

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Методические указания

к выполнению курсового проекта по дисциплине
«Системотехнические методы и средства отображения
информации» для студентов специальности 6.050802 ЭлС
(Электронные системы)

**Рассмотрено
на заседании кафедры
электронной техники.
Протокол №3 от 17.10.2010г
Утверждено на заседании**

учебно-издательского совета

**ДОННТУ протокол №263
Пр. №2 от 12. 12. 2010г**

Донецк, 2010

ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Разработать микропроцессорную систему (МПС) контроля технологических процессов. Варианты заданий на проектирование аппаратной части системы приведены в таблице 1. Варианты заданий на разработку программного обеспечения (ПО) приведены в таблице 2.

Курсовой проект оформляется в виде пояснительной записки (ПЗ). ПЗ должна содержать:

- титульный лист;
- лист задания;
- реферат;
- перечень сокращений;
- содержание;
- введение;
- текст ПЗ;
- выводы;
- перечень ссылок;
- приложения.

Текст ПЗ должен включать в себя следующие пункты:

1. Разработка структурной схемы МПС.
2. Выбор элементной базы.
3. Разработка функциональной схемы МПС и словесного алгоритма ее функционирования.
4. Разработка принципиальной схемы МПС.
5. Разработка ПО.
 - 5.1. Разработка блок-схем ПО.
 - 5.2. Разработка программы.
6. Имитационное моделирование МПС в пакете Proteus.

Структурная, функциональная и принципиальная схемы МПС, а также спецификация элементов и листинг ПО оформляются по ГОСТ и выносятся в приложение.

Структурная схема МПС должна отражать структурные элементы системы и связи между ними.

При **выборе элементной базы** необходимо кратко обосновать выбор микросхем, указать их назначение, принцип работы, назначение выводов, привести условное графическое обозначение микросхемы на принципиальной схеме и ее основные технические характеристики.

Функциональная схема МПС отличается от принципиальной отсутствием линий питания, блокировочных конденсаторов, разъемов и прочих элементов, не отражающих принцип функционирования системы. Для улучшения читабельности однотипные сигналы на функциональной схеме (ФС) необходимо объединить в шины. Именно функциональная схема отражает принцип функционирования системы. Ее разработку следует разбить на этапы. К примеру, разработка ФС узла центрального процессора, узла ОЗУ, узла индикации, узла клавиатуры, узла АЦП/ЦАП, узла последовательного канала ввода/вывода, узла датчиков. Далее будет рассмотрен пример разработки ФС МПС.

Блок-схемы ПО составляются для каждой подпрограммы и не должны быть слишком простыми или слишком сложными. Рекомендуемое количество блоков от 5 до 20 на подпрограмму. Листинг программы выносится в приложение. Пояснительная записка должна содержать текстовую часть с описанием структуры ПО, а также описанием назначения и функций каждой подпрограммы.

При составлении **имитационной модели** принципиальной схемы МПС необходимо учитывать, что элементная база пакета Proteus (как и любого другого пакета моделиро-

вания) имеет ограниченный объем. Несмотря на то, что данная программа позволяет пополнять библиотеки новыми моделями, созданными фирмами-разработчиками

Таблица вариантов заданий на курсовой проект

Вариант	МК	АЦП	ЦАП	ОЗУ	RS232	Матрица на светодиодах	Кнопки	ЖКИ	Семисегм.	N1	N2
1	AT89C52	8р.		4кБ	+	6симв.	2	+	3с	4	4
2	AT89C55	10р.	8р.	16кБ	+		3	+	4ж	4	8
3	AT89S52	12р.		8кБ	+	4симв.	4	+	3ж	8	4
4	AT89S8252	16р.	10р.	16кБ	+		6	+	4с	8	8
5	AT89S53	8р.		32кБ	+	8симв.	9	+	3ж	4	4
6	AT89S8253	10р.	12р.	16кБ	+		12	+	4с	4	8
7	AT89C51RB2	12р.		4кБ	+	4симв.	16	+	5с	8	4
8	AT89C51RC2	16р.	16р.	16кБ	+		2	+	3ж	8	8
9	AT89C51IC2	8р.		8кБ	+	6симв.	3	+	4с	4	4
10	AT89C51AC2	10р.	8р.	16кБ	+		4	+	5ж	4	8
11	AT89C51AC3	12р.		32кБ	+	4симв.	6	+	4ж	8	4
12	AT89C51RD2	16р.	10р.	16кБ	+		9	+	5с	8	8
13	AT89C51ED2	8р.		4кБ	+	6симв.	12	+	3с	4	4
14	AT89C51ID2	10р.	12р.	16кБ	+		16	+	5ж	4	8
15	AT89C55WD	12р.		8кБ	+	8симв.	2	+	3ж	8	4
16	AT89C51RC	16р.	16р.	16кБ	+		3	+	4с	8	8
17	AT89C52	8р.		32кБ	+	4симв.	4	+	4ж	4	4
18	AT89C55	10р.	8р.	16кБ	+		6	+	3ж	4	8
19	AT89S52	12р.		4кБ	+	4симв.	9	+	5ж	8	4
20	AT89S8252	16р.	10р.	16кБ	+		12	+	3с	8	8

Функции программного обеспечения

1. Циклический опрос АЦП с периодом $T_{сек}$ N –раз и запись в оперативную память.
2. Циклический опрос дискретных датчиков и запись их в оперативную память (ОП).
3. Обработка измеренных кодов.
 - 3.1. Поиск максимального значения (MAX) среди измеренных кодов в ОП.
 - 3.2. Поиск минимального значения (MIN) среди измеренных кодов в ОП.
 - 3.3. Подсчет среднего арифметического значения измеренных кодов в ОП.
4. Выдача содержимого ОП и результата обработки на LCD.
5. Выдача результата обработки на цифровую индикацию.
6. Вывод кода результата обработки на ЦАП.
7. Выдача содержимого ОП и результата обработки на UART.
8. Ввод с кнопок.
 - 8.1. Нажатие кнопки 1 порождает циклический опрос АЦП.
 - 8.2. Нажатие кнопки 2 порождает циклический опрос дискретных датчиков.
 - 8.3. Нажатие кнопки 3 порождает цикл передачи на UART.
9. Вывод дискретного кода с двоичной индикацией.

Задания на разработку ПО для МПС приведено в таблице 2.

Задания по вариантам представлено в виде набора функций ПО для разработки МПС.

Таблица 2. Варианты заданий на разработку программного обеспечения

Вариант	Период опроса T,с	Последовательность заданных функций ПО
1	0,3	1, 3.1, 4, 5, 8;
2	0,5	1, 3.1, 4, 6, 8;
3	0,75	1, 3.2, 4, 5, 8;
4	1,0	1, 3.3, 4, 6, 8;
5	1,25	1, 2, 4, 7, 8;
6	1,5	1, 2, 4, 8, 9;
7	2,0	1, 2, 4, 7, 8;
8	0,3	1,3.3, 4, 7;
9	0,5	1, 2, 4, 7;
10	0,75	1, 2, 4, 7, 9;
11	1,0	1, 3.1, 4, 8;
12	1,25	1, 3.2, 4, 8;
13	1,5	1, 3.3, 4, 5, 8;
14	2,0	1, 2, 4, 6, 8;
15	0,3	1, 2, 4, 8;
16	0,5	1, 2, 3.1, 4, 7, 8;
17	0,75	1, 2, 3.3, 4, 5, 8;
18	1,0	1, 2, 3.2, 4, 8, 9;
19	1,25	1, 2, 3.3, 4, 9;
20	1,5	1, 2, 4, 5, 8;

электронных компонентов, или моделями, составленными собственными силами, рекомендуется использовать элементы из категории **Modelling Primitives**. Эта категория содержит модели базовых элементов электроники и носит обобщающий характер. Модели здесь максимально упрощены, что облегчает процесс работы с ними. Однако они не имеют физических аналогов и поэтому выводы элементов из данной категории не пронумерованы. При оформлении принципиальных схем в курсовом проекте *допускается оставлять без нумерации выводы элементов из категории Modelling Primitives*. В частности, из данной категории рекомендуется взять ОЗУ, дешифратор шины адреса, АЦП, ЦАП. Отметим, что необходимая информация по компонентам содержится в хэлпе на соответствующий элемент в *Proteus*.

ПРИМЕР ПОСТРОЕНИЯ ФУНКЦИОНАЛЬНОЙ СХЕМЫ МПС

Исходные данные: АЦП – 16р.; ОЗУ – 4 кБ; индикатор – ЖКИ; дискретные линии ввода/вывода – 4/4.

Выбор элементной базы опускается.

Разработка ФС узла центрального процессора.

