

Учебно-отладочный стенд
EV8031/AVR (V3.2)

***Методические указания к лабораторным
работам.***

Содержание

| | |
|---|----|
| Введение | 4 |
| Работа со стендом | 10 |
| Пример загрузки программы в стенд | 11 |
| Список условных сокращений | 12 |
| Лабораторная работа №1. Изучение стенда, команд однокристальной ЭВМ семейства MCS-51 | 14 |
| Лабораторная работа №2. Способы построения схем отображения информации | 17 |
| Лабораторная работа №3. Система прерываний. Опрос дискретных датчиков | 26 |
| Лабораторная работа №4. Цифроаналоговое преобразование | 33 |
| Лабораторная работа №5. Аналого-цифровое преобразование | 39 |
| Лабораторная работа №6. Обработка частотных и временных сигналов | 45 |
| Лабораторная работа №7. Изучение интерфейса I2C | 53 |
| Лабораторная работа №8. Программирование последовательного порта ОЭВМ | 59 |
| Перечень литературы | 60 |

Введение

Стенд представляет собой микропроцессорный контроллер, оснащенный памятью программ, памятью данных и разнообразными периферийными устройствами. Он позволяет отлаживать программы, написанные на языках Си и Ассемблер.

Загрузка программы производится с персонального компьютера по последовательному порту RS-232C. Подключение напряжения питания стенда (+5Вольт) и связь с персональным компьютером осуществляется при помощи одного универсального кабеля связи с компьютером (DB9+DB25+Питание).

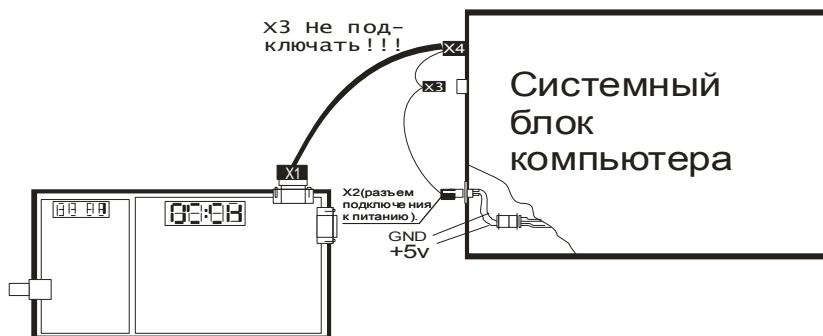
Схема подключения УОС “EV8031” к ПК. Питание стенда.

Связь учебно-отладочного стенда “EV8031” с ПК осуществляется через СОМ-порт. В комплекте со стендом поставляется универсальный соединительный кабель с двумя разъемами (25 и 9 pin) для соединения с одним из СОМ -портов. Кроме того, есть разъем подключения к компьютерному блоку питания и разъем для подключения стенда. Разъем подключения питания стенда удобно крепиться на задней панели системного блока компьютера и поэтому не возникает необходимости постоянно снимать крышку системного блока.

Внимание!!! Для исправной и правильной работы стенда, его необходимо подключать **только с одним** com-портом ПК.

Не рекомендуется отсоединять стенд от ПК при включенном питании. То же самое касается отсоединения платы расширения от стенда.

Схема подключения стенда EV8031/AVR к ПК+питание.



Вариант включения 1.



Вариант включения 2.

Структурная схема стенда представлена на рисунке 1.

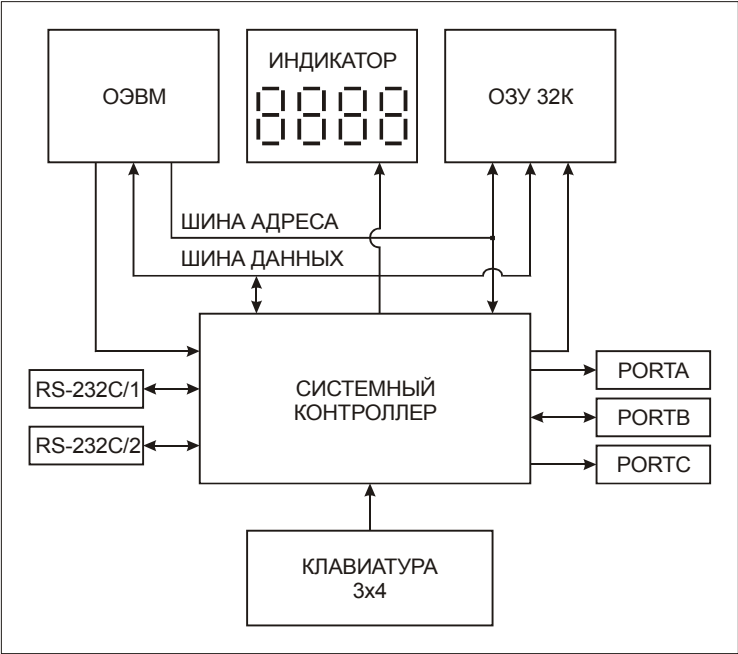


Рисунок 1. Структурная схема стенда

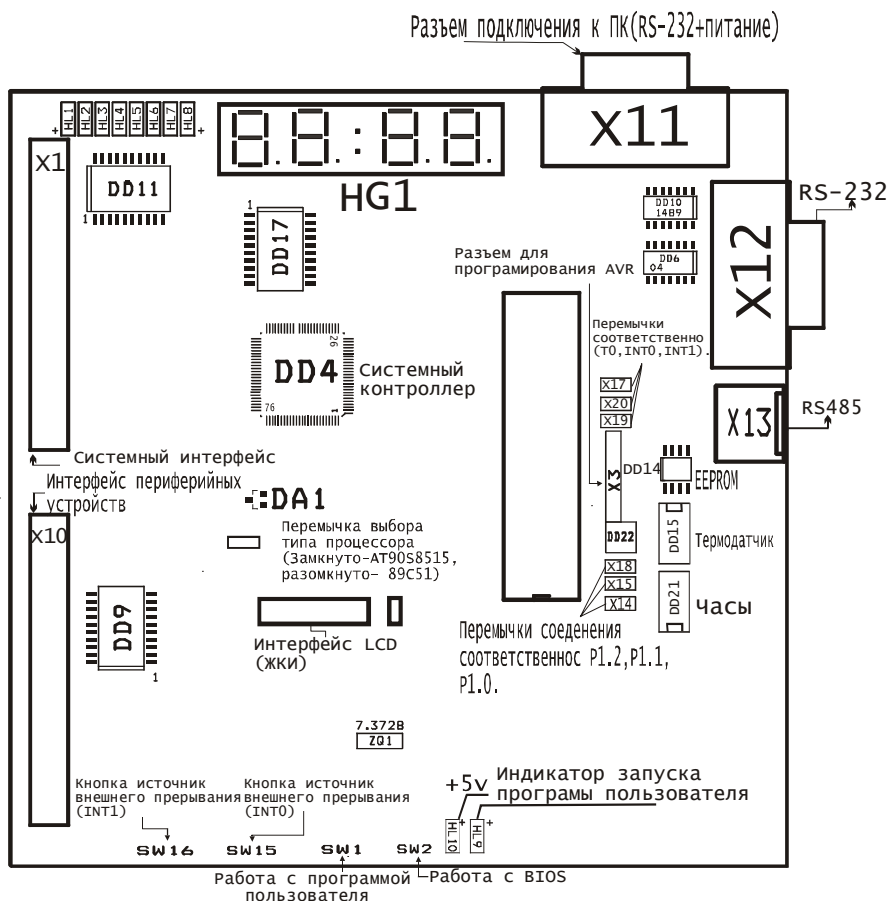
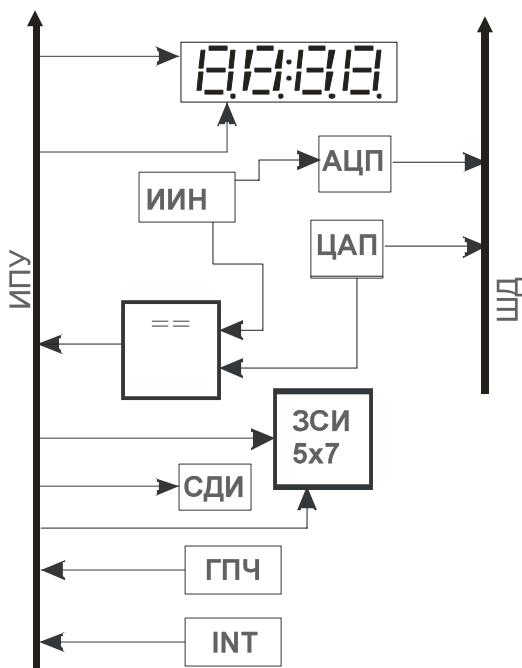


Рисунок 2. Схема расположения элементов стенда

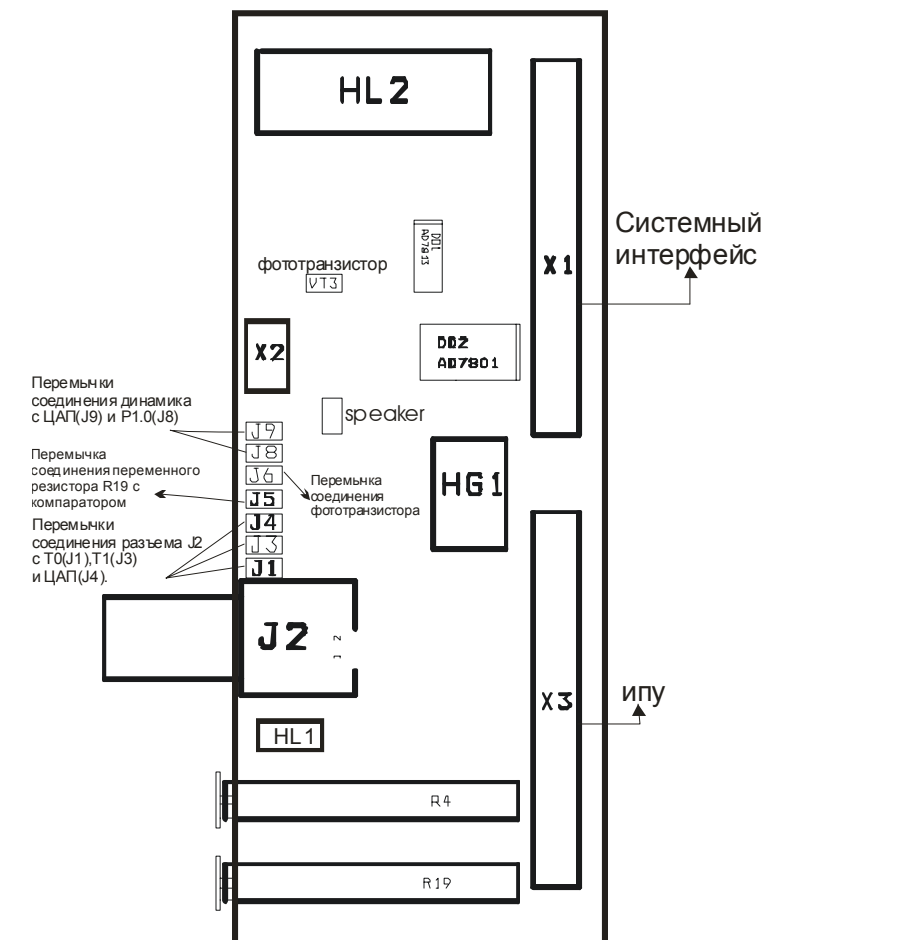
Структурная схема платы расширения представлена на рисунке 3. Плата содержит набор устройств, которые позволяют освоить основные приемы программирования. Курс лабораторных работ состоит из девяти работ, к каждой работе прилагается упрощенная функциональная схема стенда. Упрощенная схема содержит только те элементы, которые необходимы для проведения данной лабораторной работы (рекомендуется также смотреть общую принципиальную схему).

Расположение основных органов управления, и микросхем на плате приведены на рисунках 2 и 4.



8888 – 4 разрядная динамическая индикация;
 ИПУ - Интерфейс периферийных устройств;
 ЦАП - Цифроаналоговый преобразователь;
 СДИ - Светодиодные индикаторы;
 ЗСИ – Знакосинтезирующий индикатор 5x7;
 ГПЧ – Генератор с изменяемой частотой генерации;
 INT – Кнопки запроса прерывания;
 ИИН – источник измеряемого напряжения;
 ШД – шина данных.

Рисунок 3. Структурная схема платы расширения



HG1 – знаковосинтезирующий индикатор 5x7;
 HL2 – 4-х разрядная динамическая индикация;
 J2 – разъём подключения внешних контрольно-измерительных приборов.
 R4 – переменный резистор, источник входного сигнала для АЦП;
 R19 – переменный резистор, изменяет частоту генерации переменного генератора;
 HL1 – индикатор состояния выхода компаратора.

Рисунок 4. Схема расположения элементов платы расширения

Работа со стендом

В этом разделе рассмотрен один из вариантов загрузки программы в стенд (см. приложение №1).

1. На персональном компьютере загрузить текстовый редактор.
2. В текстовом редакторе набрать текст программы в мнемокодах языка Ассемблер (языка C) для 8051.
3. Сохранить набранный файл с расширением *.ASM (*.C).
4. Откомпилировать набранную программу соответствующими средствами.
5. Возможные ошибки в программе можно просмотреть в одноименном файле с расширением *.LST
6. После устранения всех ошибок, данные файла с расширением *.HEX программой EVAL32.EXE необходимо перенести в стенд. Вывод на экран подсказки о параметрах программы EVAL32.EXE, осуществляется запуском EVAL32.EXE.
7. При передаче данных с персонального компьютера в стенд на экране монитора отображаются передаваемые данные. Эти же данные отображаются на индикаторе стенда. Горит светодиод HL9.
8. Остановка, загруженной программы и переход в режим ожидания на прием данных с персонального компьютера возможно нажатием кнопки SW2. При этом гаснет светодиод HL9.
9. Запись новой программы возможна в любой момент времени работы загруженной программы.

