

Санкт-Петербургский государственный политехнический университет

Факультет технической кибернетики

Кафедра компьютерных систем и программных технологий

## **Отчёт по лабораторной работе**

«Работа с портами МК SAB 80C515

по дисциплине "Микропроцессорные системы"

Работу выполнили студенты

Дорофеев Юрий Владимирович гр. 4081/12 \_\_\_\_\_

Тимофеев Иван Дмитриевич гр. 4081/13 \_\_\_\_\_

Работу принял преподаватель

Павловский Евгений Григорьевич \_\_\_\_\_

г. Санкт-Петербург

2012

## 1. Цель работы

Целью данной лабораторной работы является:

- знакомство с организацией взаимодействия МК с внешними устройствами (цифровыми и аналоговыми), подключаемыми с помощью портов МК;
- овладение приемами программирования периферийных устройств (ЖКИ, клавиатуры, АЦП), входящих в состав лабораторного стенда.

## 2. Программа работы

Работа рассчитана на 2-3 посещения и предполагает:

Разработку структуры информационных связей МК с подключаемыми внешними устройствами (клавиатурой, модулем ЖКИ, датчиками аналоговых сигналов) и реализовать её, соединив разъёмы портов МК с разъёмами подключаемых устройств.

В ходе работы требуется:

- разработать и выполнить программу, реализующую вывод на экран ЖКИ двухстрочного текста, в котором используются различные алфавиты (латиница и кириллица):

“Microcontrollers 2012  
Дорофеев Тимофеев”

При разработке использовать модуль `indic.asm`.

- разработать программу `klav.asm`, реализующую опрос клавиатуры и запись кодов состояния клавиш в ячейки памяти с адресами `30h ... 33h`
- модифицировать программу «опрос клавиатуры», дополнив её модулем, формирующим номер нажатой клавиши. Присвоить ей имя `klav1.asm`.
- разработать программу, отображающую на экране модуля ЖКИ номер нажатой клавиши, используя программы `indic.asm` и `klav1.asm`, подключив их к разрабатываемой программе директивой `include<имя файла>`.
- разработать и выполнить программу вывода на ЖКИ цифровых кодов (эквивалентов) аналоговых сигналов от двух источников. В качестве источников аналоговых сигналов в стенде используются источник регулируемого напряжения, представленный сигналом с выхода потенциометра и сигнал с выхода интегратора. При разработке программы использовать программы `indic.asm` и `adc_read.asm`, подключив их к разрабатываемой программе директивой `include <имя файла>`.
- разработать программу, которая, наряду с отображением цифровых кодов аналоговых сигналов от двух датчиков, позволяет продемонстрировать скорость обработки изменяющегося во времени аналогового сигнала с выхода выбранного пользователем датчика в цифровой код на экране инструментальной ЭВМ.

### 3. Описание периферийных устройств микро-ЭВМ на базе МК SAB80C515

#### 3.1. Порты ввода-вывода параллельной информации

МК SAB 80C515 содержит в своем составе шесть двунаправленных портов ввода-вывода P0 – P5, предназначенных для обмена информацией с внешними устройствами (ВУ). При соответствующем управлении двунаправленный порт обеспечивает как ввод (передачу информации от ВУ в МК), так и вывод (передачу информации от МК в ВУ). 8-разрядные порты P0, P1, P2, P3 являются мультиплексируемыми. Каждый вывод этих портов может использоваться либо для выполнения стандартного ввода-вывода либо для выдачи/приема специальных функциональных сигналов, связанных с внутренними периферийными устройствами или управляющих внешними устройствами. Если отдельный специальный сигнал не используется в конкретном приложении, то соответствующий вывод может быть использован для стандартного ввода-вывода.

Порты P1-P5 являются квазидвунаправленными (функция ввода/вывода задается программно и не изменяется во времени), порт P0 – двунаправленный (динамическое переключение ввода-вывода). Двунаправленные и квазидвунаправленные порты отличаются схемотехникой формирователя выходных сигналов (рис. 1 и 2).