ФС узла центрального процессора (ЦП) представлена на рис.1. Узел ЦП формирует системную шину, которая обеспечивает взаимодействие всех узлов МПС. Системная шина состоит из 16-разрядной шины адреса A0...A15, 8-разрядной шины данных AD0...AD7 и шины управления. В состав узла ЦП входит микроконтроллер DD1, регистр-защелка

младшего байта адреса DD2, дешифратор внешнего адресного пространства DD3, формирователь сигнала сброса DD4 и инверторы сигналов чтения/записи из внешних устройств DD5.1 и DD5.2.

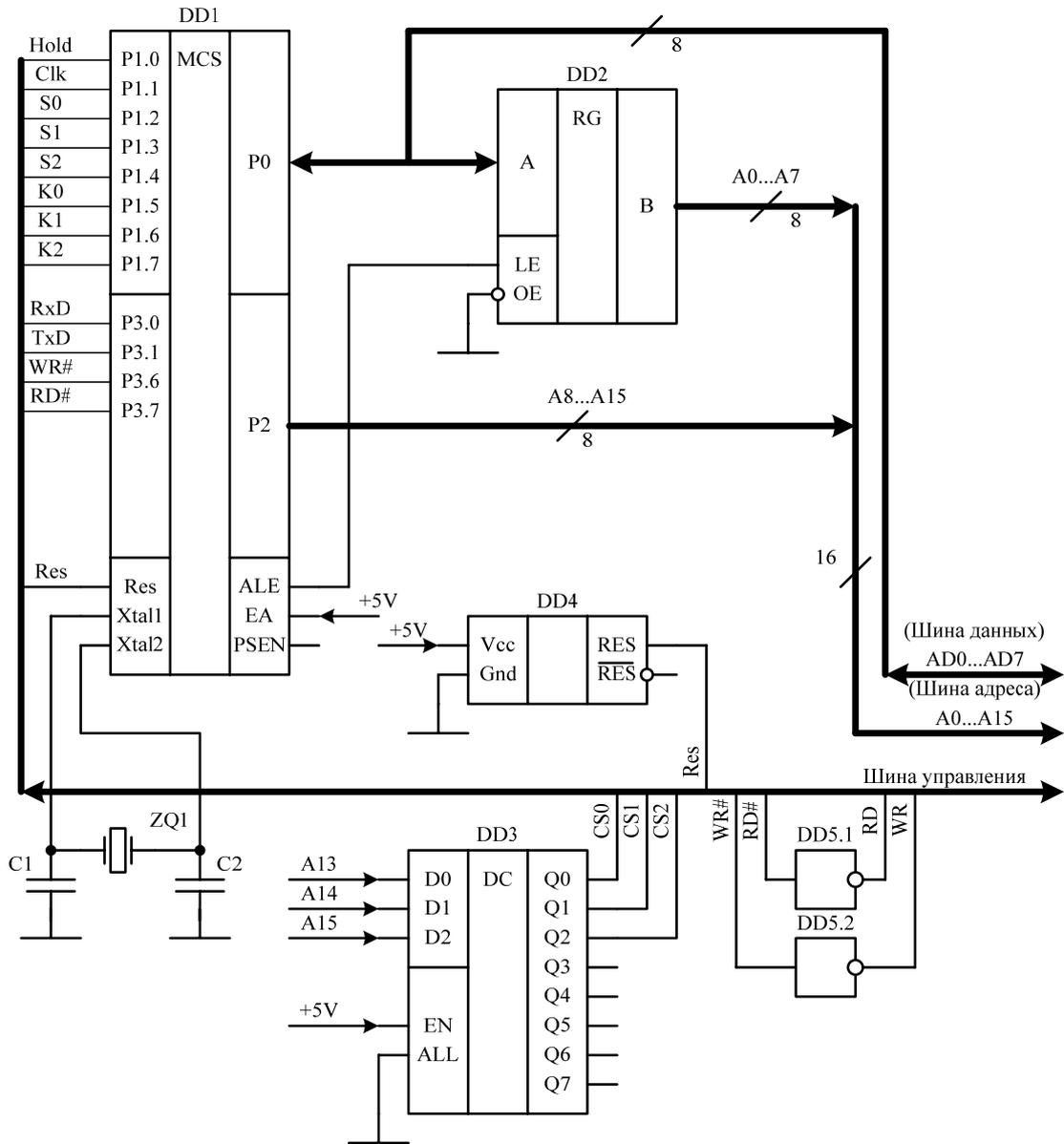


Рис.1 ФС узла центрального процессора

Дешифратор DD3 обеспечивает разбиение внешнего адресного пространства на сегменты в соответствии с рис.2.

A15	A14	A13	Сегмент адресуемой памяти	Адрес
1	1	1	64 кБ	0xFFFF 0xE000
1	1	0	56 кБ	0xDFFF 0xC000
1	0	1	48 кБ	0xBFFF 0xA000
1	0	0	40 кБ	0x9FFF 0x8000
0	1	1	32 кБ	0x7FFF 0x6000
0	1	0	24 кБ	0x5FFF 0x4000
0	0	1	16 кБ	0x3FFF 0x2000
0	0	0	8 кБ	0x1FFF 0x0000

Рис.2 Схема разбиения адресного пространства МПС

Разработка ФС узла программируемого периферийного адаптера.

ФС узла программируемого периферийного адаптера (ППА) 8255 представлена на рис.3. ППА используется для расширения возможностей параллельного ввода-вывода и содержит 3 двунаправленные 8-разрядные порта: PA, PB и PC.

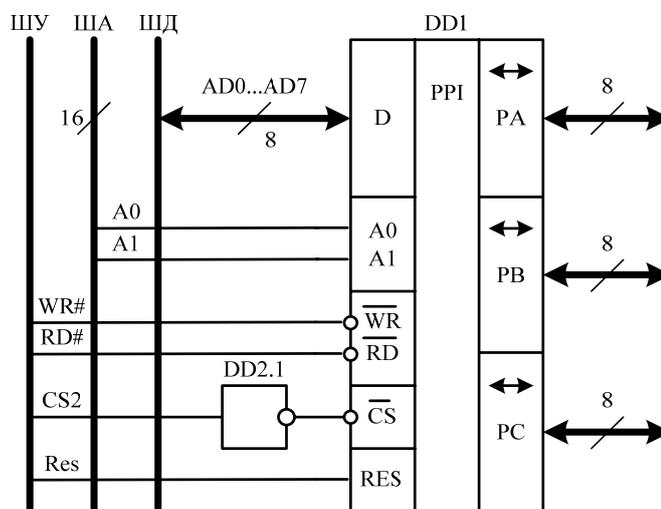


Рис.3 ФС узла программируемого периферийного адаптера 8255

Выбранное подключение ППА к системной шине (сигналы CS2, A0 и A1) обеспечивает привязку его внутренних регистров к определенным адресам, а именно: PA – 0x2000; PB – 0x2001; PC – 0x2002; регистр настройки ППА – 0x2003.

Разработка ФС узла внешнего ОЗУ.

ФС узла внешнего ОЗУ приведена на рис.4.

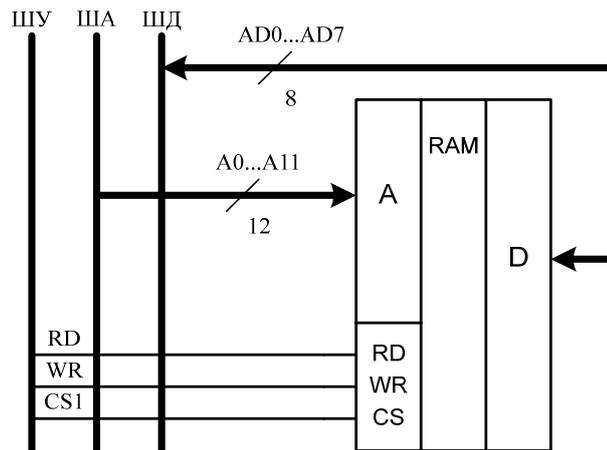


Рис.4 ФС узла внешнего ОЗУ

Разработка ФС узла индикации.

ФС узла индикации приведена на рис.5. Выбранное подключение ЖКИ к систем-ной шине (сигналы CS3, A0 и A1) обеспечивает привязку его внутренних регистров к оп-ределенным адресам, а именно: управляющий регистр – 0x4000; регистр счетчика адреса – 0x4001; регистр записи данных – 0x4002; регистр чтения данных – 0x4003.

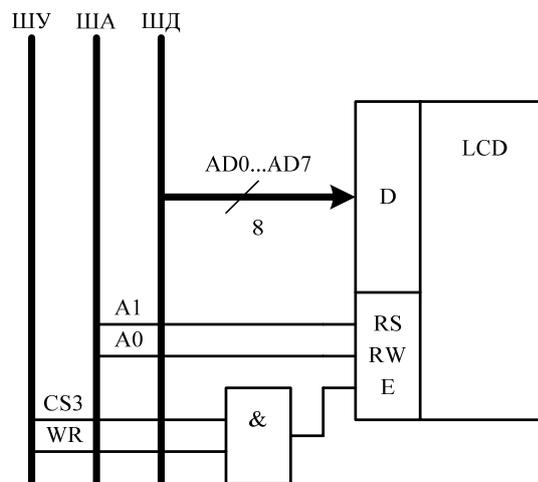


Рис.5 ФС узла индикации

Разработка ФС узла АЦП, датчиков и дискретных линий управления.