Пример загрузки программы в стенд

1. На персональном компьютере загрузить текстовый редактор.
2. Содержимое CD переписать на диск C
3. В текстовом редакторе набрать текст программы в мнемосодах языка Ассемблер для 8051:

```
ORG 0
Start:
mov DPTR, #0A000h      ; занести в регистр DPTR
                        ; адрес индикации
mov A, #28h            ; занести в регистр А данные для отображения
movx @DPTR, A          ; вынести на индикацию, адрес которой
                        ; находится в регистре DPTR число 28
jmp Start              ; переход на начало
END
```

4. Сохранить набранный файл с расширением *.ASM. (Например IND.ASM)
5. Откомпилировать набранную программу возможными средствами. (... \ASM51.EXE IND.ASM)
6. Данные файла с расширением *.HEX перенести в стенд. (... \EVAL32.EXE -hs -com 2 9600 IND.HEX).
 - ✓ *Справку о параметрах программы загрузки данных в стенд из ПК можно просмотреть, запустив программу EVAL32.EXE без параметров.*

Список условных сокращений

| <i>Сокр. Назв.</i> | <i>Описание</i> | <i>Обознач.</i> |
|--------------------|--|-------------------------|
| ОЭВМ | Однокристалльная ЭВМ | |
| Т/С | Таймер счетчик ОЭВМ | |
| ПОС | Программно отладочные средства | |
| ИМС | Интегральная микросхема | |
| ШД | Шина данных | |
| ША | Шина адреса | |
| ДД | Двоично-десятичное число | |
| ПК | Персональный компьютер | |
| R0-R7, A, B | Регистры общего назначения ОЭВМ | |
| XXH | Шестнадцатеричное число | |
| 0X | Число в котором старшая тетрада – "0", младшая любое число от нуля до шестнадцатеричного F | |
| X0 | Число в котором старшая тетрада любое число от нуля до шестнадцатеричного F, младшая – "0" | |
| HL1-HL8 | Индикаторы | HL1, HL2...HL8 |
| С_Инд. | Элемент статической индикации | HG1.0 HG1.1 HG1.2 HG1.3 |
| Д_Инд. | Элемент динамической индикации | HL2 |
| ЗС_Инд. | Знакосинтезирующая индикация | HG1 |
| ГФЧ | Генератор фиксированной частоты | DD18-1, 18-2, 18-3 |
| ГПЧ | Генератор переменной частоты | DD1.1 |
| КЛ | Клавиатура 3*4 | SW3-SW14 |
| КН1, КН2 | Кнопка 1, 2 | SW15, SW16 |
| 485 | 485 интерфейс | DD11 |

Таблица 1 Карта портов ввода/вывода стенда

| Адрес | Тип цикла | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | Имя |
|------------------------------|--------------|------------------------------|------|------|------|------|-------|------|------|------------|
| Порты периферийных устройств | | | | | | | | | | |
| 8xx0 | Запись | [Порт А] | | | | | | | | PA_REG |
| 8xx1 | Запись | [Порт В] | | | | | | | | PB_REG |
| 8xx2 | Запись | [Порт С] | | | | | | | | PC_REG |
| 8xx3 | Запись | x | x | x | x | x | TRISC | x | x | TRIS |
| ЖКИ | | | | | | | | | | |
| 8xx4 | Запись | Регистр команд ЖК индикатора | | | | | | | | LCD_CMD |
| 8xx5 | Запись | Регистр данных ЖК индикатора | | | | | | | | LCD_DATA |
| Последовательный порт | | | | | | | | | | |
| 9xxx | Чтение | CTS | DSR | DCD | RI | KL3 | KL2 | KL1 | KL0 | US_REG |
| Cxx0 | Запись | x | x | X | x | DTR | RTS | CFG1 | CFG0 | UC_REG |
| Индикатор и светодиоды | | | | | | | | | | |
| Axx0 | Запись | [Регистр индикатора 0] | | | | | | | | DISPLAY[0] |
| Axx1 | Запись | [Регистр индикатора 1] | | | | | | | | DISPLAY[1] |
| Axx2 | Запись | <зарезервировано> | | | | | | | | DISPLAY[2] |
| Axx3 | Запись | <зарезервировано> | | | | | | | | DISPLAY[3] |
| Axx4 | Запись | DP3 | DP2 | DP1 | DP0 | BL3 | BL2 | BL1 | BL0 | DC_REG |
| Axx5 | Запись | <зарезервировано> | | | | | | | | EDC_REG |
| Axx6 | Запись | LED7 | LED6 | LED5 | LED4 | LED3 | LED2 | LED1 | LED0 | LED_REG |
| Управление работой | | | | | | | | | | |
| Axx7 | Запись | x | x | X | x | x | x | x | RUN | SYS_CTL |
| | | | | | | | | | | |
| Совместимые регистры | | | | | | | | | | |
| Vxx0 | Запись | [Регистр индикатора 1] | | | | | | | | DISPLAYB |
| | | | | | | | | | | |

Лабораторные работы №1-№8

Лабораторная работа №1. Изучение стенда, команд однокристальной ЭВМ семейства MCS-51

Тема: Изучение стенда, команд однокристальной ЭВМ семейства MCS-51.

Цель работы: Изучение функциональных возможностей учебно-отладочного стенда, внутренней структуры и системы команд ОЭВМ семейства MCS-51.

Учебная задача: Изучение команд пересылок, арифметических, логических команд, команд переходов. Взаимодействие внутренних узлов ОЭВМ.

Порядок выполнения лабораторной работы:

1) Изучить структурную схему стенда, распределение памяти, назначении узлов. Изучить структуру ОЭВМ семейства MCS-51. Изучить синтаксис команд пересылки, арифметических команд, команд переходов.

2) Разработать алгоритм для выполнения индивидуального задания до начала лабораторного занятия (таблица задания №1).

3) Разработать программу для выполнения индивидуального задания до начала лабораторного занятия.

4) Ввести программу индивидуального задания на ПК.

5) Изучить программно отладочные средства (ПОС) для MCS-51.

6) С помощью ПОС проанализировать выполнение индивидуальной программы.

7) Загрузить программу в стенд ОЭВМ. Убедиться в правильном выполнении индивидуального задания, при отрицательном результате осуществить изменение алгоритма либо программы. Повторить загрузку программы в стенд ОЭВМ

8) Распечатать листинг правильно работающей программы.

9) Ответить на контрольные вопросы преподавателя

Контрольные вопросы:

1. Время выполнения команд (понятие такта, машинного цикла)
2. Типы команд. Формат команд. Команды прямой и косвенной адресации.
3. Команды сдвига. Арифметические команды. Логические команды.
4. Регистр признаков, команды вызывающие изменение регистра признаков
5. Команды работы со стеком, последовательность действий (команд) при работе со стеком
6. Назначение внутренних узлов ОЭВМ
7. Назначение и работа с внутренней памятью данных ОЭВМ
8. Система прерывания ОЭВМ. Назначение портов ОЭВМ
9. Физические характеристики выходных сигналов ОЭВМ

Краткие теоретические сведения Приложение №4

Пример выполнения лабораторной работы №1

Содержимое регистров R1 и R4 сложить и отобразить на индикаторе

```

ORG 0
mov R1, #04h      ;записать в R1 число 04
mov R4, #30h      ;установить в R4 число 30
mov DPTR, #0B000h ;установить в DPTR адрес Инд. DD17, DD18
mov A, R1          ;записать в A значение R1
add A, R4          ;сложить значение в A и R4 результат суммы в A
movx @DPTR, A      ;засветить на Инд. DD17, DD18 число хранимое в A
mov A, #00h        ;обнулить A
Con: jmp Con       ;переход на заикливание программы
END

```

Варианты индивидуальных заданий

Таблица 2 Таблица заданий к л.р. №1

| № | Текст индивидуального задания |
|----|---|
| 1 | Занести в регистр R4 двоично-десятичное число 0X, в регистр R6 двоично-десятичное число X0, сумму чисел отобразить на первом и втором знакоместе статической индикации . |
| 2 | Занести в регистр R3 двоично-десятичное число XX отобразить его на первом и четвертом знакоместе статической индикации. |
| 3 | Занести в регистр B двоично-десятичное число, с частотой 2 Гц выводить это число на первом и втором знакоместе статической индикации . |
| 4 | Занести в Аккумулятор двоично-десятичное число XX, в регистр R5 X0, число из Аккумулятора отобразить на первом и втором знакоместе статической индикации, число из R5 отобразить на третьем знакоместе статической индикации. |
| 5 | Занести в регистр R2 двоично-десятичное число 0X, в регистр R5 X0, сумму чисел отобразить на втором и третьем знакоместе статической индикации. |
| 6 | Занести в ячейку с адресом B0h внутренней памяти ОЭВМ двоично-десятичное число 0X, в регистр R3 число X0, сумму чисел отображать на втором и третьем знакоместе статической индикации с частотой 0,5Гц. |
| 7 | Занести в регистр R0 двоично-десятичное число XX, попеременно отображать младшую и старшую тетраду на первом и четвертом знакоместе статической индикации с частотой 1 Гц. |
| 8 | Занести в B двоично-десятичное число X0, в регистр R1 XX, число из B отображать на первом знакоместе статической индикации с частотой 1 Гц, число из R1 отображать на третьем и четвертом знакоместе статической индикации с частотой 0,5 Гц. |
| 9 | Считать значение регистра TCON и отобразить его на третьем и четвертом знакоместе статической индикации. |
| 10 | Занести в регистр R4 двоично-десятичное число 0X, в регистр R3 X0, |

| № | Текст индивидуального задания |
|----|---|
| | сумму чисел отобразить на втором и третьем знакоместе статической индикации с медленным (в течение 5 сек.) затуханием этого числа. |
| 11 | Занести в Аккумулятор двоично-десятичное число $X0$, в регистр В $0X$, сумму чисел отобразить на первом и четвертом знакоместе статической индикации. |
| 12 | Занести в регистр В двоично-десятичное число $0X$, в регистр R5 $X0$, два разряда суммы (десятки и единицы) поочередно отображать на первом и втором знакоместе статической индикации |
| 13 | Занести в регистр R1 двоично-десятичное число $0X$, отнимая от числа единицу отображать на третьем знакоместе статической индикации полученное значение до нуля с частотой 1 Гц. |
| 14 | Занести в регистр R3 двоично-десятичное число XX , в регистр R5 XX , попеременно отображать эти числа на первом и втором знакоместе статической индикации (R3) и на третьем и четвертом знакоместе статической индикации (R5). |
| 15 | Занести в регистр А двоично-десятичное число $0X$, в регистр R2 $X0$, число из А отобразить на четвертом знакоместе статической индикации, число из регистра R2 отображать на втором знакоместе статической индикации с частотой в 0.5 Гц. |
| * | Занести в регистр R0 число XX , вывести на левой паре знакомест статической индикации. Через 2 секунды вывести число XX , занесенное в регистр R1. Через 2 секунды на правой паре знакомест статической индикации вывести разницу чисел занесенных в регистр R0 и R1. |
| * | Занести число $0X$ в регистр R0 и число XX в регистр R1. Вывести значение R1 на правой паре знакомест статической индикации. Затем прибавлять значение R1 к значению R0, до тех пор, пока результат не достигнет определенного, заранее заданного порога. Осуществить переход на начало программы. (Время задержки 0,5 секунд). |
| * | Занести число $0X$ в регистр R0 и число XX в регистр R1. Сумму вывести на правой паре знакомест статической индикации. Затем прибавлять значение R1 к значению R0, до тех пор, пока значение суммы не достигнет определенного, заранее заданного порога. Затем от результата отнимать число занесенное в регистр R2, причем R2 не равно R1. При достижении нуля вернуться к прибавлению значения регистра R1. |

Лабораторная работа №2. Способы построения схем отображения информации

Тема: Схемы отображения информации

Цель работы: Изучение схем динамической и статической индикации

Учебная задача: Разработка программ для ОЭВМ семейства MCS-51 для отображения цифровой информации на устройствах динамического и статического типа, а также на единичных индикаторах

Порядок выполнения лабораторной работы:

- 1) Изучить принцип работы различных методов отображения.
- 2) Разработать алгоритм для выполнения индивидуального задания до начала лабораторного занятия
- 3) Разработать программу для выполнения индивидуального задания до начала лабораторного занятия с использованием подпрограмм.
- 4) Ввести программу индивидуального задания на персональном компьютере.
- 5) Изучить программно отладочные средства (ПОС) для КР1816ВЕ31.
- 6) С помощью ПОС проанализировать выполнение индивидуальной программы
- 7) Загрузить программу в стенд ОЭВМ. Убедиться в правильном выполнении индивидуального задания, при отрицательном результате осуществить изменение алгоритма либо программы. Повторить загрузку программы в стенд ОЭВМ
- 8) Распечатать листинг правильно работающей программы.
- 9) Ответить на контрольные вопросы преподавателя

Контрольные вопросы:

1. Расчет времени регенерации для динамического метода отображения.
2. Обоснование необходимости применения различных методов отображения
3. Схемотехнические решения для построения схем отображения информации
4. Схемы включения единичных индикаторов .
5. Включение жидкокристаллических индикаторов
6. Включение газоразрядных индикаторов

Краткие теоретические сведения Системы отображения информации

Простейшими приборами отображения информации в цифровых устройствах являются светодиоды и цифровые индикаторы.

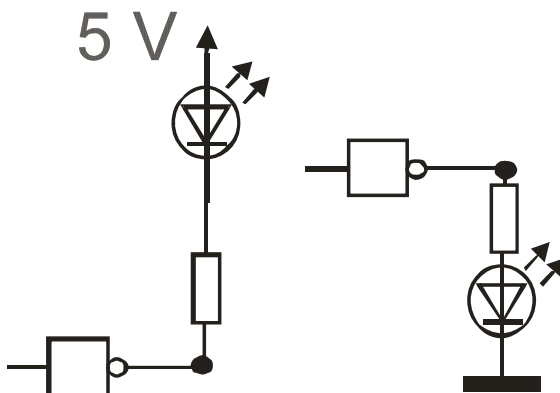


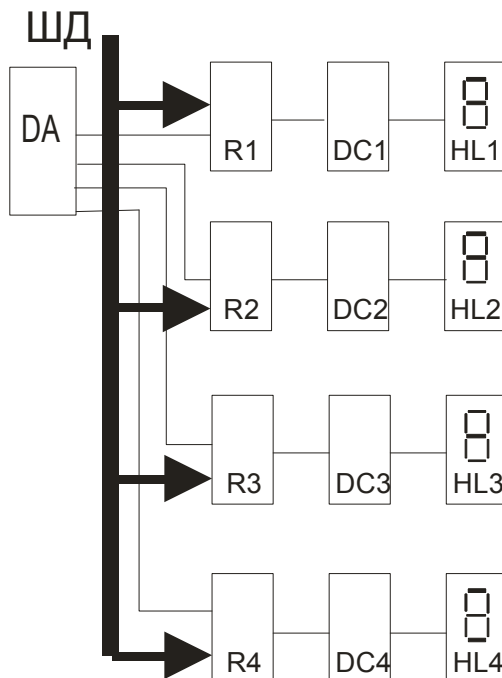
Рисунок 5. Включение единичных индикаторов.

