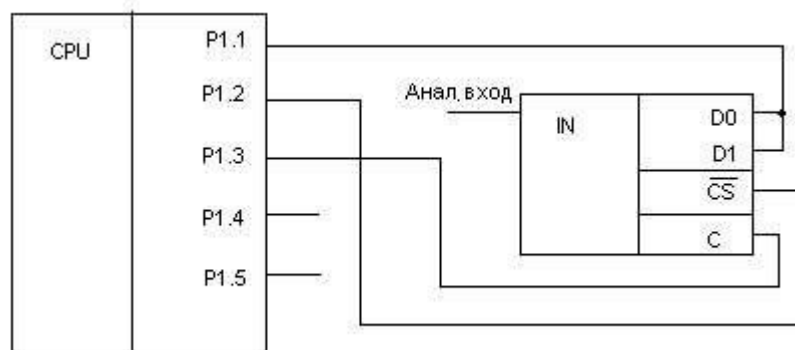


Рисунок 5.3 – Схема включения АЦП в цепь МК

Схема на рисунке 5.3 содержит:

1. Аналоговый вход IN, на который поступает аналоговый измерительный сигнал,
2. Выход D0/D1, предназначенный для передачи оцифрованного сигнала из АЦП на порт P1.1 последовательным кодом,
3. Вывод CS, предназначенный для активизации АЦП,

4. Вывод С, необходимый для синхронизации передачи данных по линии соединения P1.1–D0/D1.

Согласно рисунку 5.3, схема предполагает последовательную передачу данных из АЦП в МК.

2.2 Программное обеспечение соединения АЦП – МК

Программное обеспечение указанного соединения представлено подпрограммой, которая осуществляет запуск АЦП:

```
unsigned int read_acp( void )
{ unsigned char i;
  unsigned int acp_code =0;
  ADC =0;    delay10( );
  SDA =1;    strob( );
  for ( i =0; i<10; i++)
    {strob ( );
     acp_code <<=1;
     acp_code += SDA;
    }
  ADC =1;
  return acp_code;
}
```

2.3 Аппаратурное и методическое обеспечение многовходового измерительного канала

Схема измерительного канала, которая используется в лабораторной работе, представлена на рисунке 5.4 и состоит из коммутатора К561КП2, АЦП AD7810, МК и элементов обвязки.

На входы данного измерительного канала поданы сигналы постоянного напряжения, которые должны быть проиндицированы на ЦОУ.

Для составления программного обеспечения измерительного канала необходимо учитывать, что программа должна содержать:

1. Строку подключений пакета данных условных обозначений и переадресаций к написанной программе:

```
#include <io51.h>
```

2. Строки переименований:

```
# define CLK      P3.5  /* синхро-импульсы */
```

```
# define SDA      P3.4  /* данные в регистры */  
# define ADC      P2.3  /* выбор АЦП */  
# define KOM1     P2.0  /* выбор коммутатора */  
# define KOM2     P2.1  /* выбор коммутатора */  
# define KOM3     P2.2  /* выбор коммутатора */
```

3. Задание массива данных для цифровой индикации:

```
char als_code[3]; /* массив из 3-х элементов для индикации */
```

4. Задание массива кодов индикаторов (взять из отчета по лабораторной работе №3),
5. Подпрограмму задержки,
6. Подпрограмму создания тактовых импульсов,