ФС узла АЦП и дискретных линий ввода/вывода приведена на рис.6. Младшие 8 линий данных АЦП подключены к порту РА ППА, старшие – к РВ. 4 младшие бита порта РС ППА сконфигурированы на ввод и к ним подключаются датчики дискретных сигналов, старшие биты РС настроены на вывод и составляют линии дискретного вывода. Работой АЦП управляют сигналы *Hold* и *Clk*.

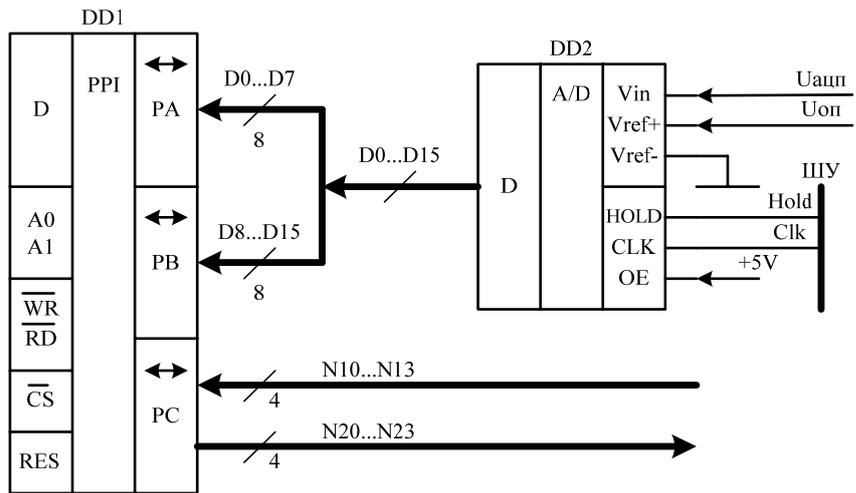


Рис.6 ФС узла АЦП и дискретных линий ввода/вывода

Полная ФС МПС представлена на рис.7.

Имитационная модель МПС приведена на рис.8.