В полупроводниковых светодиодах используется свойство p-n перехода излучать свет в видимой части спектра при протекании через него прямого тока ($I_{пр}=5-20\text{мА}$, $U_{пр}=2-3\text{В}$). Варианты включения индикаторов на рис.5.

Для отображения цифровой информации наибольшее распространение получили семисегментные индикаторы, в которых изображение цифры составляют из семи линейных светодиодных сегментов расположенных в виде цифры 8.

На основе светодиодов и семисегментных индикаторов строятся подсистемы отображения информации. При построении подсистем отображения информации различают два подхода - динамическая и статическая схема построения подсистемы индикации.

Статическая индикация заключается в постоянной подсветке индикаторов HL1-n от одного источника информации рис.6.



DA – дешифратор адреса, необходим для выборки соответствующего регистра

R1-R4- регистры, в которых временно хранится значение кода числа для отображения (соответствующий регистр выбирается DA).

DC1-DC4 – семи сегментные дешифраторы, преобразующие двоичный код в семи сегментный код.

HL1-HL4 – семи сегментные индикаторы.

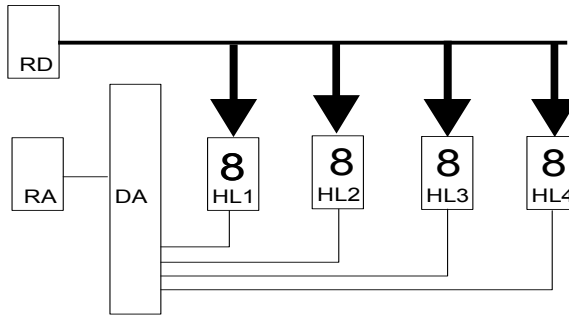
ШД – шина данных, по ней осуществляется передача данных на индикацию.

Рисунок 6. Структурная схема статической индикации.

В такой системе каждый индикатор HL1-n подключен через собственный дешифратор DC1-n и регистр-защелку RG1-n к шине данных, выборка регистров RG1-n производится при помощи селектора адреса CA. Аппаратные затраты при такой организации составляют n пар регистр + дешифратор при n десятичных разрядов индикатора.

Сущность динамической индикации заключается в поочередном циклическом подключении каждого индикатора HL1-n к источнику информации через общую шину данных, рис 7.

Выборка индикатора осуществляется дешифратором DA. В регистре RD хранится цифровой код, предназначенный для отображения. В регистре RA хранится адрес индикатора.



RD- регистр данных для временного хранения отображаемого числа либо символа.

RA- регистр адреса для временного хранения двоичного кода адреса выбираемого индикатора.

DA-для преобразования адреса задаваемого двоичным кодом а позиционный код.

HL1-HL4- семи сегментные индикаторы.

Рисунок 7. Структурная схема динамической индикации

При таком включении значительно уменьшаются аппаратные затраты. Но необходимо обеспечить достаточное время свечения одного индикатора, для того чтобы не уменьшалась яркость. Также необходимо обеспечить такую частоту перебора индикаторов, чтобы не было заметно мерцание. Преимущества такого способа заметны при количестве разрядов индикации больше 5.

В стенде статическая индикация реализована на четырех статических семисегментных индикаторах HG1 (разряды HG1.0, HG1.1, HG1.2, HG1.3). Обращение к ним производится, как к ячейкам памяти с адресами A000h(левая пара знакомест), B000h(правая пара знакомест).

Динамическая индикация реализована на плате расширения с помощью четырехразрядного семисегментного индикатора HL2. Управление динамической индикацией осуществляется с помощью порта В микросхемы системного контроллера (см. схему электрическую стенда, приложение 1) , сигналы выборки соответствующего индикатора поступают от линии порта PC0, PC1 к дешифратору адреса разряда DD3.

Знакосинтезирующая индикация реализована на плате расширения с помощью матрицы светодиодов 5x7 HG1. Управление светодиодной матрицей осуществляется по линиям PA0-PA4 и PC0-PC6. Например для того чтоб засветить точку с координатами [1;1] необходимо выставить уровень логической единицы на линии PA0, и уровень логического нуля на линии PC0.

Также в стенде есть линейка светодиодов HL1-HL8 доступ к которым осуществляется как к ячейке внешнего ОЗУ по адресу 0A006h. Светодиоды зажигаются записью логических единиц в соответствующие разряды.

Пример программы для статической индикации

С частотой 1 Гц отобразить на статическом индикаторе число 04:

```
CSEG
ORG 0
Continue:  mov A, #0
           mov DPTR, #0A004h
           movx @DPTR, A           ;отменить гашение знакомест C_инд

           mov A, #04h             ;записать в Акк. число 04
           mov DPTR, #0A000h       ;установить в DPTR адрес левой
                                   ;пары знакомест C_инд
           movx @DPTR, A           ;засветить число 04
           mov DPTR, #0B000h       ;установить в DPTR адрес правой
                                   ;пары знакомест C_инд
           movx @DPTR, A           ;засветить число 04
           CALL ZAD                ;вызов подпрограммы задержки

           mov A, #00001111b
           mov DPTR, #0A004h
           movx @DPTR, A           ;погасить все знакоместа C_инд
           CALL ZAD

           jmp Continue            ;переход на начало программы
ZAD:       mov R1, #0FFh
           mov R2, #0FFh
C2:        djnz R2, C4
C4:        djnz R1, C2
           ret                     ;выход с подпрограммы
END
```

На светодиодах HL1-HL8 запустить "бегущую единичку" уменьшая время задержки между засвечиванием светодиодов до определенного значения, после чего засветить все светодиоды. (Пример программы написан на языке C).

```
#include <8051.h>
#include "..\ev8031.lib\ev8031.c"
#include "..\ev8031.lib\bitdef.h"

int main()
{
    unsigned int svet,low,i,zad;

    righti=0x33;        //засветить числа 33 на правом
    lefti=0x33;         //и левом знакахестах ендикатора HG1
    zad=10000;

begin:
    low=1;
    LED_REG=low;        //в регистр светодиодов записываем
                        //значение low
    for (svet=1; svet<8; svet++) //номер засвеченного //светодиода HL1-HL8
    {
        delay16(zad);
        low=low<<1; //сдвинуть значение low на знак влево
        LED_REG=low;
    }
    low=256;
    for (svet=1; svet<=8; svet++)
    {
        zad=zad-100;
        if (zad==100) //уменьшаем задержку
            goto cont; //если значение zad достигло 100
                        //засветить все светодиоды
        else
            delay16(zad);
        low=low>>1; //сдвинуть значение low вправо
                    //на один знак вправо
        LED_REG=low;
    }
    goto begin;
cont:
    low=0xFF;
    LED_REG=low;
    for(i=1; i<=100;i++)
        delay16(10000);
    zad=10000;
    goto begin;
}
```

Варианты индивидуальных заданий

Таблица 3

Таблица заданий к л.р. №2

| №. | Текст индивидуального задания |
|----|---|
| 1 | Занести в регистр R1 XXH, отнимая от числа “1” отображать результат на динамическом индикаторе в младшем разряде до нуля с частотой 0,5 Гц. Включать бегущий светодиод на HL1-HL8 |
| 2 | Занести в В двоично-десятичное число X0, в регистр R1 XXH, число из В. отобразить на знако-синтезирующем индикаторе, число из R1 отображать на динамическом индикаторе в старшем разряде с частотой 0,5 Гц. |
| 3 | Включить в шахматном порядке светодиоды HL1-HL8. Занести в регистр в двоично-десятичное число 0X, в регистр R5 X0, два разряда суммы (десятки и единицы) поочередно отображать на статическом индикаторе и на динамическом индикаторе с частотой 1 Гц.. |
| 4 | Занести в R6 двоично-десятичное число XXH, в .R5 двоично-десятичное число XX, в R0 двоично-десятичное число XX, отображать эти числа с R5,R6 на динамическом индикаторе, с R0 на статическом индикаторе. |
| 5 | Попеременно отображать на знако-синтезирующем индикаторе числа от 0 до 9 дублировать эти числа на динамическом индикаторе. |
| 6 | Занести в регистр A двоично-десятичное число 0X, в регистр R2 X0, число из A отобразить на статическом индикаторе, число из регистра R2 отображать на динамическом индикаторе с частотой 0.6 Гц. |
| 7 | Занести в Аккумулятор двоично-десятичное число XX, в регистр R1 XX, младшие два разряда суммы чисел отобразить на динамическом индикаторе, при этом на знако-синтезирующем индикаторе осуществить плавное загорание числа 5. |
| 8 | Занести в регистр R6 число XXH преобразовать его в двоично-десятичное число отобразить его на динамическом индикаторе отобразить значение регистра R6 на светодиодах HL1-HL8 и его инверсное состояние с частотой 1Гц. |
| 9 | Занести в регистр В двоично-десятичное число XX, в регистр R3 XX, разность чисел отобразить на динамическом индикаторе. |
| 10 | Отобразить на знако-синтезирующем индикаторе букву У. Занести в Аккумулятор число XX, в регистр R5 X0, число из Аккумулятора отобразить на статическом индикаторе, число из R5 отобразить на динамическом индикаторе. |
| 11 | Занести в регистр R0 двоично-десятичное число XX, попеременно отображать младший и старший разряды на динамическом индикаторе с частотой 0,5 Гц. |

| №. | Текст индивидуального задания |
|----|--|
| 12 | Занести в регистр R2 двоично-десятичное число XX, в регистр R5 XX, сумму отобразить на динамическом индикаторе. |
| 13 | Занести в регистр В двоично-десятичное число, с частотой 2 Гц выводить это число на статическом индикаторе и одновременно на динамическом индикаторе. |
| 14 | Попеременно включать светодиоды HL1-HL8. Занести в ячейку с адресом 0010h внешней памяти ОЭВМ двоично-десятичное число 0X, в рег. R3 XXH, сумму чисел отобразить в старшем разряде на динамическом индикаторе. |
| 15 | Занести в регистр R1 двоично-десятичное число 0X, в регистр R3 XX, сумму отобразить на динамическом индикаторе. Шестнадцатеричное число отобразить на HL1-HL8. |
| * | Поочередно засвечивая светодиоды HL1-HL8, на статическом индикаторе параллельно высвечивать кол-во горящих светодиодов. Интервал между засвечиванием 1 с. |
| * | Отобразить на динамическом индикаторе слово из четырех букв. Вывести его следующим образом: Первая буква, из крайнего правого положения пройдя все сегменты остается гореть в крайнем левом положении. Последующие буквы, аналогично, пройдя все сегменты остаются за предыдущими. Когда слово полностью засветилось, гасить сегменты поочередно, начиная с крайнего левого сегмента с интервалом 0,5с. |

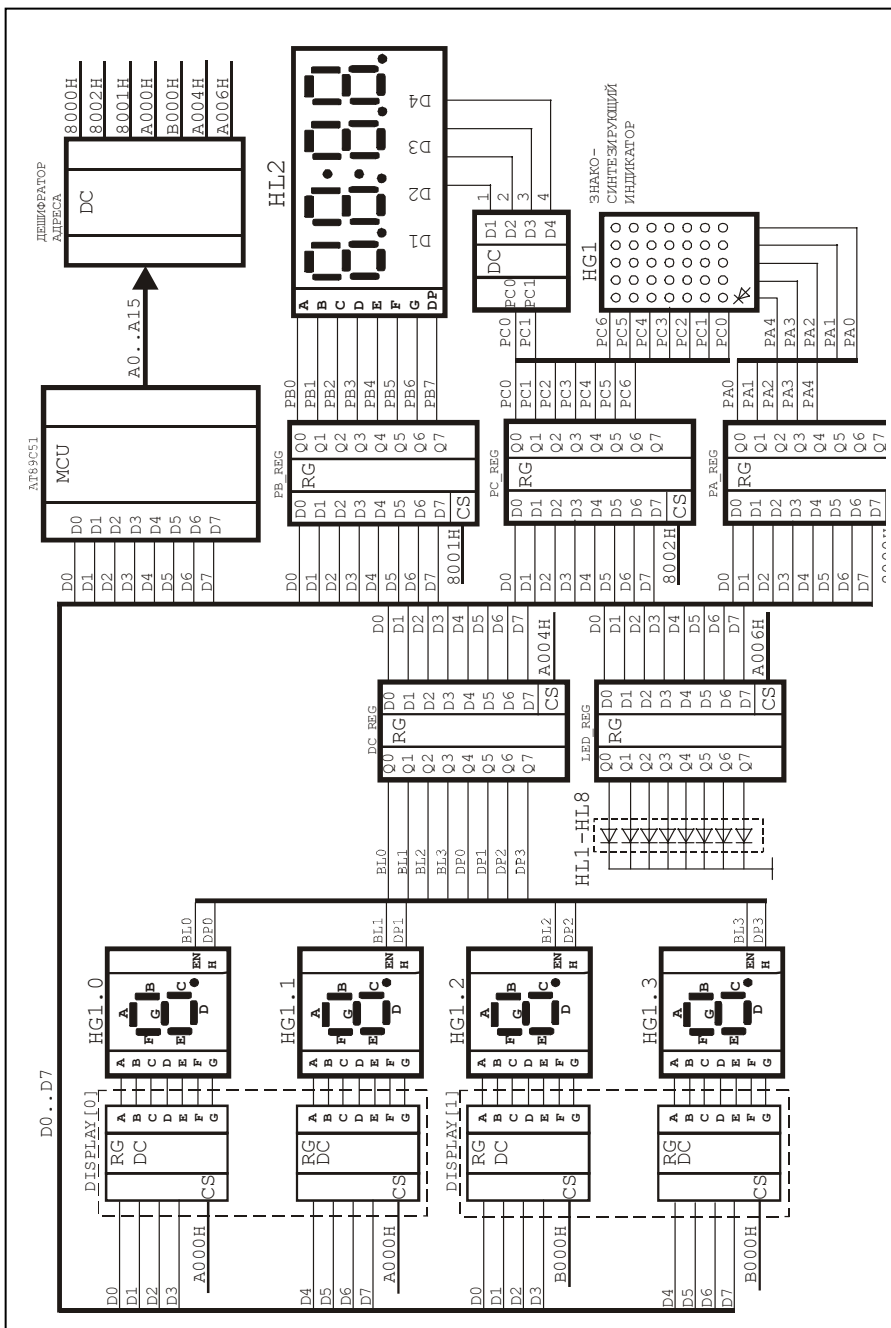


Схема для лабораторной работы №2

Лабораторная работа №3. Система прерываний. Опрос дискретных датчиков

Тема: Система прерываний. Опрос дискретных датчиков.