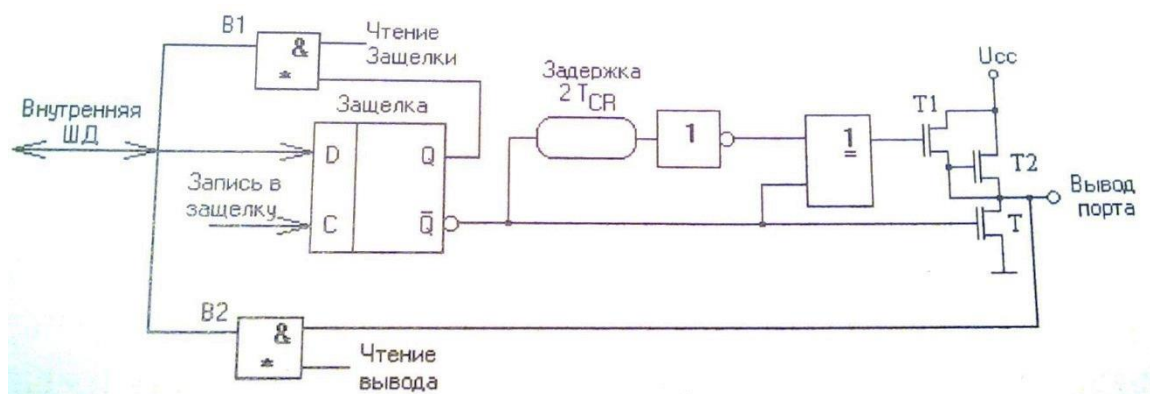


Рис. 1. Структура разряда квазидвунаправленного порта

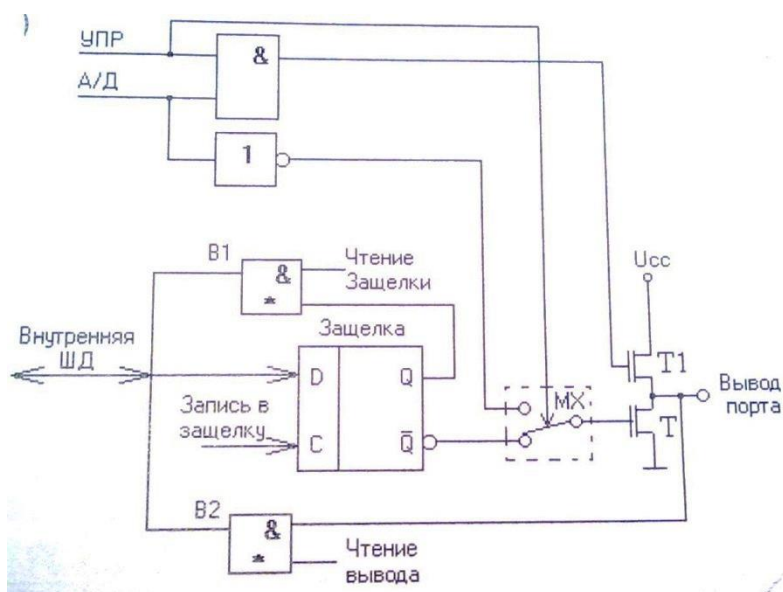


Рис. 2. Структура разряда двунаправленного порта P0

Назначение портов P0 – P3 следующее:

Порт P0 и порт P2. При работе с внешней памятью (программ и/или данных) линии портов P0 и P2 выполняют функции внешней шины адреса и двунаправленной шины данных. С помощью портов P0 осуществляется мультиплексирование младшего байта адреса и данных на внешней шине МК. При всех внешних обменах на выводы порта P0 последовательно во времени поступают младший байт адреса A7 – A0, а затем данные D7 – D0. Для мультиплексирования сигналов адрес/данные на выходе порта P0 устанавливается внешний буферный регистр, фиксирующий младший байт адреса с помощью выходного сигнала ALE. Кроме того, с помощью порта P0 задаются данные при программировании внутреннего ППЗУ и читается содержимое внутренней памяти программ.

Через порт P2 выводится старший байт адреса A15 – A8.