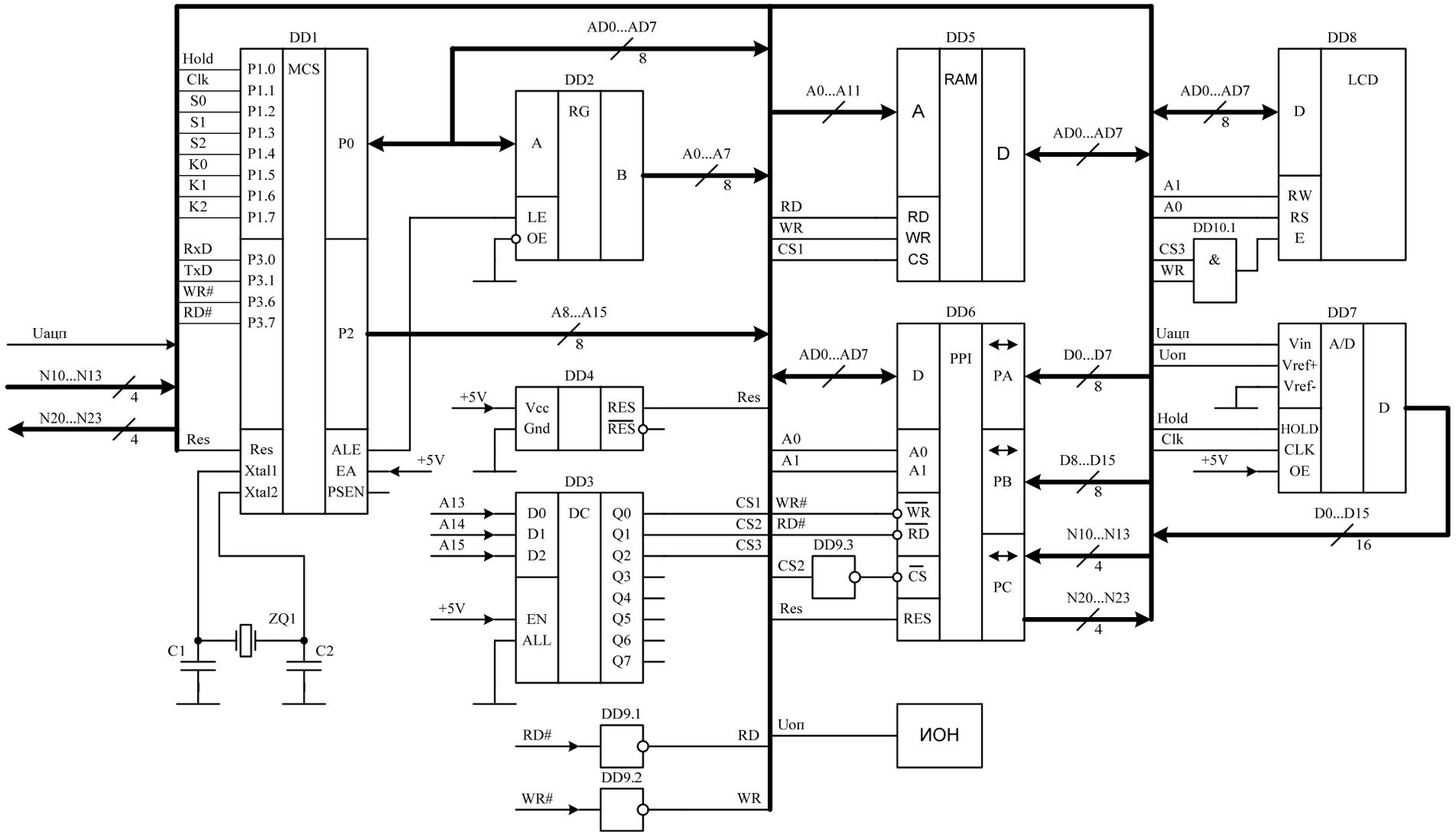


Рис.7 Функциональная схема МПС

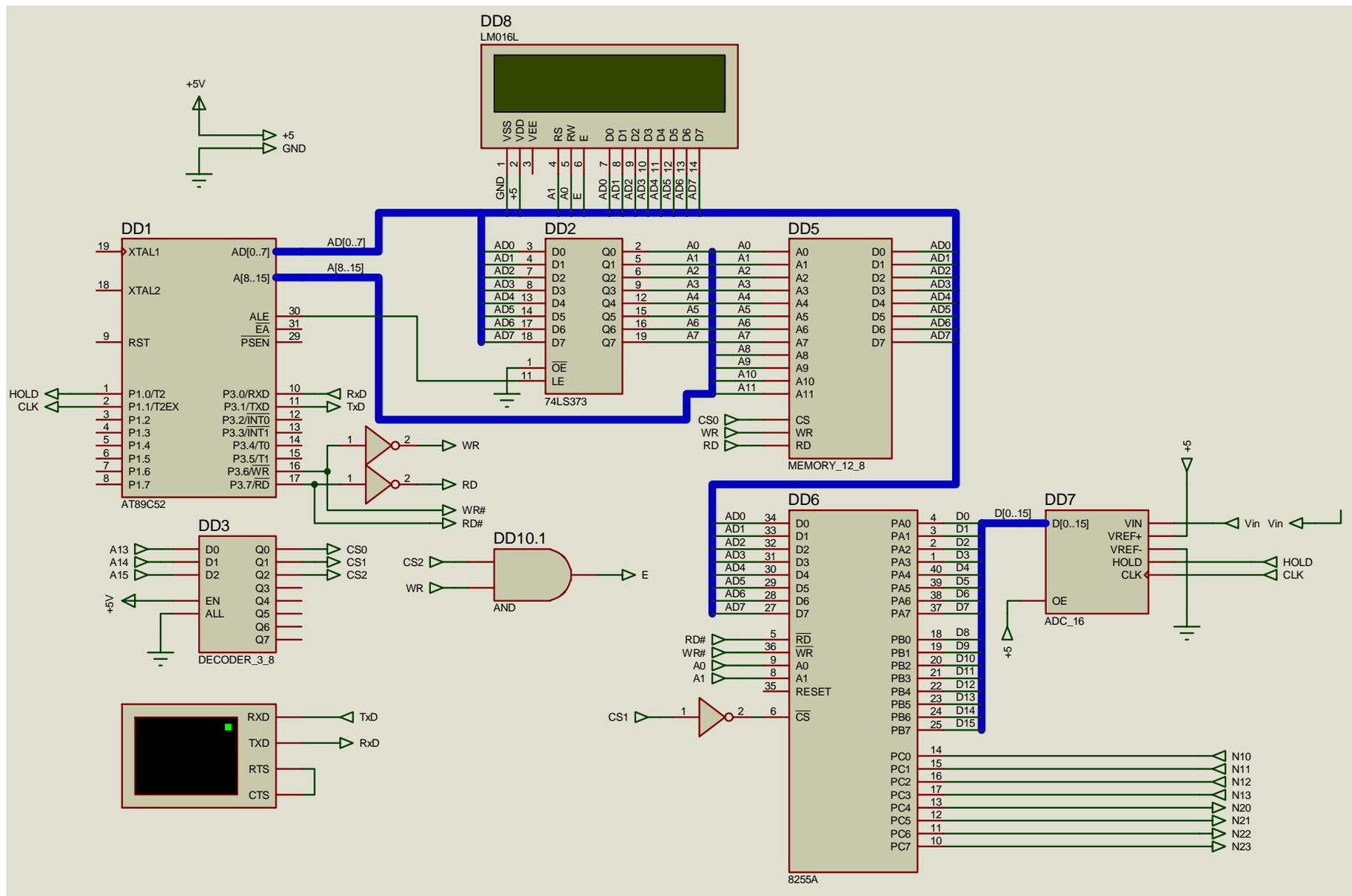


Рис.8 Имитационная модель МПС


```

//Вывод символа с кодом kod на ЖКИ в позицию с адресом adr *****
void OutSym(unsigned char adr,unsigned char kod)
{
    GKICon=(0x80 | adr);
    pausa(1);
    GKIWr=kod;
    pausa(1);
}

//Инициализация ЖКИ *****
void GKI_init()
{
    pausa(15);
    GKICon=0x30;
    pausa(5);
    GKICon=0x30;
    pausa(1);
    GKICon=0x30;
    pausa(40);
    GKICon=0x38;
    pausa(40);
    GKICon=0x38;
    pausa(40);
    GKICon=0x01;
    pausa(2);
    GKICon=0x04;
    pausa(5);
    GKICon=0x0C;
    pausa(5);
}

//Обработчик прерываний от T0 *****
//Опрос АЦП и выдача кода АЦП по UART
void timer_T0() interrupt 1
{
    N_T0++;//Инкремент счетчика переполнений T0
    display=N_T0;
    if (N_T0==10)
        {
            N_T0=0;//Сброс счетчика
            ConvADC();//Выполнение преобразования АЦП
            printf ("kodADC=%lu\n", kodADC);//Выдали код АЦП по UART
        }
}

//Основная системная функция *****
void main()
{
    unsigned char i;
    unsigned char S[]="123456789ABCDEF";//Строка вывода на ЖКИ
    //Инициализация сигналов управления АЦП
        HOLD=0;
        clk=0;
    //Настройка UART
        SCON=0x50; //Режим 1 UART, прием РАЗРЕШЕН
        RCAP2H=0xFD; //Данные для
        RCAP2L=0x8F; //перезагрузки таймера T2 (1200 Бод) Fosc=24.000 МГц
        T2CON=0x34; //Настроили T2 и запустили
        TI=1;
    //Настройка таймера
        TMOD=0x01;
    //Настройка системы прерываний
        ET0=1;//Разрешили прерывания от T0
}

```

```

EA=1;//Общее разрешение прерываний
//Настройка ППА
PPACon=0b10010010;
GKI_init();//Инициализация ЖКИ
for (i=0;i<16;i++) OutSym(i,S[i]);//Вывод строки S
TR0=1;//Пуск T0
PC=0x0F;
display=0xF0;//Зажигаем все сегменты семисегментного индикатора
while(1);//Зацикливание
}

```

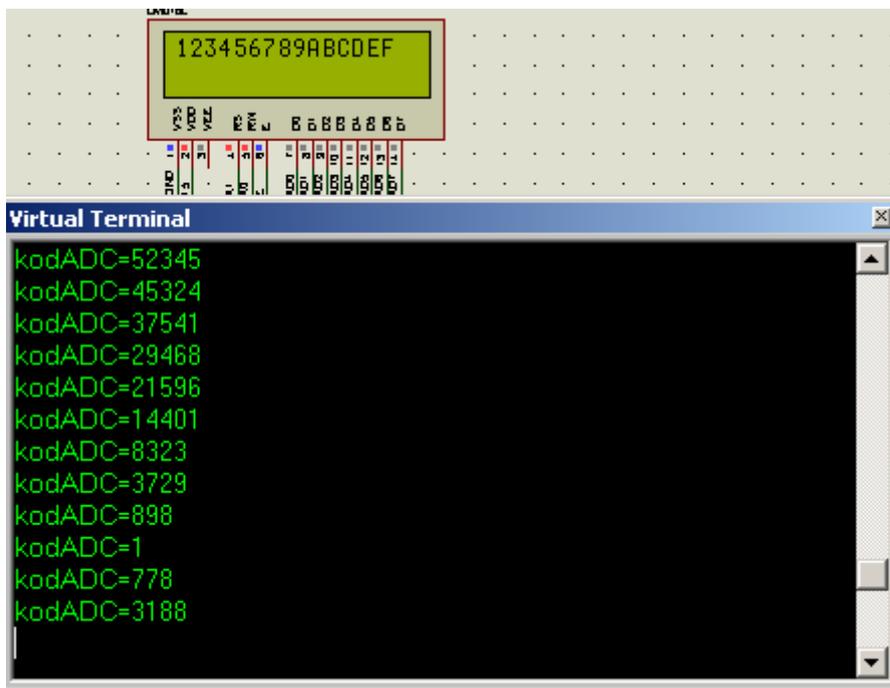


Рис.9 Результаты работы программы

ДОПОЛНИТЕЛЬНЫЕ МАТЕРИАЛЫ

Программируемый периферийный адаптер КР580ВВ55

(отечественный аналог INTEL 8255; старое наименование – К580ИК55)

Программируемый периферийный адаптер (ППА) КР580ВВ55 представляет собой три канала ввода/вывода А, В и С по 8 бит каждый. Существует три режима работы микросхемы. В режиме 0 три упомянутых канала могут быть независимо друг от друга запрограммированы на ввод или вывод информации. Режим 1 позволяет передавать информацию по каналам А и В, используя отдельные биты канала С для управления обменом данными. В режиме 2 канал А приобретает свойство двунаправленной шины, управляемой опять же отдельными битами канала С. Кроме того, возможны установка и сброс отдельных битов канала С. Микросхема выпускается в 40-выводном DIP-корпусе.

Условное графическое обозначение (УГО) ППА приведено на рис.8.

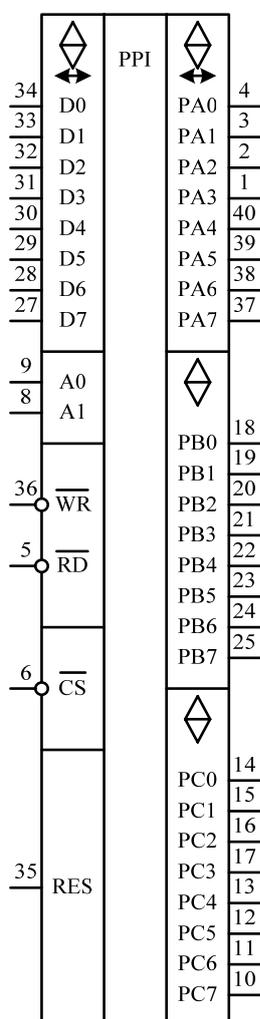


Рис.8а. УГО ППА

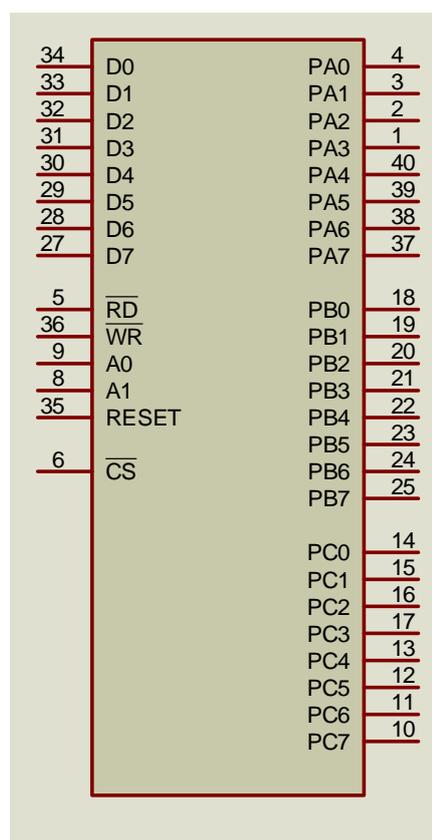


Рис.8б. УГО ППА в Proteus

Основные особенности микросхемы:

- 3 канала ввода/вывода, 8 бит каждый;
- 3 режима работы (0 – статический ввод/вывод, 1 – тактируемый ввод/вывод, 2 – двунаправленный ввод/вывод);
- раздельное программирование каждого канала на ввод или вывод.