Цель работы: Изучение режимов работы системы прерывания ОЭВМ, программная обработка дискретных сигналов

Учебная задача: Изучение систем прерывания режимов ввода дискретной информации, разработка программ опроса сигналов от датчиков

Порядок выполнения лабораторной работы.

- 1) Изучить систему прерывания ОЭВМ K1816BE31, особенности опроса дискретных датчиков с механическими контактами.
- 2) Разработать алгоритм для выполнения индивидуального задания до начала лабораторного занятия.
- 3) Разработать программу для выполнения индивидуального задания до начала лабораторного занятия
- 4) Ввести программу индивидуального задания на ПК.
- 5) С помощью ПОС проанализировать выполнение индивидуальной программы
- 6) Загрузить программу в стенд ОЭВМ. Убедиться в правильном выполнении индивидуального задания, при отрицательном результате осуществить изменение алгоритма либо программы. Повторить загрузку программы в стенд ОЭВМ
- 7) Распечатать листинг правильно работающей программы.
- 8) Ответить на контрольные вопросы преподавателя

Контрольные вопросы:

1. Структура системы прерывания ОЭВМ K1816BE31.
2. Назначение, примеры применения системы прерывания
3. Регистры управления, распределение памяти в ОЭВМ 1816 BE31.
4. Прерывание от таймеров, последовательного приемопередатчика
5. Аппаратное устранениедребезга контактов для схем с TTL и КМОП.
6. Программное устранениедребезга контактов
7. Необходимость применения аппаратного либо программного устранениядребезга контактов

Краткие теоретические сведения

Опрос дискретных сигналов. Для ввода информации широко применяются кнопочные переключатели и контактные клавиатуры. Сигнал таких переключателей формируется путем замыкания (размыкания) электрической цепи. Сигнал, формируемый контактной парой, сопровождается дребезгом, длительность которого составляет ~8-12мс рисунки 8.

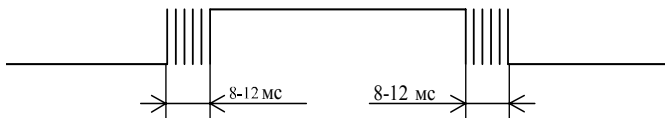


Рисунок 8. Сигнал контактной пары

Для устранения дребезга в получаемом сигнале на выходе контактной пары устанавливают специальные формирователи. Пример такого формирователя основанного на принципе непосредственной установки RS-триггера приведён на рисунке 9.

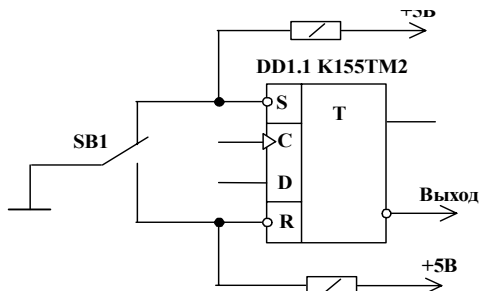


Рисунок 9. Схема устранения дребезга с помощью RS-триггера

Для уменьшения аппаратных затрат применяют программное подавление дребезга. Оно заключается в повторном опросе контактной пары с задержкой в 12мс, при совпадении результатов опроса кнопка была нажата, иначе в результате первого опроса был зафиксирован дребезг.

Система прерываний ОЭВМ 1816BE31

Регистр приоритетов прерывания (IP) предназначен для установки уровня приоритета прерывания для каждого из 5-ти источников прерывания.

Обозначение разрядов регистра IP показано в таблице 3, а назначение указано ниже.

Таблица 4 Регистр приоритетов прерываний IP

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|-----|-----|-----|-----|-----|
| X | X | X | PSP | PT1 | PX1 | PT0 | PX0 |

PX0 - установка уровня приоритета прерывания от внешнего источника /INT0.

PT0 - установка уровня приоритета прерывания от T/C0.

PX1 - установка уровня приоритета прерывания от внешнего источника. /INT1.

PT1 - установка уровня приоритета прерывания от T/C 1.

PS - установка уровня приоритета прерывания от последовательного порта.

X - резервный разряд.

Наличие в разряде IP "1" устанавливает для соответствующего источника высокий уровень приоритета, а наличие в разряде IP "0"-низкий уровень приоритета. При чтении резервных разрядов, соответствующие линии магистрали данных не определены.

Регистр разрешения прерываний (IE) предназначен для разрешения или запрещения прерываний от соответствующих источников. Обозначение разрядов регистра IE показано в таблице 4, а назначение указано ниже.

Таблица 5 Регистр разрешения прерываний IE

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----|---|---|----|-----|-----|-----|-----|
| EA | X | X | ES | ET1 | EX1 | ET0 | EX0 |

EA - управление всеми источниками прерываний одновременно. Если EA = 0, то прерывания запрещены. Если EA = 1, то прерывания могут быть разрешены индивидуальными разрешениями EX0, ET0, EX1, ET1, ES.

X - резервный разряд.

ES - управление прерыванием от последовательного порта. ES = 1 разрешение. ES = 0 - запрещение.

ET1 - управление прерыванием от т/с 1. ET = 1 -разрешение. ET1 = 0 - запрещение.

EX1 - управление прерыванием от внешнего источника /INT1. EX1 = 1 - разрешение. EX1 = 0 - запрещение.

ET0 - управление прерыванием от т/с 0. ET0 = 1 - разрешение. ET0 = 0 - запрещение.

EX0 - управление прерыванием от внешнего источника /INT0. EX0 = 1 - разрешение. EX0 = 0 - запрещение.

При чтении резервных разрядов соответствующие линии магистрали не определены.

Структура прерываний

Механизм прерываний в ОМЭВМ позволяет автоматически реагировать на внешние и на внутренние события (переполнение таймеров/счетчиков; завершение последовательного обмена).

Каждое из внешних прерываний /INT0, /INT1 может быть активизировано по уровню или по фронту сигналов P3.2, P3.3 с помощью битов IT0 и IT1 регистра TCON. При поступлении запроса внешнего прерывания /INTx устанавливается флаг IEx регистра TCON. Очистка флага IEx производится аппаратно: при прерывании по фронту IEx сбрасывается при обращении к соответствующей подпрограмме обработки прерывания; при прерывании по уровню флага очищается при снятии запроса внешнего прерывания, то есть в IEx отслеживается состояние вывода /INTx.

Чтобы внешнее прерывание по уровню было распознано, необходимо, чтобы низкий уровень на выводе INTx удерживался в течение не менее 12 периодов сигнала тактовой частоты. Если же прерывание активизируется по переходу из состояния высокого уровня в состояние низкого уровня, то циклу низкого уровня должен предшествовать цикл высокого уровня на выводе /INTx. Если внешнее прерывание активизируется по уровню, запрос должен удерживаться до начала обслуживающей подпрограммы и сниматься перед завершением этой подпрограммы для предотвращения повторного обслуживания.

Прерывания от таймеров/счетчиков выполняются по флагам TF0 и TF1 регистра TCON, которые устанавливаются при переполнении соответствующих регистров таймеров/счетчиков (за исключением режима 3). Очистка флагов TF0 и TF1 происходит при переходе к подпрограмме обслуживания прерывания.

Прерывание от последовательного порта выполняется по флагу окончания приема RI или по флагу окончания передачи TI, которые устанавливаются в регистре SCON.

В отличие от всех остальных флагов, RI и TI сбрасываются только программным путем обычно в пределах подпрограммы обработки прерывания, где определяется, какому из флагов RI или TI соответствует прерывание.

В случае одновременного поступления запросов прерывания с одинаковым уровнем приоритета, равным 0 или 1, обработка их производится в порядке внутреннего опроса флагов:

IEO -> TFO -> TE1 -> TF1 -> (TI + RI)

Установка флагов прерывания происходит в конце машинного цикла, а их опрос в следующем цикле. И только после выполнения последнего цикла текущей команды производится аппаратный вызов соответствующей подпрограммы обслуживания, эквивалентный команде LCALL.

В общем случае, обращение к подпрограмме обслуживания задерживается при выполнении хотя бы одного из следующих условий:

- производится обработка прерывания с таким же или высшим приоритетом;
- текущий машинный цикл (цикл опроса флага) не является последним циклом выполняемой команды;
- выполняемая команда текущей программы RETI или любая команда обращения к регистрам IE, IP.

В последнем условии после окончания одной из вышеуказанных команд обязательно выполнится еще одна команда текущей программы перед вызовом подпрограммы обслуживания прерывания.

Флаг прерывания, установленный во время действия блокировки прерывания по одному из трех указанных условий и сброшенный до их снятия, не вызовет обслуживания соответствующего запроса прерывания.

Подпрограмма обслуживания прерывания продолжается до выполнения команды RETI по которой восстанавливается состояние логики прерывания и состояние программного счетчика PC из двух верхних ячеек стека. При использовании команды RET восстанавливается только состояние программного счетчика, а состояние логики прерывания остается неизменным.

Таблица 6 Начальные адреса векторов прерываний

| <i>Источник прерывания</i> | <i>Адрес</i> |
|----------------------------|--------------|
| Внешнее прерывание 0 | 0003H |
| Переполнение таймера 0 | 000BH |
| Внешнее прерывание 1 | 0013H |
| Переполнение таймера 1 | 001BH |
| Последовательный порт | 0023H |

В составе учебно-отладочного стенда имеется две отдельные кнопки S10-S11, которые могут опрашиваться, как программно, так и с помощью использования функций прерываний INT0-INT1 соответственно.

В составе учебно-отладочного стенда имеется матричная 3x4 клавиатура SW3-SW14. Клавиатура подключена к шине данных ОЭВМ при помощи микросхемы буфера DD1 74245(АП6).

Опрос всей клавиатуры производится за три раза (за один раз считывается состояние только одного столбца клавиатуры). Чтобы произвести опрос столбца клавиатуры (SW3,SW6,SW9,SW12; SW4,SW7,SW10,SW13; или SW5,SW8,SW11,SW14) необходимо выставить на соответствующей линии адреса (A0,A1,A2 для первого, второго и третьего столбца соответственно) уровень логического нуля, а на других линиях уровень логической единицы и прочесть состояние буфера клавиатуры, подключенного к шине данных ОЭВМ как доступная для чтения ячейка памяти с адресом 9000h. Если кнопка нажата то соответствующий бит в считанном байте будет равен нулю, если же не нажата то единице.

Таблица 7

| Столбец (кнопки) | Адрес |
|-----------------------|-------|
| 1 (SW3,SW6,SW9,SW12) | 9006h |
| 2 (SW4,SW7,SW10,SW13) | 9005h |
| 3 (SW5,SW8,SW11,SW14) | 9003h |

Пример программного опроса дискретного сигнала

```

CSEG
ORG 0
    jnb p3.2,$                ;опрос нажатия кнопки SW15, если
                                ;кнопка нажата, программа выполняется
                                ;дальше

Continue:
    mov A,#0
    mov DPTR,#0A004h
    movx @DPTR,A              ;отменить гашение знакомест C_инд

    mov A,#04h                 ;записать в Акк. число 04
    mov DPTR,#0A000h           ;установить в DPTR адрес левой пары
                                ;знакомест C_инд
    movx @DPTR,A               ;засветить число 04
    mov DPTR,#0B000h           ;установить в DPTR адрес правой пары
                                ;знакомест C_инд
    movx @DPTR,A               ;засветить число 04
    CALL ZAD                   ;вызов подпрограммы задержки
    mov A,#00001111b
    mov DPTR,#0A004h
    movx @DPTR,A               ;погасить все знакоместа C_инд
    CALL ZAD

    jmp Continue                ;переход на начало программы
ZAD:
    mov R1,#0FFh
C2:    mov R2,#0FFh
C4:    djnz R2, C4
        djnz R1, C2
        ret                    ;выход с подпрограммы

```

END

Отображать на C_инд. Числа 33.33 и 32.32 при нажатии кнопок SW16, SW15 соответственно (программа написана на языке C) .

```

#include <8051.h>
#include "..\ev8031.lib\ev8031.c"
#include "..\ev8031.lib\bitdef.h"

int main()
{
begin:
    DC_REG=0x0F;                //гасить индикатор HG1
    if (!P3_2)
    {
        new_dotsi=0x20;
        lefti=0x32;
        righti=0x32;            //если кнопка
                                //SW15 нажата вывести 32.32
    }
    if (!P3_3)
    {
        new_dotsi=0x20;
        lefti=0x33;
        righti=0x33;            //если кнопка
                                //SW16 нажата вывести 33.33
    }

    goto begin;
}

```

Варианты индивидуальных заданий

Таблица 8

Таблица заданий к л.р. №3

| № | Текст индивидуального задания |
|----|---|
| 1 | Подсчитать и отобразить на статическом индикаторе количество нажатий кнопки SW15 |
| 2 | Реализовать опрос клавиатуры. Номер клавиши отображать путем засвечивания соответствующей точки на знако-синтезирующем индикаторе. |
| 3 | Реализовать опрос клавиатуры. Номер клавиши последовательно отображать в каждом разряде динамическом индикаторе. |
| 4 | По нажатии SW15 запускать бегущий огонь на светодиодах HL1-HL8, при отпускании плавное загорание числа 3 на знако-синтезирующем индикаторе. |
| 5 | По нажатии SW16 включить секундомер с отображением на статическом индикаторе значения секунд при отпускании запускать бегущую “тень” на светодиодах HL1-HL8 |
| 6 | Реализовать опрос клавиатуры. Номер клавиши индицировать позиционным кодом на светодиодах HL1-HL8, с отображением значения кнопки на динамическом индикаторе. |
| 7 | Реализовать программу ввода четырехзначного числа с клавиатуры, используя статическом индикаторе и дублируя значение нажатой кнопки на знако-синтезирующем индикаторе. |
| 8 | Реализовать опрос клавиатуры после 2-х нажатий SW16. Номер клавиши отображать на динамическом индикаторе. |
| 9 | По нажатии SW15 запускать любое бегущее значение на знако-синтезирующем индикаторе, а по нажатии SW16 зажечь все точки в шахматном порядке. |
| 10 | Отображать значение секунд на статическом индикаторе. По прерыванию INT0 остановить секундомер и засветить светодиоды HLn (n-нечетное) |
| 11 | Отображать число 7543 на динамическом индикаторе. По прерыванию INT1 засветить светодиоды HLn (n-четное) |
| 12 | На статическом индикаторе отобразить число 5555. По прерыванию INT0 отображать “шахматку” на знако-синтезирующем индикаторе, По прерыванию INT1 отобразить на статическом индикаторе число 3333 |
| 13 | После нажатия SW15 реализовать программу ввода трехзначного числа с клавиатуры с отображением на статическом индикаторе. |
| 14 | По нажатии SW16 запускать бегущую “тень” на знако-синтезирующем индикаторе, а при повторном нажатии SW16 потушить все точки. |
| 15 | Реализовать опрос клавиатуры. Номер клавиши индицировать двоичным кодом на светодиодах HL1-HL8 |