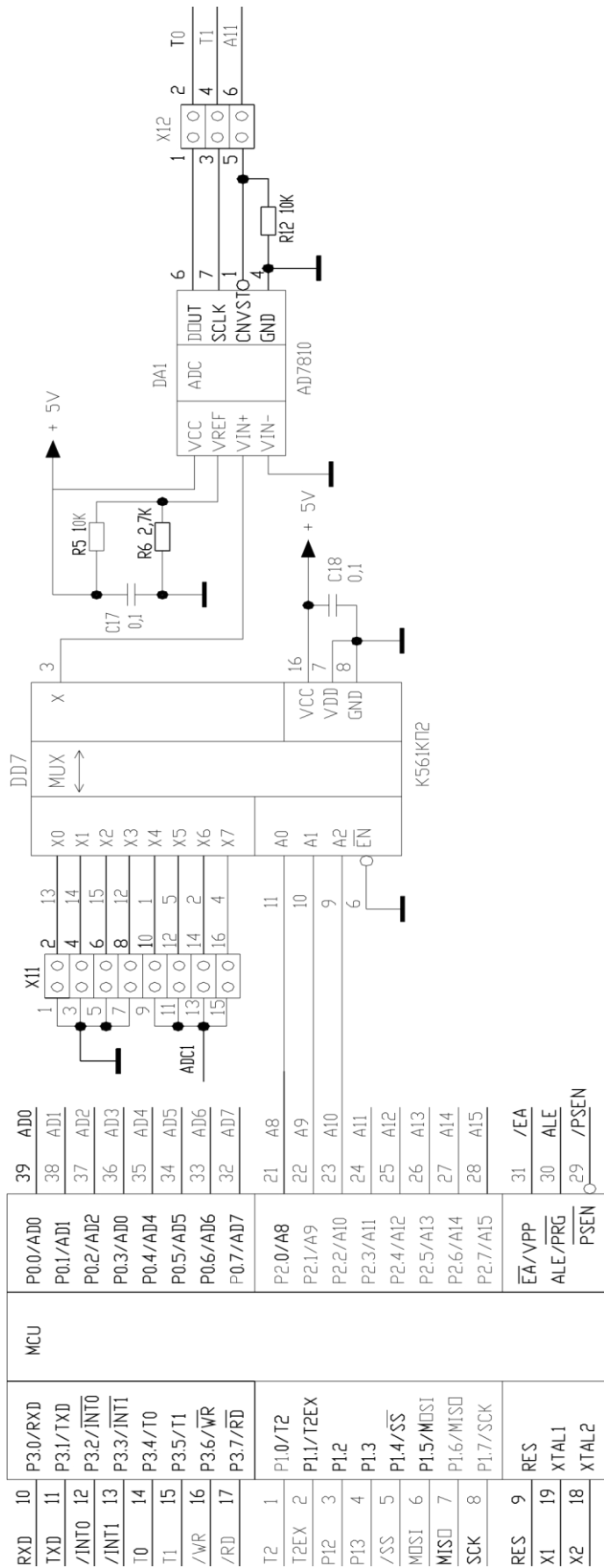


Рисунок 5.4 – Схема измерительного канала

```

/* Подпрограмма инициализации */ void
init(void)
{
    ADC = 1;
    SDA = 0;
    CLK = 0;
    D1=1;
    D2=1;
    КОМ1=1;
    КОМ2=1;
    КОМ3=1;
}

```

8. Подпрограмму индикации результата на ЦОУ,

7. Подпрограмму перевода числа в двоично-десятичную систему,

8. Подпрограмму чтения из АЦП (см. п.2.2 данной лабораторной работы настоящих методических указаний),

6. Основную программу:

```

/* Основная программа */
void main(void)
{
    char svindic;
    int , ust1, ust2, ust3, ust4 ;    init();
/* инициализация */    delay10();

do
{
    КОМ1=1;
    КОМ2=0;
    КОМ3=0; /* выбор первого канала */
delay10();
    ust1=read_asp(2); /* определение аналоговой величины */
    КОМ1=1;
    КОМ2=1;
    КОМ3=1;

    КОМ1=0;
    КОМ2=1;
    КОМ3=0; /* выбор первого канала */    delay10();
ust2=read_asp(2); /* определение аналоговой величины */
    КОМ1=1;
    КОМ2=1;
    КОМ3=1;
    КОМ1=1;
    КОМ2=1;
}

```

```

    КОМ3=0; /* выбор первого канала */
delay10();
    ust3=read_acp(2); /* определение аналоговой величины */
    КОМ1=1;
    КОМ2=1;
    КОМ3=1;

    КОМ1=0;
    КОМ2=0;
    КОМ3=1; /* выбор первого канала */    delay10();
ust4=read_acp(2); /* определение аналоговой величины */
    КОМ1=1;
    КОМ2=1;
    КОМ3=1;

    decode(ust1); /* выбор уставки для индикации */    delay10();
    outals(als_code); /* высвечивание выбранного параметра */
    delay10();    delay10();

    decode(ust2); /* выбор уставки для индикации */    delay10();
    outals(als_code); /* высвечивание выбранного параметра */
    delay10();    delay10();

    decode(ust3); /* выбор уставки для индикации */    delay10();
    outals(als_code); /* высвечивание выбранного параметра */
    delay10();    delay10();

    decode(ust4); /* выбор уставки для индикации */    delay10();
    outals(als_code); /* высвечивание выбранного параметра */
    delay10();    delay10();
}
while(1); /* непрерывный цикл работы */
}

```

3. ЗАДАНИЕ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1 Осуществить циклический опрос измерительных каналов

1. Изучить схему подключения коммутатора и АЦП к МК.
2. Написать программу циклического опроса аналоговых сигналов, поступающих на коммутатор, с последующей индикацией на ЦОУ. Для написания программы МК необходимо использовать структуру программы и фрагменты, описанные в п.2.3 методических указаний к данной лабораторной работе.
3. Отладить и запрограммировать МК написанной программой.
4. Убедиться в правильности функционирования опроса клавиатуры.

5. Отредактировать программу с целью увеличить время индикации оцифрованного сигнала каждого канала.