ППА структурно разбит на две группы (А и В). Группа А включает в себя канал А и старшую тетраду канала С. Группа В состоит из канала В и младшей тетрады канала С. Доступ к каналам ввода/вывода и регистру управления осуществляется с помощью логики чтения-записи в соответствии с табл. 3.

Табл. 3.

Доступ к каналам ввода/вывода и регистру управляющего слова

A1	A0	/RD	/WR	/CS	
0	0	0	1	0	Шина данных <- канал А
0	1	0	1	0	Шина данных <- канал В
1	0	0	1	0	Шина данных <- канал С
1	1	0	1	0	Данные не определены
0	0	1	0	0	Шина данных -> канал А
0	1	1	0	0	Шина данных -> канал В
1	0	1	0	0	Шина данных -> канал С
1	1	1	0	0	Шина данных -> регистр управления
X	X	X	X	1	Шина данных в высокоомном состоянии

Назначение этих и остальных выводов микросхемы приведено в табл. 4.

Табл. 4

Назначение выводов

D0...D7	Двунаправленная шина данных. Предназначена для передачи и приема данных процессором, а также передачи управляющих команд и слова состояния
/CS	Chip Select. Выбор микросхемы. Низкий уровень инициирует обмен между процессором и ППА
/RD	Read. Чтение. Низкий уровень на этом входе позволяет ППА передать данные или слово состояния процессору
/WR	Write. Запись. Низкий уровень на этом входе позволяет ППА принять данные или управляющую команду
RESET	Reset. Сброс. Высокий уровень на этом входе сбрасывает все внутренние ре-

	гистры, включая регистры управления, переключает все каналы (А, В, С) в режим 0, одновременно переводя их в режим ввода (только при /RD = 0 и /WR = 0)
A0, A1	Адресные входы. Выбор канала. Позволяют выбрать один из каналов (А, В, С) или регистры управляющего слова и состояния (в зависимости от сигналов на входах /RD и /WR, см. табл. 2)
PA0...PA7	Канал А
PB0...PB7	Канал В
PC0...PC7	Канал С

Первоначальное включение или подача сигнала высокого уровня на вход сброс RESET микросхемы переключает все каналы (А, В, С) в режим 0, одновременно переводя их в режим ввода.

Режим работы и направление передачи для каждого канала задается с помощью записи управляющего слова в регистр управления в соответствии с табл.5.

Табл.5

Формат управляющего слова для выбора режима работы

D7	D6	D5	D4	D3	D2	D1	D0
Флаг управления 1 = выбор режима	Группа А			Группа В			
	Выбор режима: 00 = режим 0 01 = режим 1 1x = режим 2	Канал А: 1 = ввод 0 = вывод	Старшая тетрада канала С: 1 = ввод 0 = вывод	Выбор режима: 0 = режим 0 1 = режим 1	Канал В: 1 = ввод 0 = вывод	Младшая тетрада канала С: 1 = ввод 0 = вывод	

Необходимо отметить, что использование режимов 1 и 2 подразумевает относительно "разумную" реакцию внешних устройств на сигналы ППА, что сопряжено с усложненной схмотехникой последних. В большинстве же случаев можно обойтись режимом 0 простого ввода/вывода.

ИНТЕРФЕЙСЫ RS232/RS485

Интерфейс RS232

Широко используемый последовательный интерфейс синхронной и асинхронной передачи данных, определяемый стандартом EIA RS-232-C и рекомендациями V.24 CCITT. Изначально создавался для связи компьютера с терминалом. В настоящее время используется в самых различных применениях.

Интерфейс RS-232-C соединяет два устройства. Линия передачи первого устройства соединяется с линией приема второго и наоборот (полный дуплекс). Для управления соединенными устройствами используется программное подтверждение (введение в поток передаваемых данных соответствующих управляющих символов). Возможна организация аппаратного подтверждения путем организации дополнительных RS-232 линий для обеспечения функций определения статуса и управления.

Стандарт	EIA RS-232-C, CCITT V.24
Скорость передачи	115 Кбит/с (максимум)
Расстояние передачи	15 м (максимум)
Характер сигнала	несимметричный по напряжению
Количество драйверов	1
Количество приемников	1
Схема соединения	полный дуплекс, от точки к точке

Формат передаваемых данных показан на рисунке 9. Собственно данные сопровождаются стартовым битом, битом четности и одним или двумя стоповыми битами. Получив стартовый бит, приемник выбирает из линии биты данных через определенные интервалы времени. Очень важно, чтобы тактовые частоты приемника и передатчика были одинаковыми, допустимое расхождение - не более 10%). Скорость передачи по RS-232C может выбираться из ряда: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/с.

Все сигналы RS-232C передаются специально выбранными уровнями, обеспечивающими высокую помехоустойчивость связи (рис.10.). Отметим, что данные передаются в инверсном коде (логической единице соответствует низкий уровень, логическому нулю - высокий уровень).

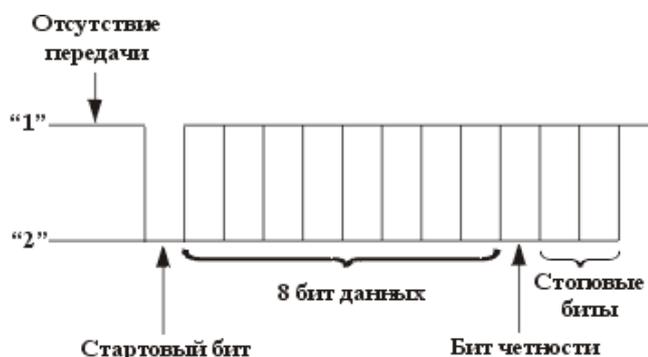


Рис.9 Формат данных RS-232C

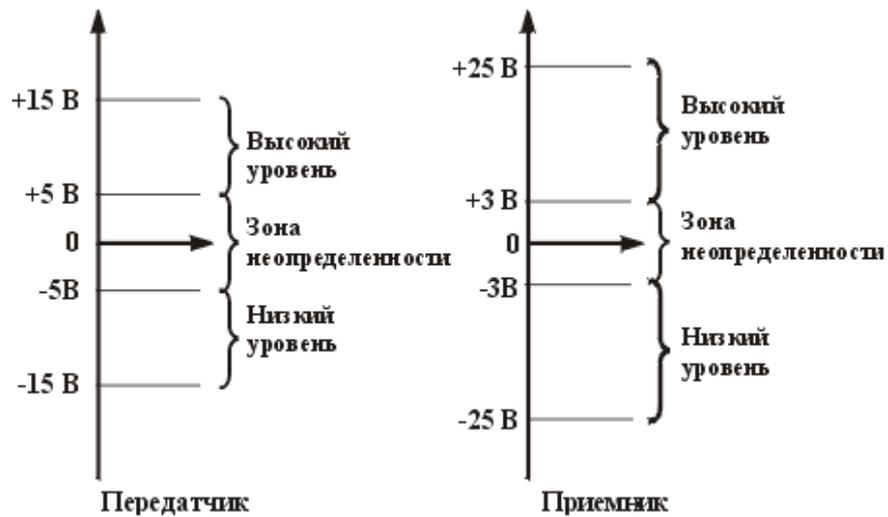


Рис.10 Уровни сигналов RS-232C на передающем и принимающем концах линии связи

Для приведения уровней сигналов TxD и RxD UART МК в соответствие уровням стандарта RS-232 необходимо использовать соответствующие микросхемы преобразователей уровней. Фирма MAXIM, например, выпускает микросхему MAX232. Схема ее включения приведена на рис.11.