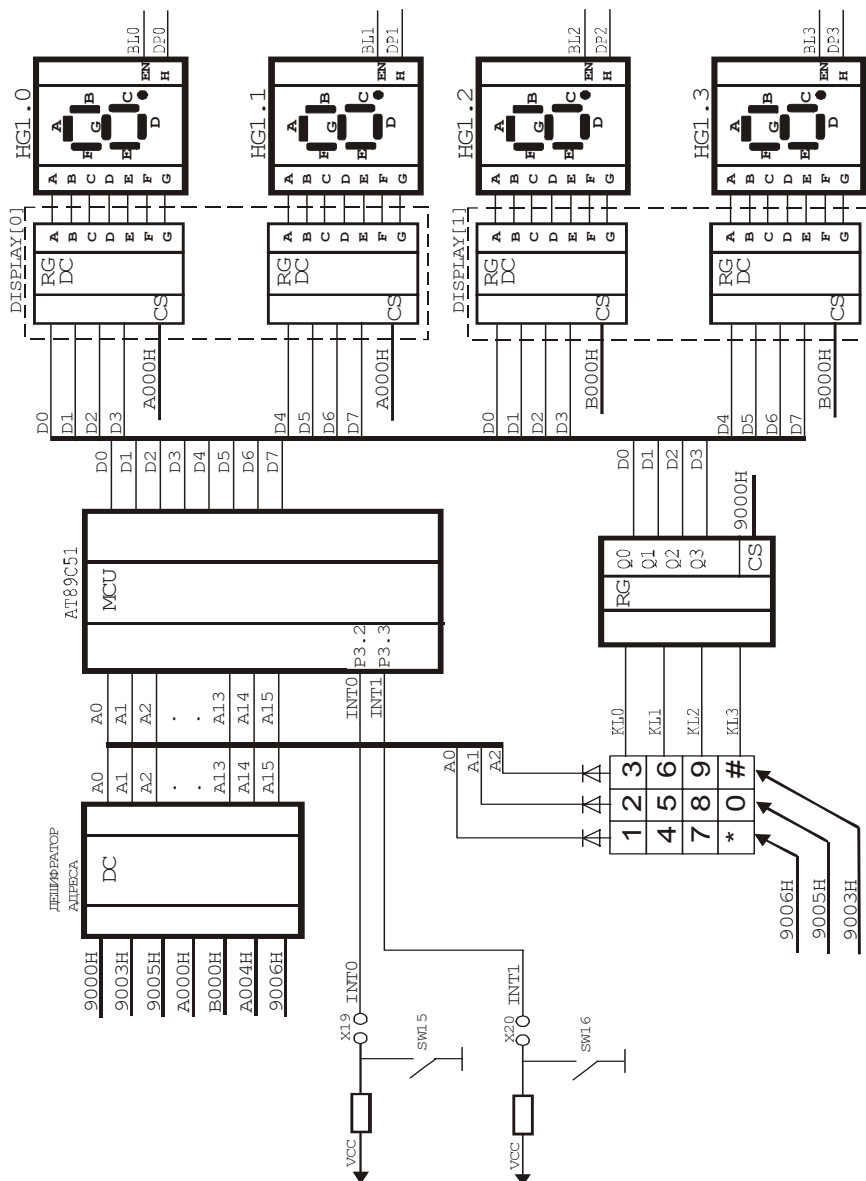


Схема к лабораторной работе №3.

Лабораторная работа №4. Цифроаналоговое преобразование

Тема: Цифроаналоговое преобразование

Цель работы: Изучение методов цифроаналогового преобразования

Учебная задача: Разработка программ для формирования различных аналоговых сигналов.

Порядок выполнения лабораторной работы

- 1) Изучить структурную схему модуля ЦАП на плате расширения
- 2) Разработать алгоритм для выполнения индивидуального задания до начала лабораторного занятия
- 3) Разработать программу для выполнения индивидуального задания до начала лабораторного занятия
- 4) Ввести программу индивидуального задания на персональном компьютере.
- 5) С помощью ПОС проанализировать выполнение индивидуальной программы
- 6) Загрузить программу в стенд ОЭВМ. Убедиться в правильном выполнении индивидуального задания (формирование заданной формы сигнала) используя осциллограф, при отрицательном результате осуществить изменение алгоритма либо программы. Повторить загрузку программы в стенд ОЭВМ
- 7) Распечатать листинг правильно работающей программы.
- 8) Ответить на контрольные вопросы преподавателя

Контрольные вопросы:

1. Методы и типы ЦАП.
2. Статические параметры ЦАП
3. Понятие дискретности, квантование, разрешающая способность ЦАП
4. Характеристика преобразования, нелинейность ЦАП.
5. Напряжение смещения нуля, допустимое напряжение на выходе ЦАП.
6. Динамические параметры ЦАП
7. Факторы, влияющие на погрешность ЦАП
8. Аппаратные реализации ЦАП
9. Примеры практического применения ЦАП

Краткие теоретические сведения

Цифро-аналоговые преобразователи (ЦАП) вырабатывают напряжение или ток, функционально связанные с управляющим кодом. Применяются ЦАП для формирования выходных аналоговых сигналов цифровых измерительных и вычислительных устройств. Для преобразования двоичного кода в аналоговый сигнал обычно формируются токи, пропорциональные

весам разрядов кода, и затем суммируются те из токов, которые соответствуют ненулевым разрядам входного кода.

Применяются в основном два метода ЦАП: суммирование единичных эталонных величин и суммирование эталонных величин, веса которых различаются. В первом при формировании выходной аналоговой величины используется только одна эталонная величина весом в один квант. Во втором методе применяются эталонные величины с весами, зависящими от номера разряда, и в суммировании участвуют только те эталонные величины, для которых в соответствующем разряде входного кода установлена единица.

В случае использования на входе двоичного позиционного кода значения всех разрядов поступает одновременно, и работа таких ЦАП описывается выражением

$$X = \frac{P}{2^b - 1} (a_0 \cdot 2^0 + a_1 \cdot 2^1 + a_2 \cdot 2^2 + \dots + a_{b-1} \cdot 2^{b-1})$$

где X - аналоговая величина.

a_i - коэффициенты соответствующих двоичных разрядов, которые принимают дискретные значения единица или ноль.

P - опорный сигнал

b - число разрядов

В преобразователях из опорного напряжения формируются эталонные величины, соответствующие значениям разрядов входного кода, которые суммируются и образуют дискретные значения выходной аналоговой величины.

Классификация ЦАП может быть проведена по признакам.

Способу формирования выходного напряжения (С суммирование напряжений, делением напряжений, суммированию токов).

Роду выходного сигнала (с токовым выходом, выходом по напряжению).

Полярности выходного напряжения (постоянному, переменному) и т.д. Основные структуры используемые в ЦАП интегрального исполнения –это:

Структуры с суммированием токов.

ЦАП - со взвешенными резисторами в цепях эмиттеров.

ЦАП - со взвешенными резисторами в цепях нагрузки.

ЦАП с лестничной матрицей $R=2R$ в цепях эмиттеров транзисторов источников токов.

ЦАП с выходной лестничной матрицей $R=2R$

Параметры ЦАП.

Число разрядов n управляющего кода, номинальный выходной ток, время установления выходного сигнала после изменения входного управляющего кода, погрешность полной шкалы, погрешность линейности, дифференциальная нелинейность.

Погрешности ЦАП могут быть выражены в процентах или других относительных единицах, а также в долях кванта

Упрощенная схема ЦАП для понимания работы преобразования входного кода в ток (напряжение) определенной величины.

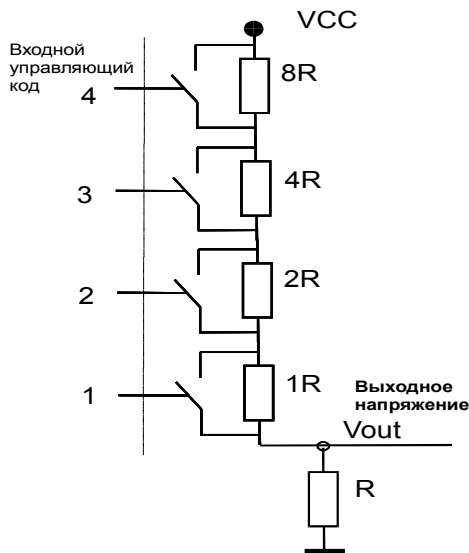


Рисунок 10.

Если соотношение резисторов ЦАП как 8R, 4R, 2R, 1R, то при включении всех коммутаторов, в соответствии с законом Ома напряжение в точке OUT будет равняться напряжению VCC. Если включить коммутатор резистора R8, напряжение в точке OUT будет равно $VCC/2$. Аналогично можно знать значение напряжения, при различных комбинациях входного кода.

В составе учебно-отладочного стенда имеется микросхема ЦАП AD7801. Она представляет собой восьмиразрядный ЦАП (с параллельным интерфейсом) с размахом выходного напряжения равным напряжению питания. Для наблюдения выходного сигнала с микросхемы ЦАП необходимо подключить осциллограф к BNC разъему и замкнуть перемычку J4. Доступ к ЦАП осуществляется как к ячейке внешнего ОЗУ по адресу 0F000h.

Пример выполнения программы цифро-аналогового преобразования

```
ORG 0
Begin:
    mov A, #0                ;записать в Акк. 0
    mov DPTR, #0F000h        ;установить в DPTR адрес ЦАПa
    movx @DPTR, A            ;установить на ЦАП код из Акк.

    call ZAD                 ;вызов подпрограммы задержки
continue:
    inc A                    ;увеличить значение Акк. на единицу
    movx @DPTR, A            ;установить на ЦАП код из Акк.
    call ZAD
    cjne A, #255d, continue ;если код не достиг максимального значения
                                ;перейти на метку continue
    jmp Begin                ;иначе переход на метку Begin
                                ;подпрограмма задержки
ZAD:
    mov R4, #005h
C2:    mov R2, #0FFh
C3:    djnz R2, C3
        djnz R4, C2
        ret                ;выход с подпрограммы
END
```

Сформировать пилообразный сигнал. ____ (программа написана на языке C) .

```
#include <8051.h>
#include "..\ev8031.lib\ev8031.c"
#include "..\ev8031.lib\bitdef.h"

int main()
{
    unsigned char y;
    DC_REG=1;
    lefti=0x0D;
    righti=0xAC;
    y=0;
    DAC_REG=y;
    delay16(10);
begin:
    for (y=1; y<255; y++) //увеличивать код ЦАПa на
                            //единицу до максимального
    {
        DAC_REG=y;
        delay16(10);
    }
    for (y=255; y>1; y--) //уменьшать код ЦАПa на
                            //единицу до минимального
    {
        DAC_REG=y;
        delay16(10);
    }
    goto begin;
}
```

Варианты индивидуальных заданий

Таблица 9

Таблица заданий к л.р. №4

| № | Текст индивидуального задания |
|----|--|
| 1 | Сформировать пилообразное напряжение с частотой повторения 50 Гц. Отображать на статическом индикаторе число сгенерированных импульсов. |
| 2 | По нажатию SW3 сформировать треугольные импульсы, передний фронт 20мсек задний 10 мсек, каждый 10 импульс отображать на динамическом индикаторе. |
| 3 | По нажатию SW4 сформировать трапецевидные импульсы, передний фронт 13мсек задний 15 мсек каждую секунду зажигать светодиод HL1. |
| 4 | Сформировать синусоиду с частотой повторения 120 Гц. |
| 5 | Сформировать пилообразное напряжение с частотой повторения 200Гц и длительностью переднего фронта 2 мсек. |
| 6 | По нажатию SW5 сформировать синусоиду с частотой повторения 100Гц. |
| 7 | Сформировать прямоугольные импульсы, с длительностью 25мсек и скважностью 4 |
| 8 | По нажатию SW16 сформировать треугольные импульсы, передний фронт 25мсек задний 5 мсек. |
| 9 | Сформировать синусоиду с частотой повторения 300 Гц. По нажатию SW15 изменить частоту на 100 Гц. |
| 10 | Сформировать два прямоугольных импульса, один максимальной амплитудой длительностью и второй 2/3 амплитуды максимальной с периодом повторения 40 Гц. |
| 11 | Сформировать прямоугольные импульсы, с длительностью 25мсек по нажатию SW6 на клавиатуре сформировать треугольные импульсы. |
| 12 | Сформировать синусоиду с частотой повторения 70 Гц, по нажатию SW7 на клавиатуре прямоугольные импульсы, с длительностью 25мсек и скважностью 2. |
| 13 | По нажатию SW15 сформировать треугольные импульсы, передний фронт 15мсек задний 40 мсек. |
| 14 | По нажатию SW9 на клавиатуре сформировать трапецевидные импульсы, передний фронт 20сек задний 20 мсек |
| 15 | Сформировать три прямоугольных импульса, один 1/3 максимальной амплитуды, 2-ой 2/3 максимальной амплитуды, 3-ий максимальной амплитуды с периодом повторения 100 Гц. |



Лабораторная работа №5. Аналого-цифровое преобразование

Тема: Аналого-цифровое преобразование

Цель работы: Научиться измерять аналоговую величину.

Учебная задача: Разработка программ измерения аналоговых величин для различных методов измерения и типов АЦП

Порядок выполнения лабораторной работы

- 1) Изучить структурную схему модуля АЦП на плате расширения
- 2) Разработать алгоритм и программу для выполнения индивидуального задания до начала лабораторного занятия
- 3) Ввести программу индивидуального задания на персональном компьютере.
- 4) С помощью ПОС проанализировать выполнение индивидуальной программы
- 5) Загрузить программу в стенд ОЭВМ. Убедиться в правильном выполнении индивидуального задания, изменить значение подаваемого напряжения на вход АЦП, повторить преобразование, при отрицательном результате осуществить изменение алгоритма либо программы.
- 6) Повторить загрузку программы в стенд ОЭВМ
- 7) Распечатать листинг правильно работающей программы.
- 8) Ответить на контрольные вопросы преподавателя

Контрольные вопросы

1. Методы и типы АЦП.
2. Статические параметры АЦП
3. Понятие дискретности, квантование, разрешающая способность
4. Характеристика преобразования, дифференциальная нелинейность АЦП, отклонение коэффициента преобразования.
5. Напряжение смещения нуля.
6. Динамические параметры АЦП
7. Время преобразования, время задержки запуска, время цикла преобразования, максимальная частота преобразования.
8. Понятие апертурное время.
9. Факторы, влияющие на погрешность АЦП
10. Аппаратные реализации АЦП.
11. Примеры практического применения АЦП
12. Построение схем АЦП с помощью микросхем ЦАП

Краткие теоретические сведения.