2.2 Осуществить индикацию аналоговой величины заданного канала

1. В данном опыте необходимо разработать и отладить устройство, у которого аналоговая информация индицируется на ЦОУ только после выбора нажатием кнопки нужного канала. Для написания программы МК необходимо модифицировать основную программу предыдущего опыта:

```
/* Основная программа */
void main(void)
{
    char svindic;    int , ust1, ust2, ust3,
ust4 ;;    init();    /*
инициализация */

    do{
KOM1=1;
    KOM2=0;
    KOM3=0; /* выбор первого канала */
delay10();
    ust1=read_asp(2); /* определение аналоговой величины */
    KOM1=1;
    KOM2=1;
    KOM3=1;

    KOM1=0;
    KOM2=1;
    KOM3=0; /* выбор первого канала */    delay10();
ust2=read_asp(2); /* определение аналоговой величины */
    KOM1=1;
    KOM2=1;
    KOM3=1;

    KOM1=1;
    KOM2=1;
    KOM3=0; /* выбор первого канала */    delay10();
ust3=read_asp(2); /* определение аналоговой величины */
    KOM1=1;
    KOM2=1;
    KOM3=1;

    KOM1=0;
    KOM2=0;
```

```

KOM3=1; /* выбор первого канала */
delay10();
ust4=read_acp(2); /* определение аналоговой величины */
KOM1=1;
KOM2=1;
KOM3=1;

if (!P2.0) { decode(ust1); /* выбор уставки 1 для индикации */
}
if (!P2.1) { decode(ust2); /* выбор уставки 2 для индикации */
}

if (!P2.2) { decode(ust3); /* выбор уставки 3 для индикации */
}

if (!P2.3) { decode(ust4); /* выбор уставки 4 для индикации */
}

outals(als_code); /* высвечивание выбранного параметра */
}
while(1); /* непрерывный цикл работы */
}

```

2. Отладить и запрограммировать МК написанной программой.
3. Убедиться в правильности функционирования программы.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Объясните структуру и принципы управления коммутатором. Какие преимущества и недостатки?
2. Объясните совокупность связей в принципиальной схеме подключения коммутатора к МК.
3. Объясните совокупность связей в принципиальной схеме подключения АЦП к МК.
4. Какие преимущества последовательной передачи информации из АЦП на МК?

СПИСОК ЛИТЕРАТУРЫ

1. Гутников В. С. Интегральная электроника в измерительных устройствах. - Л.: Энергия, 1980. - С.202-206.

2. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах. – М.: Энергоатомиздат, 1990. -224 с.

ЛАБОРАТОРНАЯ РАБОТА №6

РАЗРАБОТКА ИЗМЕРИТЕЛЬНОГО УСТРОЙСТВА

Цель работы: ознакомление с принципами разработки измерительных устройств с использованием микроконтроллеров; приобретение навыков создания принципиальных схем измерительных устройств с использованием блоков включения клавиатуры, светоиндицирующих устройств, ЦОУ, АЦП, регистров сдвига, дешифраторов и т.д.; приобретение практических навыков при программировании МК в схемах измерительных устройств.

1 РАЗРАБОТКА СТРУКТУРНОЙ СХЕМЫ ИЗМЕРИТЕЛЬНОГО УСТРОЙСТВА

Проектирование измерительных устройств, систем контроля и диагностики, систем регулирования и управления для различных производственных процессов с использованием МК основано на последовательной отработке ряда алгоритмов. Как правило, последовательность указанных алгоритмов следующая:

1. Ознакомление с наблюдаемым объектом и изучение технического задания,
2. Разработка структурной схемы устройства,
3. Отработка расчетных и схмотехнических позиций отдельных блоков структурной схемы в виде принципиальных схем,
4. Составление принципиальной схемы,
5. Разработка и отладка программного обеспечения разрабатываемого устройства.

Рассмотрим **пример** построения информационно-измерительной системы теплового технологического агрегата со следующими исходными данными:

необходимо измерить напряжение электрической сети $220\text{В}^{+15^{10}\%}$, фазного тока асинхронного двигателя, температуры объема в диапазоне $0,200^{\circ}\text{C}$, значение температуры уставки $0,200^{\circ}\text{C}$, обеспечить цифровую индикацию выбранного канала и выбор канала для индикации с помощью кнопочной станции, предусмотреть световую и звуковую индикацию.

Для проектируемого устройства разрабатывается структурная схема, представленная на рисунке 6.1.