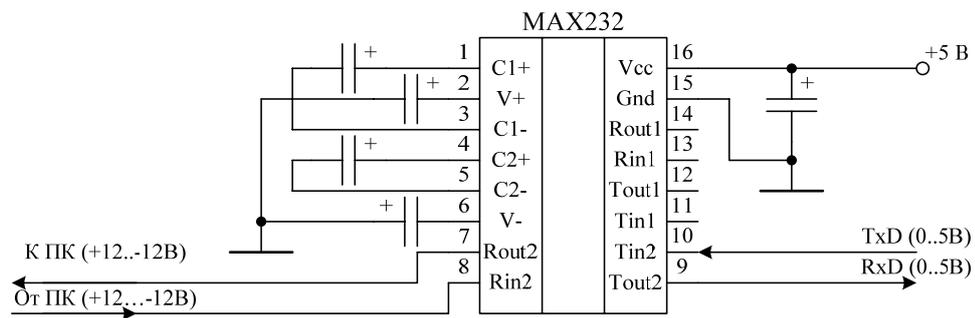


Рис.11 Схема включения микросхемы MAX232

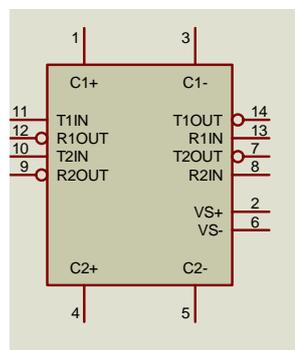
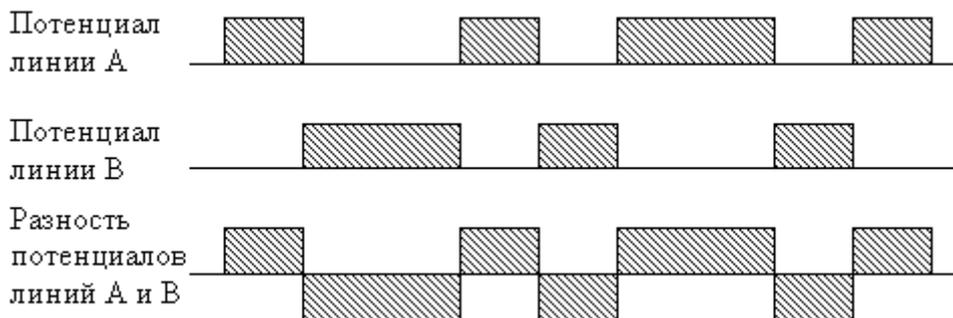


Рис.12 УГО микросхемы MAX232 в программе Proteus

Интерфейс RS-485

Сеть, построенная на интерфейсе RS-485, представляет собой приемопередатчики, соединенные при помощи витой пары – двух скрученных проводов. В основе интерфейса RS-485 лежит принцип дифференциальной (балансной) передачи данных. Суть его заключается в передаче одного сигнала по двум проводам. Причем по одному проводу (условно

А) идет оригинальный сигнал, а по другому (условно В) - его инверсная копия. Другими словами, если на одном проводе "1", то на другом "0" и наоборот. Таким образом, между двумя проводами витой пары всегда есть разность потенциалов: при "1" она положительна, при "0" - отрицательна.

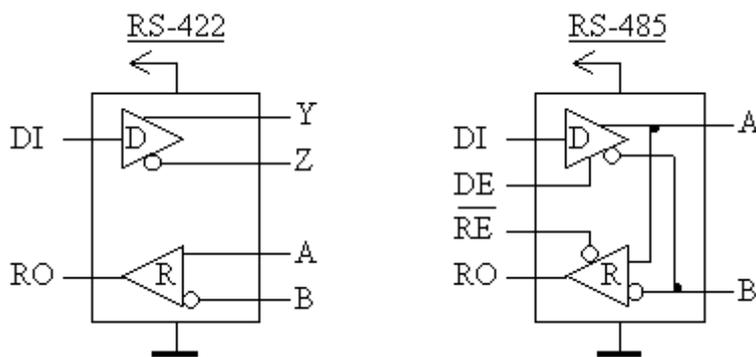


Именно этой разностью потенциалов и передается сигнал. Такой способ передачи обеспечивает высокую устойчивость к синфазной помехе. Синфазной называют помеху, действующую на оба провода линии одинаково. К примеру, электромагнитная волна, проходя через участок линии связи, наводит в обоих проводах потенциал. Если сигнал передается потенциалом в одном проводе относительно общего, как в RS-232, то наводка на этот провод может исказить сигнал относительно хорошо поглощающего наводки общего ("земли"). Кроме того, на сопротивлении длинного общего провода будет падать разность потенциалов земель - дополнительный источник искажений. А при дифференциальной передаче искажения не происходит. В самом деле, если два провода пролегают близко друг к другу, да еще перевиты, то наводка на оба провода одинакова. Потенциал в обоих одинаково нагруженных проводах изменяется одинаково, при этом информативная разность потенциалов остается без изменений.

Аппаратная реализация интерфейса - микросхемы приемопередатчиков с дифференциальными входами/выходами (к линии) и цифровыми портами (к портам UART контроллера). Существуют два варианта такого интерфейса: RS-422 и RS-485.

RS-422 - полнодуплексный интерфейс. Прием и передача идут по двум отдельным парам проводов. На каждой паре проводов может быть только по одному передатчику.

RS-485 - полудуплексный интерфейс. Прием и передача идут по одной паре проводов с разделением по времени. В сети может быть много передатчиков, так как они могут отключаться в режиме приема.



D (driver) - передатчик;

R (receiver) - приемник;

DI (driver input) - цифровой вход передатчика;

RO (receiver output) - цифровой выход приемника;

DE (driver enable) - разрешение работы передатчика;

RE (receiver enable) - разрешение работы приемника;

А - прямой дифференциальный вход/выход;
 В - инверсный дифференциальный вход/выход;
 Y - прямой дифференциальный выход (RS-422);
 Z - инверсный дифференциальный выход (RS-422).

Остановимся поподробнее на приемопередатчике RS-485. Цифровой выход приемника (RO) подключается к порту приемника UART (RX). Цифровой вход передатчика (DI) к порту передатчика UART (TX). Поскольку на дифференциальной стороне приемник и передатчик соединены, то во время приема нужно отключать передатчик, а во время передачи – приемник. Для этого служат управляющие входы - разрешение приемника (RE) и разрешения передатчика (DE). Так как вход RE инверсный, то его можно соединить с DE и переключать приемник и передатчик одним сигналом с любого порта контроллера. При уровне "0" – работа на прием, при "1" – на передачу.

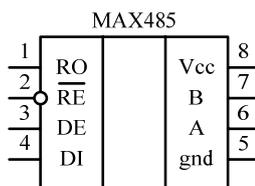


Рис.13 УГО микросхемы MAX487

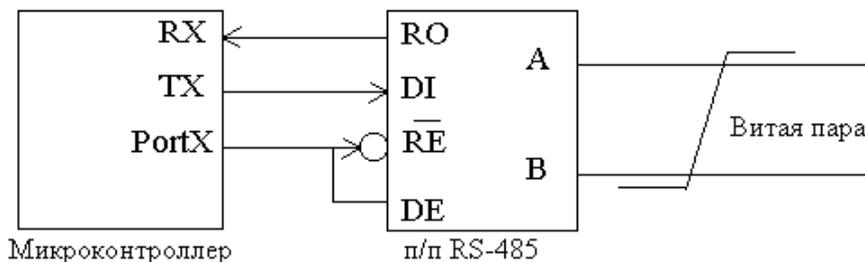


Рис.14 Способ подключения микросхемы MAX487 к МК

Приемник, получая на дифференциальных входах (АВ) разность потенциалов (UAB) переводит их в цифровой сигнал на выходе RO. Чувствительность приемника может быть разной, но гарантированный пороговый диапазон распознавания сигнала производители микросхем приемопередатчиков пишут в документации. Обычно эти пороги составляют ± 200 мВ. То есть, когда $UAB > +200$ мВ - приемник определяет "1", когда $UAB < -200$ мВ - приемник определяет "0". Если разность потенциалов в линии настолько мала, что не выходит за пороговые значения - правильное распознавание сигнала не гарантируется. Кроме того, в линии могут быть и не синфазные помехи, которые исказят столь слабый сигнал.

Все устройства подключаются к одной витой паре одинаково: прямые выходы (А) к одному проводу, инверсные (В) - к другому.

Входное сопротивление приемника со стороны линии (RAB) обычно составляет 12 КОм. Так как мощность передатчика не беспредельна, это создает ограничение на количество приемников, подключенных к линии. Согласно спецификации RS-485 с учетом согласующих резисторов передатчик может вести до 32 приемников. Однако есть ряд микросхем с повышенным входным сопротивлением, что позволяет подключить к линии значительно больше 32 устройств.