Аналогово-цифровые преобразователи (АЦП) применяются в измерительных системах и измерительно-вычислительных комплексах для согласования аналоговых источников измеряемых сигналов с цифровыми устройствами обработки и представления результатов измерения.

Существуют различные методы построения АЦП. Они отличаются по сложности реализации, помехоустойчивости, быстродействию.

В системах где основным критерием является быстродействие применяют АЦП параллельного преобразования. Но АЦП этого типа достаточно сложны в реализации. Для n -разрядного АЦП необходимо $2^n - 1$

компараторов и параллельный делитель напряжения, который вырабатывает $2^n - 1$ уровней квантования.

Для реализации систем с высокой помехоустойчивостью применяют интегрирующие АЦП. Такой АЦП состоит из двух преобразователей. Измеряемое напряжение преобразовывается в длительность импульса, а потом длительность импульса преобразовывается в цифровой код.

Одним из самых распространенных является АЦП, построенный на цифро-аналоговом (ЦАП) преобразователе. Схема этого АЦП приведена на рисунке 11.

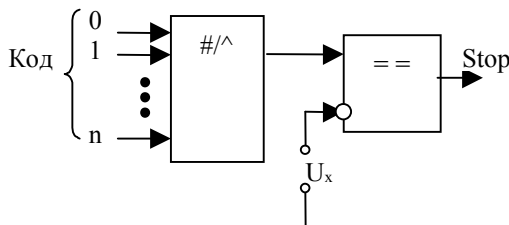


Рисунок 11. Схема АЦП построенного на ЦАП

Код формируется счетчиком, при организации жесткой логики, или программно, если АЦП работает в составе вычислительного комплекса. Входной код преобразовывается в аналоговый сигнал с помощью ЦАП. Напряжение с выхода ЦАП поступает на один из входов компаратора. На другой вход подается измеряемое напряжение U_x . В момент, когда напряжение ЦАП будет равно измеряемому, компаратор формирует сигнал 'Stop' который свидетельствует об окончании цикла измерения.

При формировании кода используются различные алгоритмы. Простейшим алгоритмом является поразрядное уравнивание. При таком подходе код меняется от минимального путем приращения единицы младшего разряда до тех пор, пока напряжение ЦАП не сравняется с измеряемым напряжением. Недостатком поразрядного уравнивания является маленькое быстродействие.

Для сокращения времени преобразования применяется метод половинных приближений. Уравнивание начинается со старшего разряда. В этом разряде устанавливается единица и читается состояние компаратора. Если напряжение ЦАП больше измеряемого то разряд сбрасывается, а если меньше, то разряд сохраняет свое состояние. Далее таким же образом обрабатывается следующий разряд. Преобразование заканчивается тогда, когда будут обработаны все разряды.

В системах слежения, за какими либо параметрами часто необходимо непрерывно считывать состояние датчика. Это обеспечивается малым временем преобразования за счет применения следящего АЦП. Суть данного

алгоритма заключается в том, что сначала код формируется методом половинных приближений. А после сравнения с измеряемым напряжением АЦП отслеживает изменение напряжения. Если напряжение растет то код поразрядно увеличивается до тех пор, пока напряжение ЦАП не сравняется с измеряемым, и наоборот.

В стенде АЦП построен на микросхемах AD7801(восьмиразрядный ЦАП) и LM358 (ОУ в качестве компаратора) (см. схему стенда). Состояние компаратора можно считать с вывода порта P1.7 однокристалльной ЭВМ, о окончании цикла преобразования также свидетельствует свечение светодиода подключенного к выходу компаратора. Доступ к ЦАП осуществляется как к ячейке внешнего ОЗУ по адресу 0F000h.

Измеряемое напряжение U_x формируется переменным резистором (если установлена перемычка J5), либо источник сигнала подключается к клеммнику.

В расширенной комплектации стенд поставляется с интегральным десятибитным АЦП с параллельным интерфейсом AD7813. Доступ к АЦП AD7813 осуществляется как к ячейке внешнего ОЗУ по адресу 0E000h.

Пример программы для аналого-цифрового преобразования.

```
ORG 0
Begin:
    mov A, #0                      ;записать в Акк. 0
nextTest:
    mov DPTR, #0F000h              ;установить в DPTR адрес ЦАПа
    movx @DPTR, A                  ;установить на ЦАП код из Акк.
    mov r3, #50;
z2:    djnz r3, z2
jnb    P1.7, ShowResult            ;проверка срабатывания компаратора
    inc A                          ;увеличить значение Акк. на единицу
    cjne A, #0ffh, nextTest        ;если код не достиг максимального значения
ShowResult:
    mov DPTR, #0B000h              ;вывод значения кода на индикатор GH1
    movx @DPTR, A
    call ZAD
    jmp Begin                      ;иначе переход на метку Begin
ZAD:
    mov R4, #0ffh
C2:    mov R2, #0FFh
C3:    djnz R2, C3
        djnz R4, C2
        ret                        ;выход с подпрограммы
END
```

(программа на языке C) .

```
#include <8051.h>
#include "..\ev8031.lib\ev8031.c"
#include "..\ev8031.lib\bitdef.h"

int main()
{
    unsigned char y,x,z;
    DC_REG=1;
    lefti=0x0A;
    righti=0xDC;
start:
    if (!P3_2)                     //запустить программу при
                                    //нажатии кнопки SW15
    {
begin:
        y=0;
        DAC_REG=y;
        for (y=0; y<255; y++) /*подавать код на ЦАП
увеличивая его на 1 до тех пор пока напряжение с ЦАПа не
сравнится с неизвестным напряжением*/
        {
            DAC_REG=y;
            delay16(10);
            if (!P1_7)
            {
                x=y/100;
                lefti=x;
                z=x*100;
                x=y-z;
                righti=x; //вывести код на индикатор HG1
                delay16(300);
                goto begin;
            }
        }
        goto begin;
    }
    goto start;
}
```

Варианты индивидуальных заданий

Таблица 10 Таблица заданий к л.р. № 5

| <i>№</i> | <i>Алгоритм для реализации</i> |
|----------|---|
| 1 | Поразрядное уравнивание, значение отображать на статическом индикаторе. |
| 2 | Метод половинных приближений, значение отображать на динамическом индикаторе. |
| 3 | Следящий с первоначальным поразрядным уравниванием, значение отображать на статическом индикаторе. |
| 4 | Следящий с первоначальным половинным приближением, значение отображать на статическом индикаторе. |
| 5 | По нажатию кнопки SW15 запустить АЦП половинных приближений, значение отображать на динамическом индикаторе. |
| 6 | По нажатию кнопки SW3 на клавиатуре запустить АЦП поразрядного уравнивания, значение отображать на статическом индикаторе. |
| 7 | По нажатию кнопки SW4 на клавиатуре запустить АЦП следящего типа с первоначальным поразрядным уравниванием, значение отображать на динамическом индикаторе. |
| 8 | По нажатию кнопки SW5 на клавиатуре запустить АЦП следящего типа с первоначальным половинным приближением, значение отображать на статическом индикаторе. |
| 9 | По нажатию кнопки SW6 на клавиатуре запустить АЦП половинных приближений результат отобразить на динамическом индикаторе по нажатию кнопки SW3 |
| 10 | По прерыванию INT0 запустить АЦП поразрядного уравнивания результат отобразить на статическом индикаторе по нажатию кнопки SW3 |
| 11 | По нажатию кнопки SW7 запустить АЦП следящего типа с первоначальным поразрядным уравниванием результат отобразить по нажатию кнопки SW8 |
| 12 | По нажатию кнопки SW8 запустить АЦП следящего типа с первоначальным половинным приближением результат отобразить на динамическом индикаторе по нажатию кнопки SW9 |

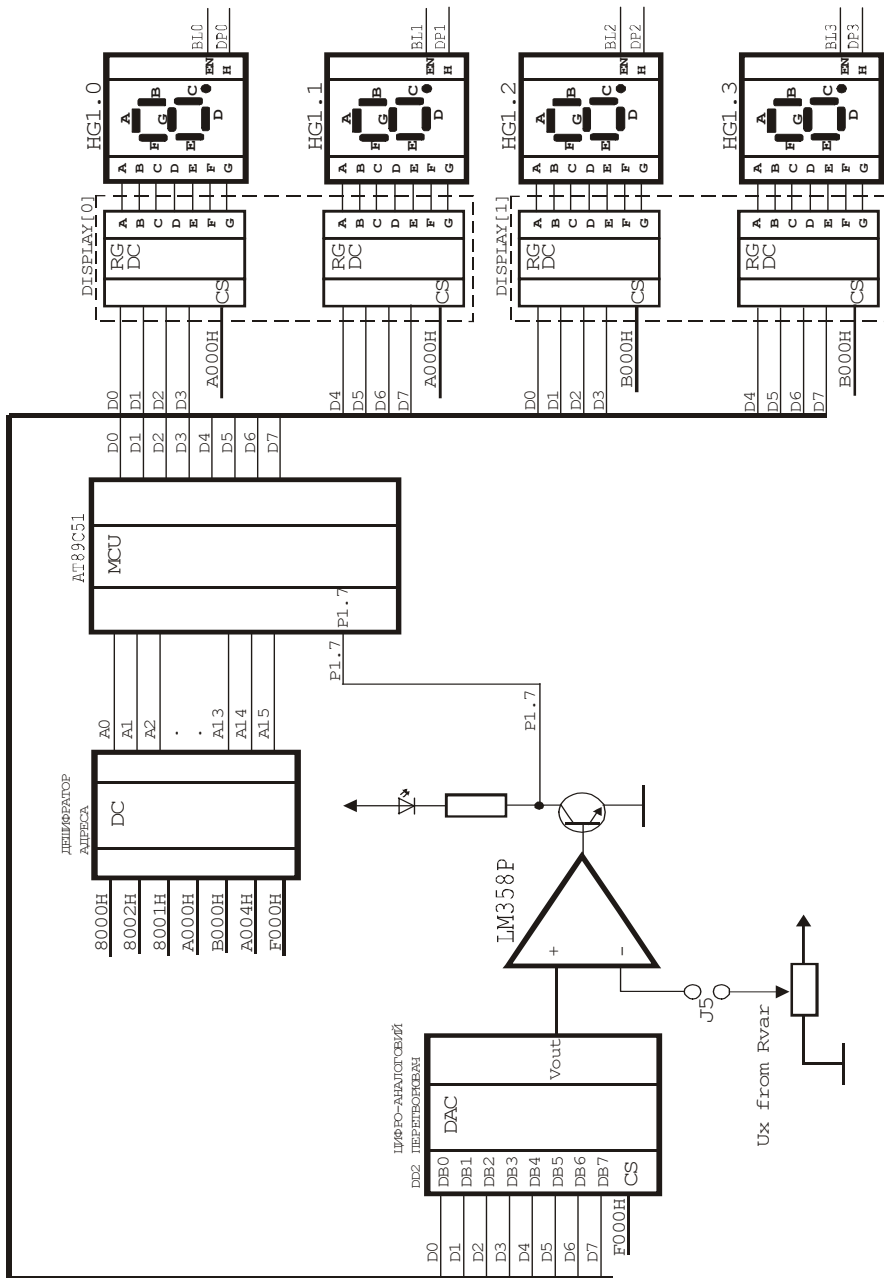


Схема к лабораторной работе №5

Лабораторная работа №6. Обработка частотных и временных сигналов

Тема: Обработка частотных и временных сигналов

Цель работы: Изучение методов частотного преобразования

Учебная задача: Научиться осуществлять измерение частоты, периода, длительность дискретных сигналов с помощью программируемых счетчиков/ таймеров ОЭВМ 1816 ВЕ31, а также с использованием внешнего эталонного генератора.

Порядок выполнения лабораторной работы.

- 1) Изучить методы частотного преобразования (аппаратные и с помощью программируемых счетчиков таймеров)
- 2) Изучить методы измерения временных интервалов.
- 3) Разработать алгоритм для выполнения индивидуального задания до начала лабораторного занятия.
- 4) Разработать программу для выполнения индивидуального задания до начала лабораторного занятия
- 5) Ввести программу индивидуального задания на персональном компьютере.
- 6) С помощью ПОС проанализировать выполнение индивидуальной программы
- 7) Загрузить программу в стенд ОЭВМ. Убедиться в правильном выполнении индивидуального задания, изменить значение измеряемой частоты, повторить преобразование, при отрицательном результате осуществить изменение алгоритма либо программы.
- 8) Распечатать листинг правильно работающей программы.
- 9) Ответить на контрольные вопросы преподавателя

Контрольные вопросы.

1. Методы и типы частотного и временного преобразования.
2. Параметры частотного преобразования.
3. Факторы, влияющие на погрешность частотного преобразования.
4. Понятие разрешающая способность частотного преобразования.
5. Измерение периода.
6. Характеристика преобразования, нелинейность частотного преобразования.
7. Аппаратные реализации частотного и временного преобразования.
8. Примеры практического применения частотного и временного преобразования

Краткие теоретические сведения

В системах автоматического управления часто приходится измерять такие величины как: частота f , период T , длительность τ , сдвиг фаз φ . Для этого применяется преобразование частота-код. В зависимости от того, какой именно параметр нужно измерить применяют разный подход.

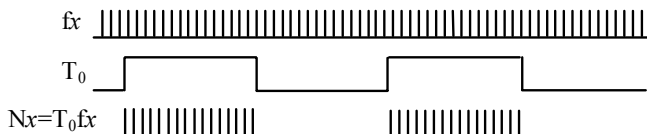


Рисунок 12.

При измерении частоты f_x производится счет импульсов входного сигнала, в течении фиксированного отрезка времени рис.12.

Такой метод приемлем для измерения высоких частот больше 100Гц. Верхний предел ограничивается быстродействием элементов схемы и разрядностью счетчиков.