Порт P1 и порт P3. Все линии портов P1 и P3 могут использоваться для ввода-вывода альтернативных сигналов, при этом отдельные линии порта P1 являются многофункциональными. С помощью порта P1, кроме того, задается младший байт адреса при программировании внутреннего ППЗУ и при чтении внутренней памяти программ.

### 3.2. Модуль ЖКИ

Для управления работой ЖКИ-модуля применен встроенный управляющий контроллер HD44780 фирмы Hitachi. Контроллер управляет 2-мя строками по 20 символов в каждой, при матрице символа 5x7 точек.

Для соединения ЖКИ-модуля с управляющей системой используется параллельная синхронная шина, насчитывающая 8 линий данных DB0...DB7, линию выбора операции R/W, линию выбора регистра RS и линию стробирования/синхронизации E. Кроме линий управляющей шины имеются две линии для подачи напряжения питания 5 В – GND и VCC, и линия для подачи напряжения питания драйвера ЖКИ – VO.

Схема подключения питания и включения модуля в систему с микро-ЭВМ показана на рис. 3 и рис. 4. Подстроечный резистор R1 позволяет плавно менять напряжение питания драйвера ЖКИ, что приводит к изменению угла поворота жидких кристаллов. Этим резистором можно отрегулировать фактическую контрастность при некотором преимущественном угле наблюдения (снизу-вверх или сверху-вниз).

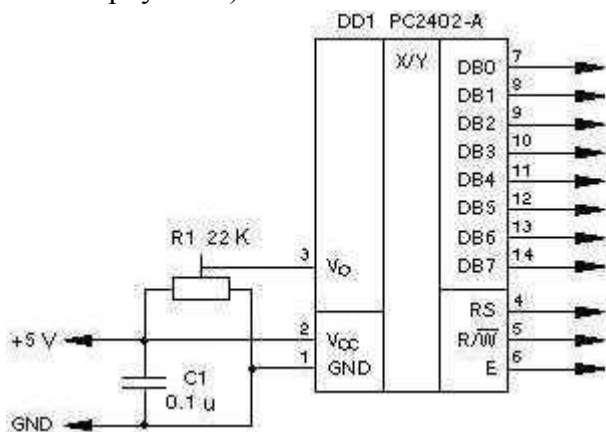


Рис. 3. Подача питания на ЖКИ-модуль

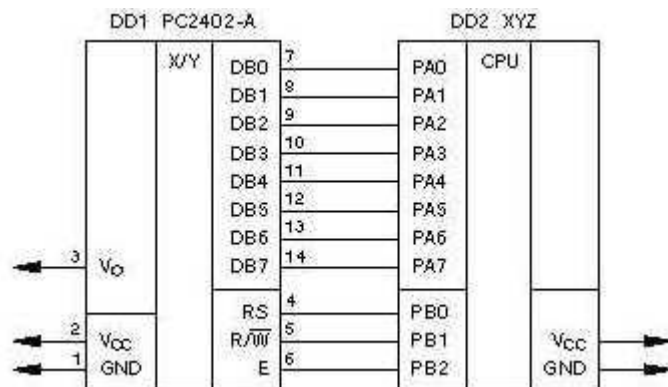


Рис. 4. Подключение к управляющей системе

В исходном состоянии сигнал Е (линия разрешения программирования) = 0, сигнал R/W («чтение»\«запись») = 0, значение сигнала RS (выбор типа посылки «команда» - «данные») – произвольное, шина данных DB0...DB7 в состоянии высокого импеданса (HI). Такое состояние управляющих сигналов (Е и R/W) должно поддерживаться всё время в промежутках между операциями обмена с ЖКИ-модулем. Шина данных в эти моменты свободна и может использоваться в мультиплексном режиме для каких-либо других целей, например, для сканирования матрицы клавиатуры.

Упрощенная структурная схема контроллера приведена на рис. 5. Можно выделить основные элементы, с которыми приходится взаимодействовать при программном управлении: регистр данных (DR), регистр команд (IR), видеопамять (DDRAM), ОЗУ знакогенератора (CGRAM), счетчик адреса памяти (AC), флаг занятости контроллера.