Микроконтроллеры семейства MCS-51 фирмы Atmel

Наименование	Память программ, Кб	Внутри-схемное программир.	Память данных EEPROM, Кб	RAM, байт	Таймер/счетчик	Интерфейсы	Аппаратный сторожевой таймер	АЦП	Дополнительно	Встроенный генератор	RC - генератор	Линий ввода/вывода	Напряжение питания	Источников прерываний	Уровней прерываний	Макс. тактовая частота, МГц	Функция x2	Темп. диапазон	Корпус	Развитие версии
In-System Programmable Flash, one cycle																				
AT89LP2052*	2(Flash)	SPI		256	2	UART, SPI	+	-	аналог.компаратор, BOD	+	-	15	2,4-5,5В	6	4	20	-	I	PDIP20, SOIC20, TSSOP20	
AT89LP4052*	4(Flash)	SPI		256	2	UART, SPI	+	-	аналог.компаратор, BOD	+	-	15	2,4-5,5В	6	4	20	-	I	PDIP20, SOIC20, TSSOP20	
In-System Programmable Flash																				
AT89S2051*	2(Flash)	SPI		256	2	UART	-	-	аналог.компаратор	+	-	15	2,7-5,5В	5	4	24	-	C, I	PDIP20, SOIC20	AT89C2051
AT89S4051*	4(Flash)	SPI		256	2	UART	-	-	аналог.компаратор	+	-	15	2,7-5,5В	5	4	24	-	C, I	PDIP20, SOIC20	AT89C4051
AT89LS51	4(Flash)	SPI		128	2	UART	+	-		+	-	32	3В	5	2	16	-	C, I	PDIP40, PLCC44, TQFP44	
AT89S51	4(Flash)	SPI		128	2	UART	+	-		+	-	32	5В	5	2	33	-	C, I	PDIP40, PLCC44, TQFP44	
AT89LS52	8(Flash)	SPI		256	3	UART	+	-		+	-	32	3В	6	2	16	-	C, I	PDIP40, PLCC44, TQFP44	
AT89S52	8(Flash)	SPI		256	3	UART	+	-		+	-	32	5В	6	2	33	-	C, I	PDIP40, PLCC44, TQFP44	
AT89LS8252	8(Flash)	SPI	2	256	3	UART, SPI	+	-		+	-	32	3В	9	3	12	-	C, I	PDIL40, PLCC44, VQFP44	
AT89S8252	8(Flash)	SPI	2	256	3	UART, SPI	+	-		+	-	32	5В	9	3	24	-	C, I	PDIL40, PLCC44, VQFP44	
AT89S53**	12(Flash)	SPI		256	3	UART, SPI	+	-		+	-	32	5В	9	3	24	-	C, I	PDIL40, PLCC44, VQFP44	
AT89LS53**	12(Flash)	SPI		256	3	UART, SPI	+	-		+	-	32	3В / 5В	9	3	12	-	C, I	PDIL40, PLCC44, VQFP44	
AT89S8253	12(Flash)	SPI	2	256	3	UART, SPI	+	-		+	-	32	4-6В	6	4	33	+	C, I	PDIL40, PLCC44, VQFP44	AT89S8252, AT89S53
T89C5115	16(Flash), 2(Boot Flash)	UART, API	2	512	3+PCA	UART	+	+		-	-	16, 20	3В / 5В	14	4	40	+	I	SOIC28, PLCC28, VQFP32	
AT89C51RB2	16(Flash)	UART, API		1280	3+PCA	UART, SPI	+	-	Keyb, BRG	-	-	32	3В / 5В	10	4	60	+	C, I	PDIL40, PLCC44, VQFP44	T89C51RB2
AT89C51RC2	32(Flash)	UART, API		1280	3+PCA	UART, SPI	+	-	Keyb, BRG	-	-	32	3В / 5В	10	4	60	+	C, I	PDIL40, PLCC44, VQFP44	T89C51RC2
AT89C51IC2	32(Flash)	UART, API		1280	3+PCA	UART, SPI, I2C	+	-	Keyb, BRG, RTC	+	-	34	3В / 5В	10	4	60	+	C, I	PLCC44, VQFP44	T89C51IC2
AT89C51AC2	32(Flash), 2(ROM)	UART, API	2	1280	3+PCA	UART	+	+		-	-	34	5В	14	4	40	+	I	PLCC44, VQFP44	T89C51AC2
AT89C51AC3*	64(Flash), 2(ERAM)	UART, API	2	256	3+PCA	UART, SPI	+	+		-	-	36	3В / 5В	14	4	60	+	I	PLCC44, VQFP44, PLCC52, VQFP64	T89C51AC2
AT89C51RD2	64(Flash)	UART, API		1792	3+PCA	UART, SPI	+	-	Keyb, BRG, RTC, PFD	-	-	32,48	2,7-5,5В	9	4	60	+	I	DIL40, PDIL40, PLCC44, VQFP44, PLCC68, VQFP64	T89C51RD2
AT89C51ED2	64(Flash)	UART, API	2	1792	3+PCA	UART, SPI	+	-	Keyb, BRG, RTC, PFD	-	-	32,48	2,7-5,5В	9	4	60	+	I	DIL40, PDIL40, PLCC44, VQFP44, PLCC68, VQFP64	T89C51RD2
AT89C51ID2	64(Flash)	UART, API	2	1792	3+PCA	UART, SPI, I2C	+	-	Keyb, BRG, RTC, PFD	-	-	32,48	2,7-5,5В	9	4	60	+	I	PLCC44, VQFP44	T89C51RD2
Flash																				
AT89C2051	2(Flash)			128	2	UART	-	-	аналог.компаратор	+	-	14	3В / 5В	6	2	24	-	C, I	PDIP20, SOIC20	
AT89C2051x2	2(Flash)			128	2	UART	-	-	аналог.компаратор	+	-	14	3В / 5В	6	2	24	+	C, I	PDIP20, SOIC20	AT89C2051
AT89C4051	4(Flash)			128	2	UART	-	-	аналог.компаратор	+	-	14	3В / 5В	6	2	24	-	C, I	PDIP20, SOIC20	
AT89C55WD	20(Flash)			256	3	UART	+	-		+	-	32	5В	8	3	33	-	C, I	PDIP40, PLCC44, TQFP44	
AT89LV55	20(Flash)			256	3	UART	-	-		+	-	32	3В	8	3	12	-	C, I	PDIP40, PLCC44, TQFP44	
AT89C51RC	32(Flash)			512	3	UART	+	-		+	-	32	5В	8	3	33	-	C, I	PDIP40, PLCC44, TQFP44	
One Time Programmable (OTP)																				
TS87C52X2	8(OTP)			256	3	UART	-	-		-	-	32	3В / 5В	6	4	40, 30	+	C, I	PDIP40, PLCC44, VQFP44	
AT87C5103	12(OTP)			256	2	SPI	+	-	PWI	+	-	16	2,7-5,5В	5	4	16	+	I, A	SSOP16, SSOP24	
TS87C54X2	16(OTP)			256	3	UART	+	-		-	-	32	3В / 5В	6	4	60, 40, 30	+	C, I	PDIP40, PLCC44, VQFP44	
TS87C51RB2	16(OTP)			512	3+PCA	UART	+	-		-	-	32	3В / 5В	7	4	60, 40, 30	+	C, I	PDIP40, PLCC44, VQFP44	
TS87C58X2	32(OTP)			256	3	UART	+	-		-	-	32	3В / 5В	6	4	60, 40, 30	+	C, I	PDIP40, PLCC44, VQFP44	
TS87C51RC2	32(OTP)			512	3+PCA	UART	+	-		+	-	32	3В / 5В	7	4	60, 40, 30	+	C, I	PDIP40, PLCC44, VQFP44	
TS87C51RD2	64(OTP)			1024	3+PCA	UART	+	-		+	-	48	3В / 5В	7	4	60, 40, 30	+	C, I	PDIL40, PLCC44, VQFP44	
ROM																				