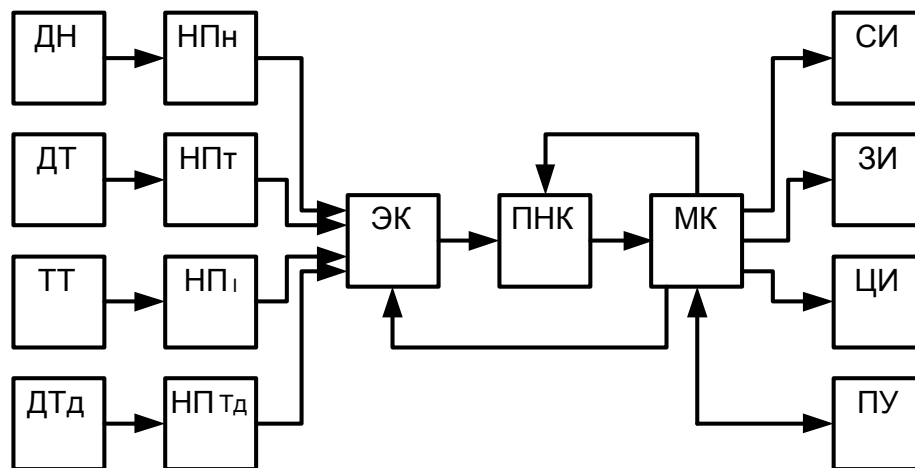


Рисунок 6.1 – Структурная схема информационно-измерительной системы

На схеме рисунка 6.1 обозначены следующие блоки:

ДН – датчик напряжения;

ДТ – датчик температуры;

ТТ - трансформатор тока;

ДТу – датчик температуры уставки;

НПн – нормирующий преобразователь напряжения;

НПт – нормирующий преобразователь температуры;

НПт – нормирующий преобразователь тока;

НПтд – нормирующий преобразователь температуры уставки;

ЭК – электрический ключ (аналоговый коммутатор);

ПНК – преобразователь напряжение в код (АЦП);

МК – микроконтроллер;

СИ – световая индикация;

ЗИ – звуковая индикация;

ЦИ – цифровая индикация;

ПУ – панель управления.

Работа устройства производится следующим образом. Оператор с ПУ задает номер измерительного канала для индикации на ЦОУ. ПУ в данном случае выполняется в виде 4-х кнопок. Если нажата первая кнопка, сигнал первого канала должен поступить на МК, если нажата вторая – сигнал второго канала должен поступить на МК и т.д. МК обрабатывает полученную информацию из кнопочной станции и выставляет на ЭК соответствующий вектор управляющих сигналов. По этому вектору ЭК пропускает на выход сигнал из выбранного канала, который в дальнейшем проходит на вход ПНК. Аналоговый сигнал поступает на ПНК, где преобразовывается в цифровой код. По команде из МК последовательным способом цифровой код из ПНК записывается в МК. МК обрабатывает соответствующий поступивший сигнал, вырабатывает соответствующие управляющие сигналы, которые отображаются в виде СИ, ЗИ и ЦИ.

Рассмотрим схемы каналов, задействованных в рассматриваемом измерительном устройстве. На рисунке 6.2 представлен канал измерения напряжения.

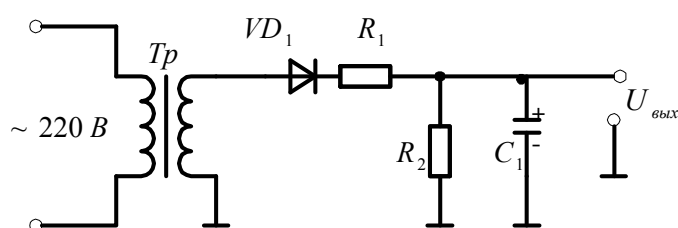


Рисунок 6.2 – Канал измерения напряжения

Схема состоит из сетевого трансформатора Tr , диода VD_1 , постоянных резисторов R_1 , R_2 , конденсатора C_1 . Трансформатор понижает фазное сетевое напряжение 220 В до уровня (7 - 8) В. Переменный сигнал с выхода Tr выпрямляется однополупериодным выпрямителем VD_1 , преобразуется на делителе напряжения R_1 , R_2 , отфильтровывается на пассивном фильтре C_1 и выводится на выход измерительного канала. Подбор номинального значения $U_{вых}$ осуществляется изменением R_1 и R_2 .

Измерительный канал температуры (рисунок 6.3) состоит из медного датчика температуры R_2 , постоянных сопротивлений R_1 , R_3 - R_7 , конденсатора C_1 , ОУ DA_1 .

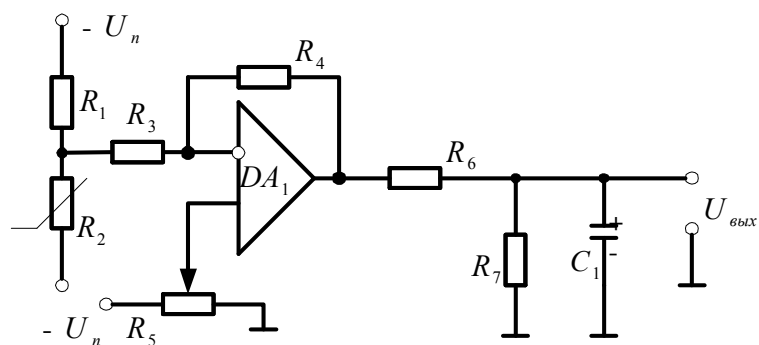


Рисунок 6.3 – Измерительный канал температуры

Изменение температуры в зоне объекта приводит к изменению сопротивления R_2 . Преобразование изменения пассивного параметра – сопротивления в активный – напряжение, производится путем включения R_2 в цепь делителя напряжения. Падение напряжения на R_2 усиливается инвертирующим усилителем на DA_1 . В дальнейшем сигнал поступает на согласовывающий делитель R_6 , R_7 и отфильтровывается пассивным фильтром C_1 . R_5 предназначен для установки нуля.