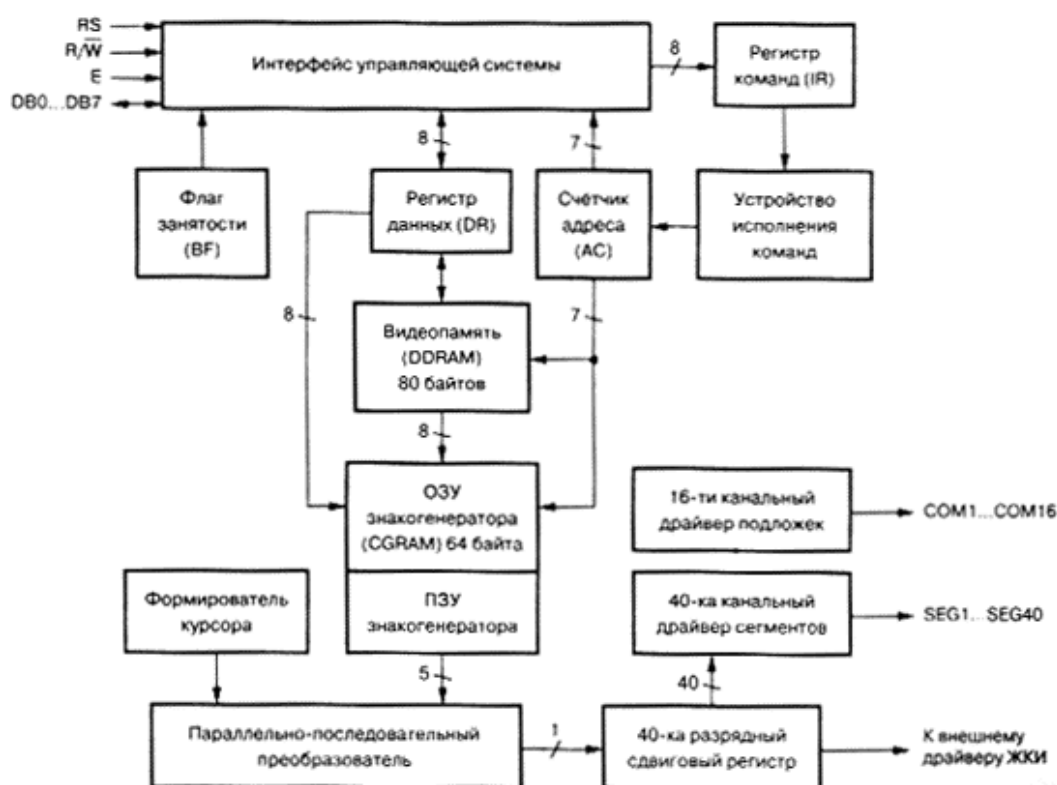


Рис. 5. Упрощённая структурная схема контроллера управления модулем ЖКИ

Управление контроллером ведется посредством интерфейса управляющей системы. Основными объектами взаимодействия являются регистры DR и IR. Выбор адресуемого регистра производится линией RS, если RS = 0 - адресуется регистр команд (IR), если RS = 1 - регистр данных (DR).

Данные через регистр DR, в зависимости от текущего режима, могут помещаться (или прочитываться) в видеопамять (DDRAM) или в ОЗУ знакогенератора (CGRAM) по текущему адресу, указываемому счетчиком адреса (AC). Информация, попадающая в регистр IR, интерпретируется устройством выполнения команд как управляющая последовательность. Прочтение регистра IR возвращает в 7-ми младших разрядах текущее значение счетчика AC, а в старшем разряде флаг занятости (BF).

У контроллера модуля ЖКИ существует набор внутренних флагов, определяющих режимы работы различных элементов контроллера (таблица 1).