TS80C52X2	8(ROM)			256	3	UART	-	-		+	-	32	3B / 5B	6	4	40	+	C, I	PDIL40, PLCC44, VQFP44	
TS83C51RB2	16(Mask ROM)			512	3+PCA	UART	+	-		+	-	32	3B / 5B	7	4	40, 30	+	C, I	PDIP40, PLCC44, VQFP44	
AT83C51RB2	16(Mask ROM)			1280	3+PCA	UART, SPI	+	-	BRG, Keyb	+	-	32	3B / 5B	7	4	60	+	C, I	PDIP40, PLCC44, VQFP44	TS83C51RB2
TS80C54X2	16(ROM)			256	3	UART	+	-		-	-	32	3B / 5B	6	4	40, 30	+	C, I	PDIL40, PLCC44, VQFP44 1.4, PQFP F1(13.9 footprint)	
TS80C58X2	32(ROM)			256	3	UART	+	-		-	-	32	3B / 5B	6	4	40, 30	+	C, I	PDIL40, PLCC44, VQFP44 1.4, PQFP F1(13.9 footprint)	
TS83C51RC2	32(ROM)			512	3+PCA	UART	+	-		+	-	32	3B / 5B	7	4	40, 30	+	C, I	PDIP40, PLCC44, VQFP44	
AT83C51RC2	32(ROM)			1280	3+PCA	UART, SPI	+	-	BRG, Keyb	+	-	32	3B / 5B	7	4	60	+	C, I	PDIP40, PLCC44, VQFP44	TS83C51RC2
AT83C51IC2	32(ROM)			1280	3+PCA	UART, SPI, I2C	+	-	BRG, Keyb, RTC	+	-	34	3B / 5B	7	4	60	+	C, I	PLCC44, VQFP44	TS83C51RC2
TS83C51RD2	64(ROM)			1024	3+PCA	UART	+	-		+	-	32, 48	3B / 5B	7	4	40, 30	+	C, I	PDIP40, PLCC44, VQFP44, PLCC68, VQFP64	
ROMless																				
TS80C31X2	-			128	2	UART	-	-		-	-	32	3B / 5B	5	4	40, 30	+	C, I	PDIL40, PLCC44, VQFP44 1.4, PQFP F1	
TS80C32X2	-			256	3	UART	-	-		-	-	32	3B / 5B	6	4	40, 30	+	C, I	PDIL40, PLCC44, VQFP44 1.4, PQFP F1(13.9 footprint)	
TS80C51RA2	-			512	3+PCA	UART	+	-		+	-	32	3B / 5B	7	4	40, 30	+	C, I	PDIL40, PLCC44, VQFP44 1.4, PLCC68, VQFP64 1.4	
AT80C51RD2	-			1280	3+PCA	UART, SPI	+	-	BRG, Keyb	+	-	32	3B / 5B	7	4	60	+	C, I	PDIL40, PLCC44, VQFP44 1.4, PLCC68, VQFP64 1.4	TS80C51RA2
AT80C51ID2	-			1280	3+PCA	UART, SPI, I2C	+	-	BRG, Keyb, RTC	+	-	34	3B / 5B	7	4	60	+	C, I	PDIL40, PLCC44, VQFP44 1.4, PLCC68, VQFP64 1.4	TS80C51RA2
C251																				
TSC87251G2D	32(OTP)			1024	3+PCA	UART, SPI, I2C, uWare	+	-	Keyb, NMI, BRG	-	-	32	3B / 5B	+	?	16, 24	-	C, I, M	PDIL40, PLCC44, VQFP44	
TSC80251G1D	16(ROM)			1024	3+PCA	UART, SPI, I2C, uWare	+	-	Keyb, NMI, BRG	-	-	32	3B / 5B	+	?	12, 16, 24	-	C, I, M	PDIL40, PLCC44, VQFP44	
TSC83251G2D	32(Mask ROM)			1024	3+PCA	UART, SPI, I2C, uWare	+	-	Keyb, NMI, BRG	-	-	32	3B / 5B	+	?	16, 24	-	C, I, M	PDIL40, PLCC44, VQFP44	
TSC80251G2D	-			1024	3+PCA	UART, SPI, I2C, uWare	+	-	Keyb, NMI, BRG	-	-	32	3B / 5B	+	?	16 & 24	-	C, I, M	PDIL40, PLCC44, VQFP44	
USB																				
AT89C5131	32(Flash), 3(Boot Flash)	UART, USB, API	1	1280	3+PCA	USB, UART, SPI, I2C	-	-	Keyb, 4 вывода светодиодов	+	-	34	3.3B/5B	11	4	40	+	C, I	SOIC28, PLCC52, MLF48, VQFP64	
AT89C5132	64(Flash), 4(Boot Flash)	UART, USB, API		2304	2	USB, UART, SPI, I2C, MMC, IDE, I2S	+	2	аудио ЦАП, Keyb	+	-	44	3B	13	4	40	+	I	TQFP80, TQFP64	
CAN Multiplexing																				
T89C51CC01	32(Flash), 2(Flash Boot)	UART, CAN	2	1280	3+PCA	CAN, UART	+	+	BRG	-	-	32+2digital	3-5,5B	14	4	40	+	I	TQFP44, PLCC44, CA-BGA64	
T89C51CC02	16(Flash), 2(Boot Flash)	UART, CAN, API	2	512	3+PCA	CAN, UART	-	+	BRG	-	-	16, 20	3-5,5B	14	4	40	+	I	PLCC28, SOIC28, TSSOP28, SOIC24	
AT89C51CC03	64(Flash), 2(Flash Boot)	UART, CAN, API	2	2048	3+PCA	CAN, UART, SPI	+	+	BRG, PFD	-	-	32+4digital	3-5,5B	14	4	40	+	I	VQFP44, PLCC44, VQFP64, PLCC52, CA-BGA64	
MP3 Readers																				
AT89C51SND1	64(Flash), 4(Boot Flash)	UART, USB, API		2304	2	USB, UART, SPI, I2C, MMC, IDE, I2S	+	+	аудио ЦАП, MP3 декодер, Keyb, BRG	+	-	44	3	13	4	40	+	I	TQFP80	
AT83C51SND1	64(ROM)			2304	2	USB, UART, SPI, I2C, MMC, IDE, I2S	+	+	аудио ЦАП, MP3 декодер, Keyb, BRG	+	-	44	3	13	4	40	+	I	TQFP80	
AT89C51SND2	64(Flash), 4(Boot Flash)	UART, USB, API		2304	2	USB, UART, SPI, I2C, MMC, IDE, I2S	+	+	аудио ЦАП, MP3 декодер, Keyb, BRG, низкое энергопотребление (до 35мА)	+	-	32	3	13	4	40	+	I	BGA100	

AT83C51SND2	64(ROM)	UART, USB, API		2304	2	USB, UART, SPI, I2C, MMC, IDE, I2S	+	+	аудио ЦАП, MP3 декодер, Кеув, BRG, низкое энергопо- требление (до 35мА)	+	-	32	3	13	4	40	+	I	BGA100	
-------------	---------	-------------------	--	------	---	--	---	---	---	---	---	----	---	----	---	----	---	---	--------	--

Донецкий национальный технический университет

КАФЕДРА ЭЛЕКТРОННОЙ ТЕХНИКИ

ЗАДАНИЕ

Для выполнения курсового проекта (работы) по курсу:

_____ (название курса)

_____ (фамилия, имя и отчество студента)

Группа _____ Срок проектирования с _____ по _____

Дата выдачи задания ” _____ ” _____ 20__ г. Руководитель _____

Тема проекта (работы) _____

Входные данные для проекта (работы)

1. Наименование проектируемого устройства: _____

2. Характеристика входных сигналов

2.1. Кол-во аналоговых входных сигналов: _____

2.2. Диапазон изменения аналоговых входных сигналов: _____

2.3. Разрядность АЦП _____

2.4. Период дискретизации сигнала по каждому каналу _____

2.5. Кол-во дискретных входных сигналов _____

2.6. Кол-во цифровых входных сигналов _____

3. Тип микроконтроллера: _____

4. Внешний АЦП _____

5. Индикатор _____

6. Клавиатура _____

7. Прочая периферия (память, часы/календарь, таймер и т.п.) _____

8. Тип интерфейса обмена данными _____

9. Характеристика выходных управляющих сигналов

9.1. Кол-во аналоговых выходных сигналов _____

9.2. Диапазон аналоговых выходных сигналов _____

9.3. Кол-во дискретных выходных сигналов _____

Календарный план выполнения проекта (работы)

Вид работы	Раздел	Срок выполнения	Отметка о выполнении
Разработка структурной схемы МПС	1		
Выбор элементной базы	2		
Разработка функциональной схемы МПС и словесного алгоритма ее работы	3		
Разработка принципиальной схемы	4		
Разработка программного обеспечения	5		
Имитационное моделирование МПС в Proteus	6		
Оформление графической части проекта (работы)			
Оформление пояснительной записки			

Содержание пояснительной записки к курсовому проекту (работе), должно соответствовать вышеприведенному перечню и требованиям предъявляемым к оформлению курсовых проектов, состав графической части, вид и необходимость выполнения и визуализации отдельных элементов проектирования, уточняется на консультациях с руководителем.

Срок сдачи проекта руководителю для проверки « _____ » _____ 20__ г.

Подпись руководителя _____

Задание принято для выполнения « _____ » _____ 20__ г.

Подпись студента _____

Список рекомендованной литературы

1. Бродин В.Б., Шагурин И.И. Микроконтроллеры. Архитектура, программирование, интерфейс.–М.: Издательство ЭКОМ, 1999.–400с.
2. Фрунзе А.В. Микроконтроллеры? Это же просто! Т1,Т2,Т3.–М.: ООО «ИД СКИМЕН».
3. Микушин А.В. Занимательно о микроконтроллерах. – СПб.: БХВ-Петербург, 2006. – 432 с.: ил.
4. Сташин В.В. и др. Проектирование цифровых устройств на однокристальных микроконтроллерах/ В.В.Сташин, А.В.Урусов, О.Ф.Мологонцева.–М.: Энергоатомиздат, 1990. – 224с.
5. Современные микроконтроллеры: Архитектура, средства проектирования, примеры применения, ресурсы сети Интернет. Под ред. Коршуна И.В.; Составление, пер. с англ. И литературная обработка Горбунова Б.Б.–М.: Издательство «Аким», 1998.–272 с., ил.
6. Новиков Ю.В. и др. Разработка устройств сопряжения для персонального компьютера типа IBM PC. Под общей редакцией Ю.В.Новикова. Практ.пособие – М.: ЭКОМ., 1998 – 224 с.
7. Николайчук О.И. Системы малой автоматизации/ О.И.Николайчук – М.: СОЛОН-Пресс, 2003. 256с.
8. Бирюков С.А. Применение интегральных микросхем серий ТТЛ. – М.: «Патриот», МП «Символ-Р», «Радио», 1992. – 120 с.
9. <http://www.gaw.ru>