Измерительный канал тока (рисунок 6.4) состоит из трансформатора тока $ТТ$, диодов VD_1 - VD_4 , постоянных резисторов R_1 - R_3 , ОУ DA_1 , конденсатора C_1 .

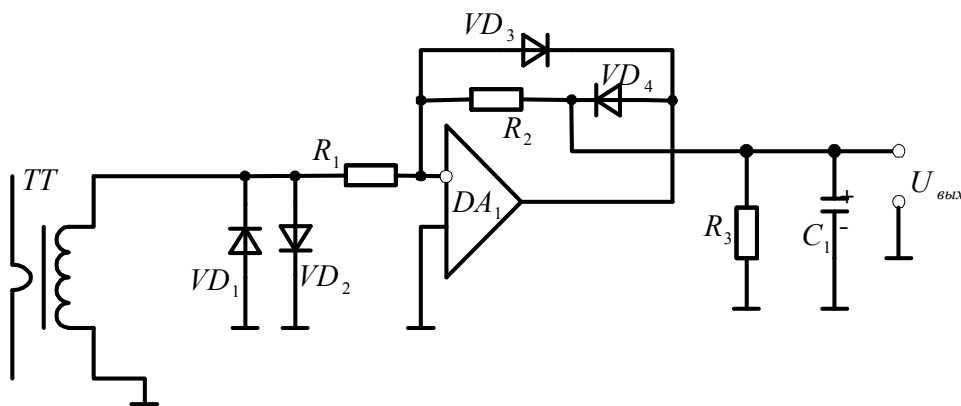


Рисунок 6.4 – Канал измерения тока

$ТТ$ преобразует ток в цепи первичной обмотки в пониженный ток во вторичной обмотке. Режим короткого замыкания $ТТ$ реализуют диоды VD_1 , VD_2 . Сигнал, снимаемый с VD_1 , VD_2 поступает на вход усилителя тока, построенного на DA_1 . Усилитель тока предполагает прямую пропорциональность между падением напряжения на R_3 и значением тока на первичной обмотке. Усилитель тока, кроме преобразования, выполняет

функции двухполупериодного выпрямления за счет диодов VD_3 , VD_4 . Выходной сигнал фильтруется конденсатором C_1 .

Измерительный канал уставки, представленный на рисунке 6.5, состоит из переменного сопротивления R_1 выполненного в виде делителя напряжения.

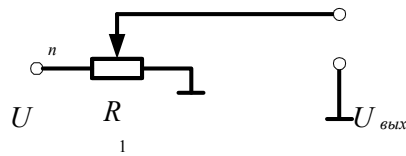


Рисунок 6.5 – Измерительный канал температуры уставки

Электронный ключ (см. рисунок 6.6) состоит из четырех аналоговых ключей и дешифратора, который управляет этими аналоговыми ключами. Если на вход CS дешифратора подан логический ноль электронный ключ включится в работу. А В С дешифратора представляют собой информационные входы необходимые для управления аналоговыми ключами.

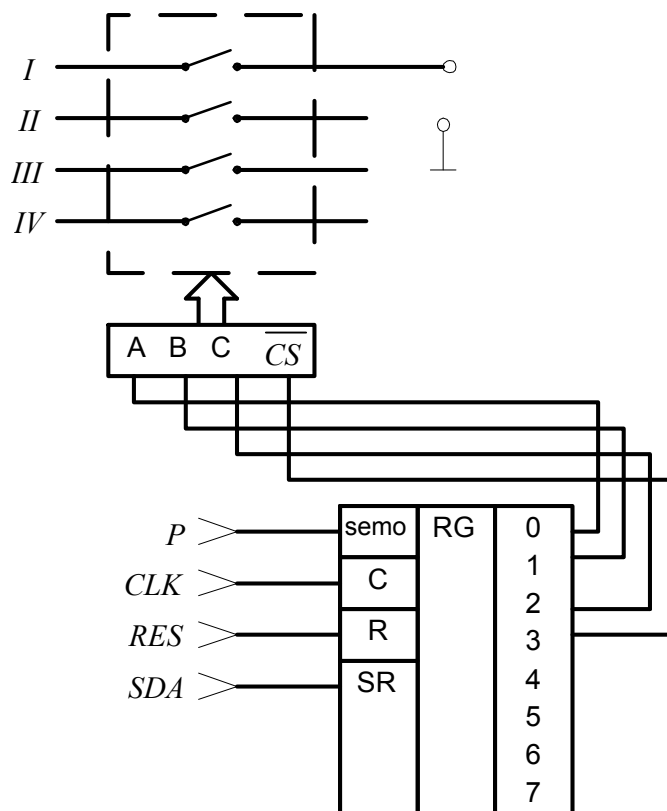


Рисунок 6.6 – Схема электронного ключа

Звуковая индикация (см. рисунок 6.8) содержит резисторы R_1 , R_2 , биполярный транзистор VT работающий в ключевом режиме, звуковой индикатор ZV типа BUZER.