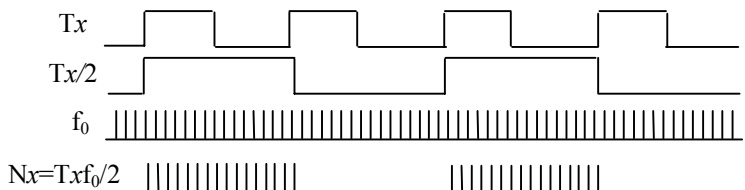


Рисунок 13. Измерение периода следования методом подсчета количества импульсов известной частоты.

Измерение частот ниже 100Гц заменяется измерением периода T_x . При этом производится подсчет импульсов фиксированной частоты f_0 за интервал равный или кратный измеряемому периоду рис.13.

Таким же образом измеряется длительность импульса τ_x , отличие состоит лишь в том, что при измерении длительности импульса не нужно делить частоту на два.

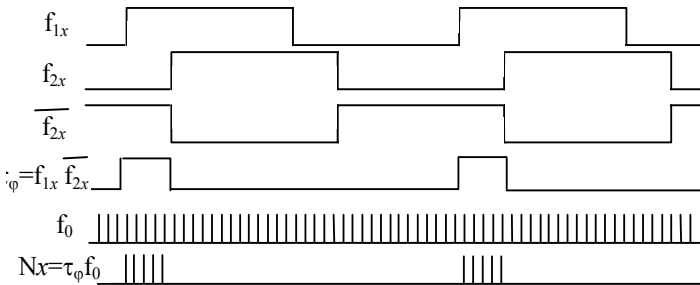


Рисунок 14. Измерение сдвига фаз

При измерении сдвига фаз временной интервал τ_ϕ формируют путем конъюнкции сигналов f_{1x} и f_{2x} рис.14, полученный интервал времени измеряют методом описанным выше.

Схемы, реализующие такой алгоритм, приведены на рисунке 15.

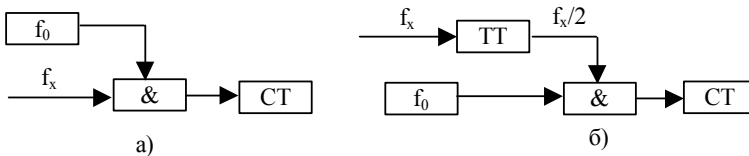


Рисунок 15. Схемы для измерения частоты а) низкой, б) высокой

В лабораторном стенде при наличии платы расширения к входу ОЭВМ T0 подключен постоянный генератор частоты, а к входу T1 подключен генератор с изменяемой частотой. Сигналы этих генераторов можно наблюдать на осциллографе, подключенном к BNC разъему. Для наблюдения сигнала T0 замкнуть перемычку J1, а для T1 перемычку J3

Таймеры/счетчики (Т/С) ОЭВМ предназначены для подсчета внешних событий, для получения программно управляемых временных задержек, выполнения времязадающих функций ОЭВМ.

В состав блока Т/С входят:

- 1) два 16-ти разрядных регистра Т/С 0, Т/С 1;
- 2) 8-ми разрядный регистр режимов Т/С (TM0D);
- 3) 8-ми разрядный регистр управления (TCON);
- 4) схема инкремента;
- 5) схема фиксации INT0, INT1, T0, T1;
- 6) схема управления флагами;
- 7) логика управления Т/С.

Два 16-разрядных регистра Т/С 0 и Т/С 1 выполняют функцию хранения содержимого счета. Каждый из них состоит из пары 8-ми

разрядных регистров, соответственно TH0, TL0 и TH1, TL1. Причем регистры TH0, TH1 - старшие, регистры TL0, TL1 - младшие 8 разрядов. Каждый из 8-ми разрядных регистров может быть использован, как POH, если T/C не используются (бит TR0 для T/C 0 и бит TR1 для T/C 1 в регистре управления TCON равен "0").

Регистр режимов T/C (TMOD) предназначен для приема и хранения кода, определяющего:

- один из 4-х возможных режимов работы каждого T/C;
- работу в качестве таймеров или счетчиков;
- управление T/C от внешнего вывода.

Регистр режимов TMOD

M0-M1(0-1,4-5) – один из 4-х режимов счётчиков

C/T(2,6) – '0' таймер '1' счётчик

GATE – разрешение управления от внешнего вывода '0' запрещено '1' разрешено (INT0, INT1)

Регистр управления (TCON) предназначен для приема и хранения кода управляющего слова. Регистр TCON нельзя использовать в качестве POH. Схема инкремента предназначена для увеличения на 1 в каждом машинном цикле содержимого регистров T/C 0, T/C 1, для которых:

- установлен режим таймера и счет разрешен;
- установлен режим счетчика, счет разрешен и на соответствующем входе схемы T0 (для T/C 0) или T1 (для T/C 1) зафиксирован счетный импульс в предыдущем машинном цикле.

Регистр управления TCON

TR – бит включения T/C '0' выключен

TF – флаг переполнения T/C

IT – определяет вид прерывания по входу INT '0' по L-уровню, '1' по фронту

IE – флаг запроса на прерывание

Пример программы измерения частоты

```

$noList
#include (c:\asm51\compile\mod51)
$list
;измерение частоты в Гц
;рассчитана на внешнюю частоту T0 6817Гц
;подсчет ведется за 1с, результат в виде XXXX Гц
    mode equ 55h ;настраивает T1 counter, T0 counter
    str equ 50h ;запись в TCON разрешает счет
    ORG 0000h
BEG: MOV TL1,#00h ;обнуление
    MOV TH1,#00h ;T/C1
    MOV TL0,#0f0h ;предустановка T/C0
    MOV TH0,#0E4h ;рассчитана на 1с
    MOV TMOD,#MODE ;инициализация режимов таймеров
    MOV TCON,#STR ;запуск счета
    JNB TR0,$ ;проверка флага переполнения T/C0
    CLR TR1 ;если произошло переполнение то
    CLR TR0 ;остановить счет T/C1 и T/C0
    MOV R0,TL1 ;считывание результата измерения
    MOV R1,TH1 ;R0 содержит младшую часть а R1 старшую
    CALL IND ;вызов подпрограммы индикации
    MOV R3,#01h ;временная задержка
M: MOV R4,#01h ;на двух регистрах
    DJNZ R4,$ ;с декрементом
    DJNZ R3,M ;во вложенном цикле
    JMP BEG ;переход на следующий цикл измерения
IND: ;подпрограмма индикации, производит перевод результата
    ;из двоичной системы в двоично-десятичную
    ;и отображение
    JMP M1
L: DB 01h,02h,04h,08h,16h,32h,64h,28h,56h,12h,24h,48h,96h
HI: DB 00h,00h,00h,00h,00h,00h,00h,01h,02h,05h,10h,20h,40h
M1: MOV R2,#00h
    MOV R6,#00h
    MOV R7,#00h
NACHALO_L: MOV A,R0
    JNB ACC.0,AGAIN_L
    MOV A,R2
    MOV DPTR,#L
    MOVC A,@A+DPTR
    ADD A,R6
    DA A
    MOV R6,A
    JNB PSW.7,PLUS_NEXT_L
    MOV A,R7
    ADD A,#01H
    DA A
    MOV R7,A
PLUS_NEXT_L: MOV A,R2
    MOV DPTR,#HI
    MOVC A,@A+DPTR
    ADD A,R7
    DA A
    MOV R7,A
AGAIN_L: MOV A,R0
    RRC A
    MOV R0,A
    INC R2
    CJNE R2,#08h,NACHALO_L
NACHALO_HI: MOV A,R1
    JNB ACC.0,AGAIN_HI
    MOV A,R2
    MOV DPTR,#L

```

```

MOV C A,@A+DPTR
ADD A,R6
DA A
MOV R6,A
JNB PSW.7,PLUS_NEXT_HI
MOV A,R7
ADD A,#01H
DA A
MOV R7,A
PLUS_NEXT_HI: MOV A,R2
MOV DPTR,#HI
MOV C A,@A+DPTR
ADD A,R7
DA A
MOV R7,A
AGAIN_HI: MOV A,R1
RRC A
MOV R1,A
INC R2
CJNE R2,#0Dh,NACHALO_HI
MOV A,R6
MOV DPTR,#0B000h ;Вывод на индикацию
MOVX @DPTR,A ;младшей части кода
MOV A,R7
MOV DPTR,#0A000h ;Вывод на индикацию
MOVX @DPTR,A ;старшей части кода
RET
END

```

Измерить частоту, переменного генератора, регулируемую резистором R4 (программа написана на языке C) .

```

int main()
{
    unsigned int i;
    xdata unsigned int freq[16];
    unsigned char f_index=0,index;

start:

    TL1=0; TH1=0; // обнулить t/c1
    TL0=0; // 7,37MHz/12 = 614400Hz 0,01sec => 6144d или 1800h
    TH0=0xe8; // предустановка t/c0 = 10000h - 1800h = e800h
    TMOD=_01010001; // c/t1-counter, c/t0-timer, mode 1 for both
    TCON=_01010000; // начали
    while(!TF0); // ожидаем переполнения t/c0 (через 0,1 сек)
    TCON=0; // стоп
    i=(TH1<<8) | TL1; // запоминаем

    if (++f_index == 16) f_index=0;
    freq[f_index] = i; // yc
    i=0; // ред
    for (index=0; index<16; index++){ // ня
        i = i + freq[f_index]; // ем
    }
    i = i >> 4;
    display_int(i,_01000000);
    delay16(20000);
    if (key_scan() == 12) goto start;
    lights_off();
}

```

Варианты индивидуальных заданий

Таблица 11 Таблица заданий к л.р. №6

| № | Текст индивидуального задания |
|----|---|
| 1 | Измерить частоту T0, отобразить на статическом индикаторе. |
| 2 | Измерить частоту T1 относительно ТФЧ, отобразить на статическом индикаторе. |
| 3 | Измерить частоту T1 относительно частоты работы процессора отобразить на статическом индикаторе. |
| 4 | Измерить разницу частот ГФЧ и ГПЧ отобразить на динамическом индикаторе. |
| 5 | С использование внутренних Т/С обеспечить плавное загорание числа 55 на статическом индикаторе. |
| 6 | Поочередно отображать 48 на втором и третьем знакоместе статической индикации и на первом и четвертом знакоместе статической индикации с интервалом 1с. Временные интервалы формировать ГФЧ |
| 7 | Подсчитать и отображать на динамическом индикаторе количество нажатий SW3 за 10с. Временные интервалы формировать Т/С ОЭВМ |
| 8 | Отображать на втором знакоместе статической индикации числа от 1 до 9 в течении 1с, и паузой между отображениями 1с. Временные интервалы формировать Т/С ОЭВМ |
| 9 | При нажатии кнопки SW4 включить секундомер с отображением на динамическом индикаторе, с использованием Т/С ОЭВМ |
| 10 | Отображать мигающие числа на четырех разрядах статической индикации с разным временем отображения паузы задавать равные 0.25, 0.5, 0.75, 1 сек. С помощью Т\С ОЭВМ. |
| 11 | Подсчитать количество нажатий кнопки SW6 за 5с. Временные интервалы формировать Т/С ОЭВМ |
| 12 | Измерить время между нажатиями кнопок SW7 и SW8.Время в сек. Отобразить на статической индикации. |
| 13 | При нажатии кнопки SW9 отобразить в секундах время ее нажатия на статической индикации. |
| 14 | Измерить период T0 отобразить на статической индикации. |
| 15 | Измерить период T1 отобразить на статической индикации. |

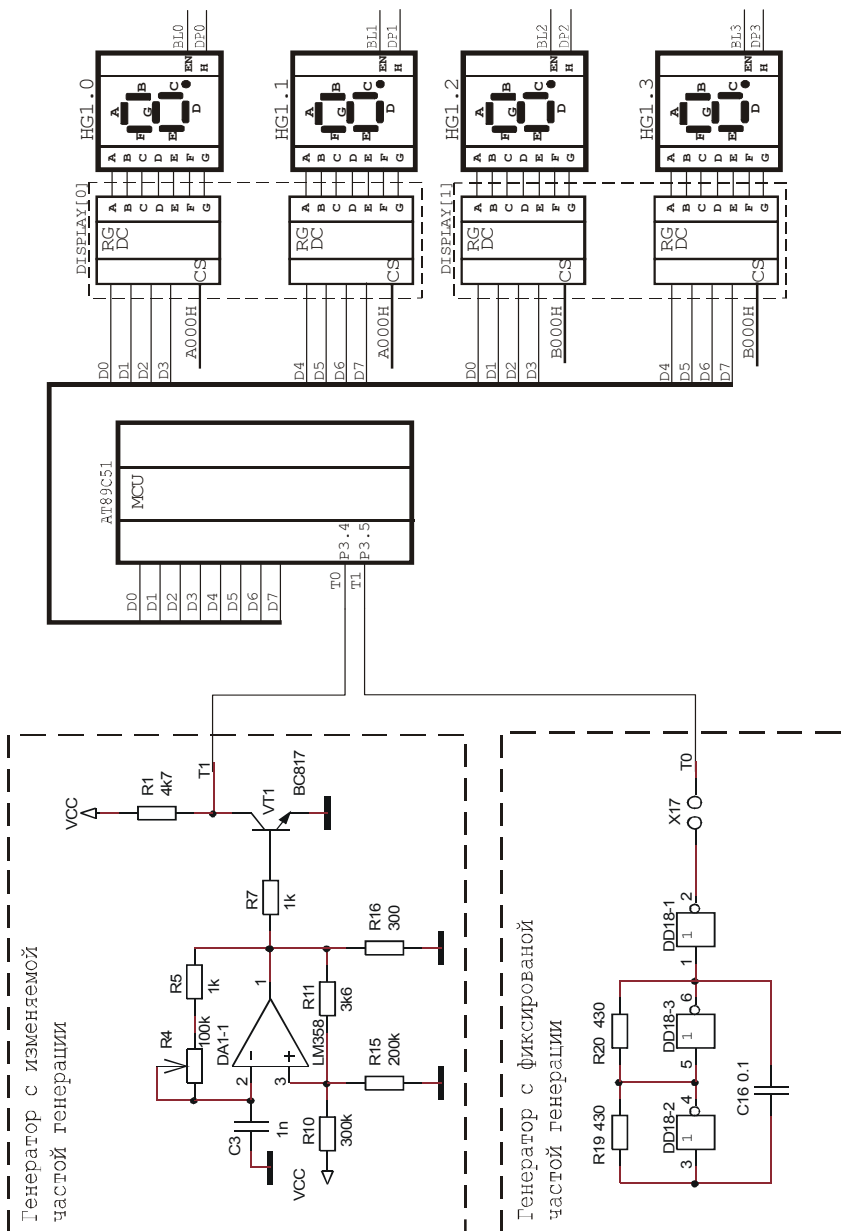


Схема к лабораторной работе №6

Лабораторная работа №7. Изучение интерфейса I2C

Тема: Изучение интегральных микросхем с последовательным вводом-выводом информации.