The diagram shows a voltage source U_n connected in series with a resistor R_1 and a load resistor R_2 . The output voltage U_n is measured across R_2 . A load ZV is connected in parallel with R_2 , and a voltage VT is measured across it.

Рисунок 6.8 – Схема звуковой индикации

ЦОУ (см рисунок 6.10) представляет собой совокупность регистров сдвига и цифровых индикаторов типа АЛС. Индикаторы отображают цифровую информацию в виде восьмисегментного кода.

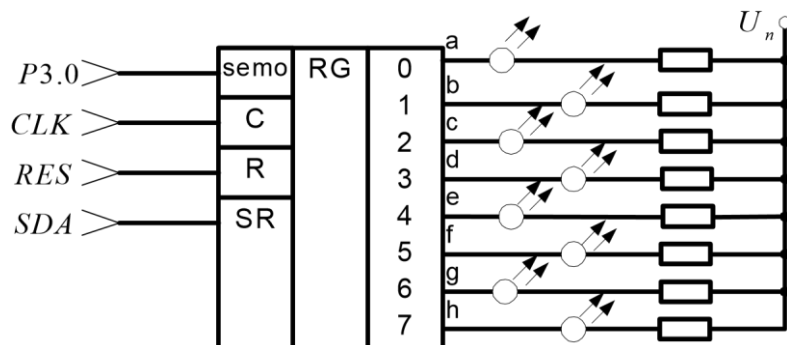


Рисунок 6.9 – Световая индикация

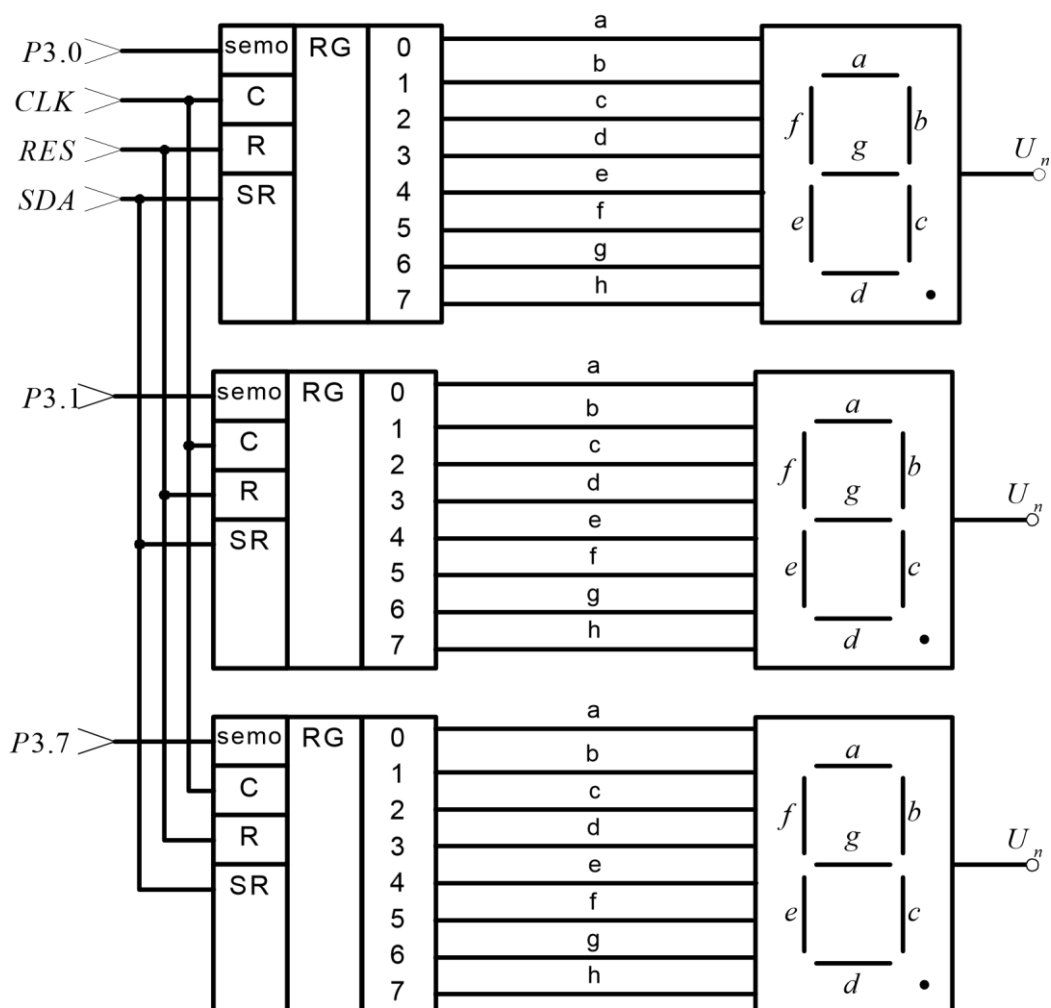


Рисунок 6.10 – Схема ЦОУ

Совокупность представленных выше аппаратурных фрагментов, позволяет сформировать принципиальную схему информационноизмерительной системы на основе разработанной структурной схемы.

Программное обеспечение для данного устройства (написанное на языке программирования Си) имеет следующий вид:

```
/* ПРОГРАММА ФУНКЦИОНИРОВАНИЯ ИИС */

#include <io51.h>

#define CLK    P1.1  /* синхро-импульсы */
#define SDA    P1.2  /* данные в регистры */
#define D1     P1.7  /* 1-й адресный вход дешифратора */
#define D2     P1.6  /* 2-й адресный вход дешифратора */
#define SV     P1.4  /* выбор индикатора аварийной сигнализации */
#define ZV     P3.7  /* звонок */
#define ADC    P1.5  /* выбор АЦП */
#define I      P3.0  /* кнопка выбора тока */
#define U      P3.1  /* кнопка выбора напряжения */
#define T      P3.3  /* кнопка выбора температуры */
#define Tz     P3.4  /* кнопка выбора заданной температуры */
#define Reg    P3.2  /* включение регулятора */
#define KEY    P1.3  /* выбор ключа */
#define RESET  P1.0  /* сброс регистров */

/* Сообщения индикаторов */ const
char NUM_CODE[] = {0x03,0x9F,0x25,0x0D,
                   0x99,0x49,0x41,0x1F,
                   0x01,0x09}; /* цифры 0-9 индикатора */ char

als_code[3]; /* массив из 3-х элементов для индикации */

/* Подпрограмма задержки 10 мкс */
void delay10(void)
{
    unsigned char i = 10;
    do {} while(i--); }

/* Подпрограмма задержки 25 мкс */
void delay50(void)
{
    unsigned char i = 300;
    do {} while(i--);
}
```

```

/* Подпрограмма инициализации */ void
init(void)
{
    ADC = 1;
    SDA = 0;
    CLK = 0;
    RESET = 0; /* сброс регистров */
delay10(); RESET = 1;
    KEY=0;
    Reg=0;
    ZV=0;
    D1=1;
    D2=1;
    SV=1;
    I=1;
    U=1;
    T=1;
    Tz=1;
}

/* Подпрограмма задержки 0,25 сек */
void delay500(void) {
    unsigned int i = 25000;
do {} while(i--); }

/* Подпрограмма задержки 0,5 сек */
void delay1000(void) {
    unsigned int i = 50000;
    do {} while(i--);
}

/* Подпрограмма создания тактовых импульсов */
void strob(void)
{
    CLK = 1;
delay10();
    CLK = 0;
}

/* Подпрограмма индикации результата */
void outals(char _code[])
{
    char i,j,mask;
for (i=0;i<3;i++)
    {
        if (i==0) {
D1=0; D2=0; }
        if (i==1) { D1=1;
D2=0; }
        if (i==2)

```

```

        { D1=0; D2=1; }
mask = 0x80;
delay10();
for(j=0;j<8;j++)
{
    SDA = _code[i] & mask ? 1:0;
    strob();
    mask >>= 1;
}
}
D1=1;
D2=1;
}

```

/* Подпрограмма записи слова в регистр */

```
void outprog (char prog)
```

```

{
    char j,mask;
    mask=0x80;

    for (j=0; j<8; j++)
    {
        SDA = prog & mask ? 1:0;
        strob();
        mask >>= 1;
    }
}

```

/* Подпрограмма перевода в двоично-десятичную систему */

```
void decode(unsigned int number)
```

```

{
    als_code[1] = 0xFF;
als_code[0] = 0xFF;
    als_code[2] = NUM_CODE[number % 10];
    number /= 10;
    if(!number) return;
    als_code[1] = NUM_CODE[number % 10];
    number /= 10;
    if(!number) return;
    als_code[0] = NUM_CODE[number % 10];
}

```

/* Подпрограмма чтения из АЦП */

```
unsigned int read_acp(char izm_code)
```

```

{
    unsigned char i;
    unsigned int acp_code=0;
    ADC=0;