Цель работы: Научиться осуществлять обмен между устройствами при помощи интерфейса I2C.

Учебная задача: Разработка программы для ОЭВМ КР1816ВЕ31 формирующая I2C интерфейс.

Порядок выполнения лабораторной работы

- 1) Изучить методы последовательной передачи данных между интегральными микросхемами.
- 2) Разработать алгоритм для выполнения индивидуального задания до начала лабораторного занятия
- 3) Разработать программу для выполнения индивидуального задания до начала лабораторного занятия
- 4) Ввести программу индивидуального задания на персональном компьютере.
- 5) С помощью ПОС проанализировать выполнение индивидуальной программы
- 6) Загрузить программу в стенд ОЭВМ. Убедиться в правильном выполнении индивидуального задания, при отрицательном результате осуществить изменение алгоритма либо программы. Повторить загрузку программы в стенд ОЭВМ
- 7) Распечатать листинг правильно работающей программы.
- 8) Ответить на контрольные вопросы преподавателя

Краткие теоретические сведения

Терминология шины I²C

Передачик - элемент, который посылает данные в шину.

Приемник - элемент, который получает данные из шины.

Master - элемент, который инициализирует передачу, генерирует сигналы синхронизации, и завершает передачу.

Slave (подчиненный) - элемент, адресованный мастером

Multi-master - система с более чем одним мастером.

Арбитраж - процедура, гарантирующая, что, если больше чем один мастер одновременно пробуют управлять шиной, только один из них получит полное управление шиной и обеспечит передачу данных без ошибок.

Синхронизация - синхронизация передачи данных между элементами.

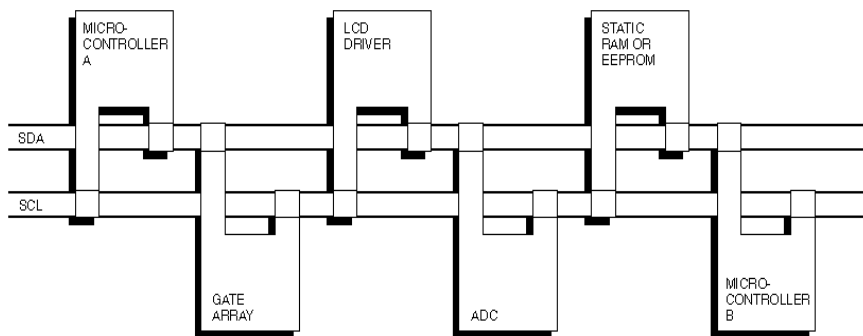


Рисунок 16. Пример конфигурации шины I²C, использующей два микроконтроллера

Общие характеристики

SDA и SCL - двунаправленные линии с открытым стоком (коллектором), соединенные с положительным питающим напряжением через резистор (см. Рисунок 17). Когда шина свободна, обе линии находятся в состоянии “1”. Данные могут передаваться по шине I²C со скоростью до 100 kbit/s в стандартном режиме, или до 400 kbit/s в быстром режиме. Число интерфейсов, соединенных с шиной исключительно зависит от емкости шины, максимальное значение которой - 400pF.

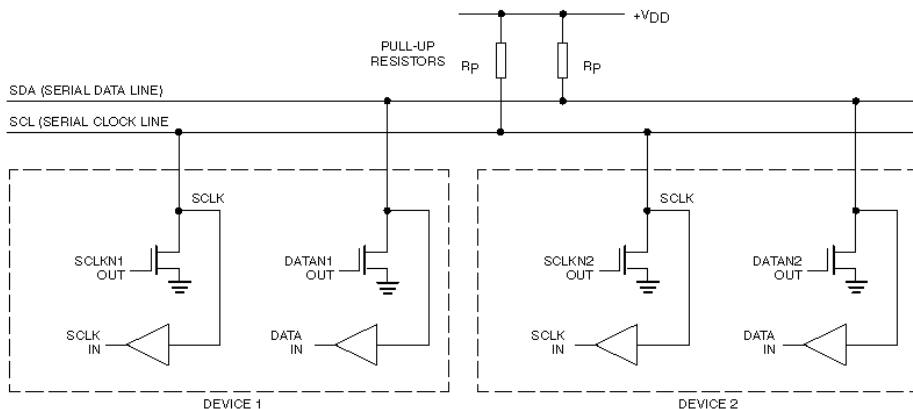


Рисунок 17. Соединение I²C устройств

Передача бита

Из-за разнообразия различных технологий (CMOS, NMOS, биполярные приборы) которые могут быть использованы с шиной I²C, уровни логического “0” и “1” не установлены и зависят от величины питающего

напряжения VDD. Для передачи одного бита данных используется один импульс сигнала синхронизации, при этом уровень на линии SDA должен быть неизменным в течение высокого уровня на линии SCL, и может изменяться только при низком уровне на SCL (Рисунок 18). Исключениями служат два особых состояния - START и STOP.

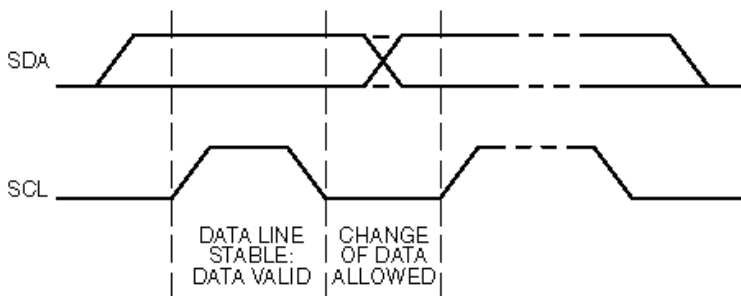


Рисунок 18. Передача бита.

START и STOP.

Существуют два особых состояния шины I²C - start и stop, которые служат для индикации начала и конца передачи и соответственно перехода шины в неактивное состояние. Следует отметить, что до тех пор, пока не установлено состояние start, сигналы на линиях SDA и SCL могут быть совершенно произвольными (рисунок 19). Это позволяет, в частности, использовать одну линию SDA и несколько линий SCL (например, при нехватке, адресов на одной шине).

Состояние START - переход от “1” к “0” на линии SDA при “1” на линии SCL.

Состояние STOP - переход от “0” к “1” на линии SDA при “1” на линии SCL.

Эти два состояния всегда генерируются мастером.

Детектирование состояний start и stop в специализированных элементах обычно производится аппаратно. При полностью программной реализации шины I²C в микроконтроллерах без аппаратной I²C-части необходимо как минимум 2 раза проверять состояние линии SDA.

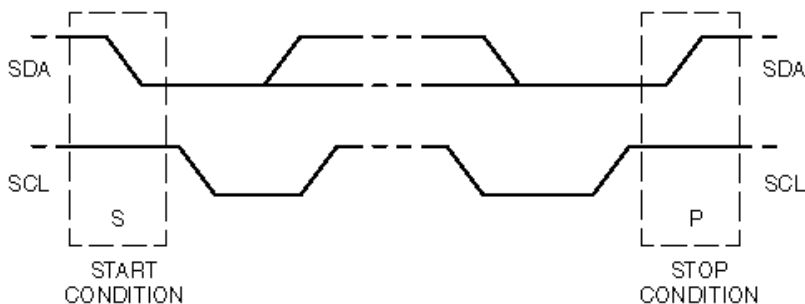


Рисунок 19. START и STOP состояния.

Передача данных

Все передачи производятся 8-разрядными байтами. Число байтов, которые могут быть переданы за одну передачу не ограничено. Каждый байт должен сопровождаться битом подтверждения (ACK). Данные передаются начиная со старшего бита (MSB), (рисунок 19).

Если приемник не может получать другой полный байт данных, он не выдает сигнал ACK, который используется передатчиком для синхронизации или сигнализации о неисправности приемника (или его отсутствии).

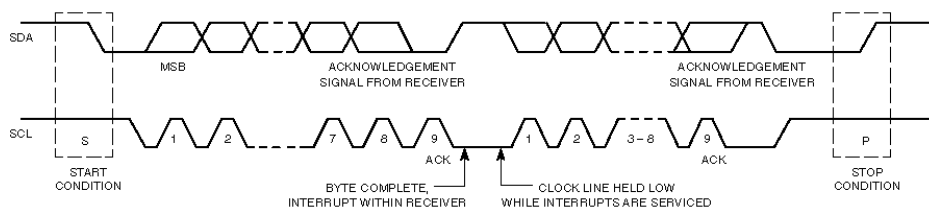


Рисунок 20. Передача данных по шине I²C.

ACKNOWLEDGE

Для подтверждения передачи байта передатчик устанавливает линию SDA в "1" в течение синхронизирующего импульса. Приемник при этом должен выставить "0" на SDA (рисунок 21). Естественно при этом должны быть выдержаны определенные временные соотношения.

Обычно, приемник, который был адресован, должен генерировать сигнал ACK после того, как каждый байт был получен, за исключением того, когда сообщение начинается с адресом CBUS.

Когда подчиненный приемник не подтверждает подчиненный адрес (например, устройство не готово, потому что это выполняет некоторую внутреннюю функцию), линия SDA данных должна быть оставлена в "1". Мастер затем может выдать состояние STOP, чтобы прервать передачу.

Если подчиненный приемник подтверждает подчиненный адрес, но, некоторое время позднее не может получить больше байты данных, мастер должен приостановить передачу.

При приеме последнего байта в серии, вместо сигнала ACK мастер может выставить состояние STOP, при этом подчиненный передатчик должен освободить линию данных.

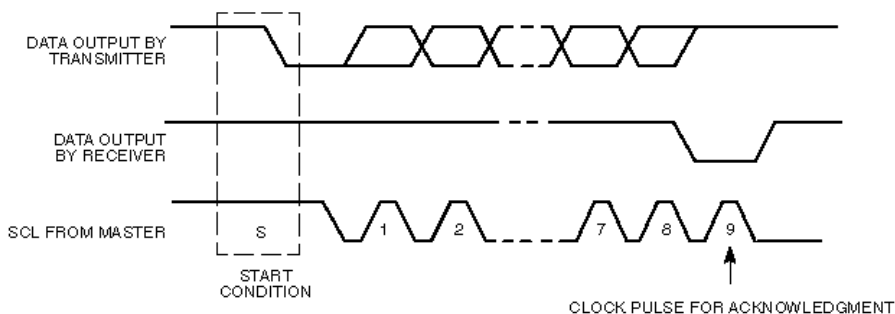


Рисунок 21. Подтверждение передачи.

Контрольные вопросы

1. Назначение и сфера применения интегральных устройств с последовательным интерфейсом
2. Временные параметры сигналов по интерфейсу I2c.
3. Протокол обмена по I2c
4. Назначение внешних адресных линий на микросхемах с I2c.
5. Использование нескольких устройств на одном интерфейсе.
6. Методы адресации различных устройств с I2c

Лабораторная работа №8. Программирование последовательного порта ОЭВМ

Тема: Программирование последовательного порта ОЭВМ.

Цель работы: Научиться осуществлять передачу данных по интерфейсу RS232.

Учебная задача: Разработка программы для ОЭВМ КР1816ВЕ31 на базе учебного стенда для связи и передачи информации от стенда на персональный компьютер и обратно и связи двух учебных стендов меж собой с помощью интерфейса RS232

Порядок выполнения лабораторной работы

1) Изучить методы последовательной передачи данных между периферийными устройствами.

2) Разработать алгоритм и программу для выполнения индивидуального задания до начало лабораторного занятия

3) Ввести программу индивидуального задания на персональном компьютере.

4) Загрузить программу в стенд ОЭВМ. Убедиться в правильном выполнении индивидуального задания, при отрицательном результате осуществить изменение алгоритма либо программы. Повторить загрузку программы в стенд ОЭВМ

5) Распечатать листинг правильно работающей программы.

6) Ответить на контрольные вопросы преподавателя

Контрольные вопросы

1. Назначение и сфера применения передачи данных по последовательному интерфейсу

2. Физическая характеристика сигналов в интерфейсе RS232.

3. Временные параметры, определение скорости обмена интерфейсу RS232.

4. Протокол обмена интерфейсу RS232.

5. Синхронные и асинхронные каналы последовательной передачи данных.

6. Регистры T/C используемые при инициализации последовательного порта ОЭВМ.

7. Адресация последовательных портов PC

Перечень литературы

- 1.Сопряжение датчиков и устройств ввода данных с компьютерами IBM PC п/р У.Томпкинса и Дж. Уэбстера, Москва, "Мир",1992
- 2.Быстродействующие интегральные микросхемы ЦАП и АЦП п/р А.К.Марцинкявичуса, Москва, "Радио и связь",1990
- 3.Основы автоматики, импульсной и вычислительной техники П.М.Грицевский, А.Е.Мамченко, Москва, "радио",1989
- 4.Интегральная электроника в измерительных устройствах В.С.Гутников Ленинград "Энегроатомиздат",1990
- 5.В.И.Зубчук Справочник по цифровой схемотехнике, Киев "Тэхника"1990
- 6.С.В.Якубовский "Цифровые и аналоговые интегральные микросхемы" Справочник. Москва "Радио и связь",1990
- 7.В.П.Шило "Популярные цифровые микросхемы" Москва "Радио и связь" 1988
- 8.Ф.В.Кушнир "Радиотехнические измерения" Москва "Связь" 1986
- 10.Абрайтис В.Б. "Быстродействующие интегральные микросхемы " Электронная промышленность, 1989
11. Шевкопляс Б.В. Микропроцессорные структуры , "Радио и связь" 1990
- 12.Г. Виглеб. Датчики. М.: Мир, 1989.
- 13.Г.П.Липовецкий. Однокристалльные микроЭВМ. М.: МП "БИНОМ",1992.
- 14.С.Т. Хвощ. Микропроцессоры и микроЭВМ в системах автоматизации и управления. Справочник. М.: Машиностроение, 1987.
- 15.В.С. Гутников. Интегральная электроника в измерительных устройствах. Л.: Энергоатомиздат, 1988.
- 16.Аналоговые интегральные схемы: Справ./ А.Л.Булычёв. В.И.Галкин, В.А.Прохоренко.- Мн.: Беларусь, 1993.- 382с.
- 17.Однокристалльные микро-ЭВМ. Техническое описание и руководство по применению / Г.П. Литвинский., Москва , 1982.
- 18.Ю.В. Новиков Разработка устройств сопряжения