```

```

    delay10();
delay10();
delay10();
delay10(); SDA=1;
    strob();
    SDA=izm_code&0x2?1:0;
    strob();
    SDA=izm_code&0x1?1:0;
    strob();
    SDA=1;
    strob();
    for (i=0;i<10;i++)
    {
strob();
        acp_code<<=1;
    acp_code|=SDA;
    }
    ADC=1;
    return acp_code;
}

/* ----- СТАРТ ----- */
/* Основная программа */
void main(void)
{
    char svindic;    int tok, napr, temp,
ust;    init();      /* инициализация
*/

    delay50();      /* задержка восьмёрок на табло */
svindic=0xFF;
    D1=1; D2=1; SV=1;      /* выбор регистра светодиодов */

    delay500();    outprog(svindic); /* гашение всех
светодиодов */

    SV=0;
delay1000();

    do { KEY=1;    outprog(0x0F); /*
выбор канала тока */    delay50();
KEY=0;    tok=read_acp(2); /*
определение тока */

    KEY=1;    outprog(0x8F); /* выбор канала
напряжения */    delay50();    KEY=0;
    napr=read_acp(2); /* определение напряжения */

```

```

KEY=1;  outprog(0x4F); /* выбор канала
температуры */  delay50();  KEY=0;
temp=read_asp(2); /* определение температуры */

KEY=1;  outprog(0xCF); /* выбор канала уставки
температуры */  delay50();  KEY=0;  ust=read_asp(2);
/* определение уставки температуры */

/* проверка каналов для аварийной индикации */
/* if (temp<70)
    { svindic&=0x7F;} else { svindic|=0x80;};
if (temp>150)
    { svindic&=0xBF;} else { svindic|=0x40;};
if (tok>150)
    { svindic&=0xDF; ZV=1;} else { svindic|=0x20; ZV=0;};
if (napr<175)
    { svindic&=0xEF;} else { svindic|=0x10;};
if (Reg)
    { svindic&=0xF7;} else { svindic|=0x08;};
*/ if (!T)
    { svindic&=0xFD;} else { svindic|=0x04;};
if (!I)
    { svindic&=0xFE;} else { svindic|=0x02;};
if (!U)
    { svindic&=0xEF;} else { svindic|=0x01;};

D1=1; D2=1; SV=1;      /* выбор регистра светодиодов */
delay10();  outprog(svindic); /* высвечивание требуемых
светодиодов */  SV=0;

if (!I) { decode(tok);      /* выбор тока для индикации */
als_code[1]&=0xFE;
    }
if (!U) { decode(napr);      /* выбор напряжения для индикации */
    }

if (!T) { decode(temp);      /* выбор температуры для индикации */
    }

if (!Tz) { decode(ust);      /* выбор уставки для индикации */
    }

outals(als_code);          /* высвечивание выбранного параметра */

if (temp<ust) { Reg=1; }; /* регулятор включен */
if (temp>ust) { Reg=0; }; /* регулятор выключен */
}
while(1); /* непрерывный цикл работы */
}

```

2 ЗАДАНИЕ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Получить техническое задание на проектирование измерительного устройства от преподавателя по таблице 6.1.
2. Разработать структурную схему устройства.
3. Разработать принципиальную схему устройства.
4. Написать программу функционирования МК на языке программирования Си.
5. Продемонстрировать структурную, принципиальную схемы и программу функционирования МК преподавателю.

Таблица 6.1 – Задание на проектирование измерительного устройства

№ вар.	Задание					
	Кол. каналов	Тип каналов	Наличие СИ	Панель упр-я	Индикация на ЦОУ	Кол. разрядов ЦОУ
1	2	I, U	+	–	Циклическая	2
2	3	I, U, T	+	Клавиатура	Выборочная	3
3	4	I, U, T, Tz	–	–	Циклическая	2
4	2	U, T	+	Клавиатура	Выборочная	3
5	3	I, U, Tz	+	Клавиатура	Выборочная	2
6	4	I, U, T, Tz	–	–	Циклическая	3
7	2	U, Tz	+	Клавиатура	Выборочная	2
8	3	U, T, Tz	+	Клавиатура	Выборочная	3

6. Запрограммировать учебный макет написанной программой.
7. Отладить программу. Результат, в виде рабочего образца проектируемого устройства, показать преподавателю.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Объясните принципы построения структурной схемы измерительных устройств с МК.
2. Объясните принципы формирования принципиальной схемы измерительных устройств с МК.
3. Объясните аппаратное обеспечение последовательной передачи данных из МК в регистр сдвига.

4. Объясните принципы формирования программного обеспечения измерительных устройств с МК.

5. В чем заключаются преимущества измерительных устройств с МК по сравнению с цифровыми устройствами, выполненными на «жесткой» логике?

СПИСОК ЛИТЕРАТУРЫ

1. Гутников В.С. Интегральная электроника в измерительных устройствах. - Л.: Энергия, 1980. - С. 202-206.

2. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристальных микроконтроллерах. – М.: Энергоатомиздат, 1990. -224 с.