*Таблица 1. Флаги, управляющие работой контроллера ЖКИ-модуля*

I/D:	режим смещения счетчика адреса AC, 0 - уменьшение, 1 - увеличение.
S:	флаг режима сдвига содержимого экрана. 0 - сдвиг экрана не производится, 1 - после записи в DDRAM очередного кода экран сдвигается в направлении, определяемым флагом I/D: 0 - вправо, 1 - влево. При сдвиге не производится изменение содержимого DDRAM. изменяются только внутренние указатели расположения видимого начала строки в DDRAM.
S/C:	флаг-команда, производящая вместе с флагом R/L операцию сдвига содержимого экрана (так же, как и в предыдущем случае, без изменений в DDRAM) или курсора. Определяет объект смещения: 0 - сдвигается курсор, 1 - сдвигается экран.
R/L:	флаг-команда, производящая вместе с флагом S/C операцию сдвига экрана или курсора. Уточняет направление сдвига: 0 - влево, 1 - вправо.
D/L:	флаг, определяющий ширину шины данных: 0 - 4 разряда, 1 - 8 разрядов.
N:	режим развертки изображения на ЖКИ: 0 - одна строка, 1 - две строки
F:	размер матрицы символов: 0 - 5 x 8 точек, 1 - 5 x 10 точек.
D:	наличие изображения: 0 - выключено, 1 – включено
C:	курсор в виде подчеркика: 0 - выключен, 1 – включен
B:	курсор в виде мерцающего знакоместа: 0 - выключен, 1 - включен.

Список управляющих комбинаций битов регистра IR и выполняемые ими команды приведены в таблице 2. Так как на момент включения ЖКИ-модуль ничего не отображает (флаг D = 0), то для того, чтобы вывести какой-либо текст необходимо, как минимум, включить отображение, установив флаг D = 1. Последовательность для инициализации ЖКИ-модуля:

- 38h устанавливает режим отображения 2-х строк с матрицей 5 x 8 точек и работу с 8-ми разрядной шиной данных;
- OCh включает отображение на экране ЖКИ-модуля, без отображения курсоров;
- 06h устанавливает режим автоматического перемещения курсора слева-направо после вывода каждого символа.

Таблица 2. Управляющие комбинации битов регистра IR

D7	D6	D5	D4	D3	D2	D1	D0	Назначение
0	0	0	0	0	0	0	1	Очистка экрана, AC = 0, адресация AC на DDRAM
0	0	0	0	0	0	1	-	AC = 0, адресация на DDRAM, сброшены сдвиги, начало строки адресуется в начале DDRAM
0	0	0	0	0	1	I/D	S	Выбирается направление сдвига курсора или экрана
0	0	0	0	1	D	C	B	Выбирается режим отображения
0	0	0	1	S/C	R/L	-	-	Команда сдвига курсора/экрана
0	0	1	DL	N	F	-	-	Определение параметров развертки и ширины шины данных
0	1	AG	AG	AG	AG	AG	AG	Присвоение счетчику AC адреса в области CGRAM
1	AD	AD	AD	AD	AD	AD	AD	Присвоение счетчику AC адреса в области DDRAM

Контроллер ЖКИ-модуля содержит полный набор русских символов в прописном и строчном вариантах, поэтому он весьма удобен для отечественных применений. В таблице 3 приведен набор символов этого контролера.

Таблица 3. Таблица кодов символов контролера ЖКИ-модуля, набор "Russian"

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	P			Б	Ю	Ч	.	Д	И
1			!	1	A	Q	a	q			Г	Я	Ш	!	Ц	И
2			"	2	B	R	b	r			Ё	Б	Ъ	"	Щ	И
3			#	3	C	S	c	s			Ж	В	Ы	"	Д	И
4			\$	4	D	T	d	t			Э	Г	Ь	Ъ	Ф	И
5			%	5	E	U	e	u			И	Ё	Э	Х	Ц	И
6			&	6	F	V	f	v			Й	Ж	Ю	Ъ	Щ	И
7			'	7	G	W	g	w			Л	Э	Я	!	'	И
8			(	8	H	X	h	x			П	И	Э	И	"	И
9			)	9	I	Y	i	y			У	Й	Э	↑	~	И
A			*	:	J	Z	j	z			Ф	К	"	↓	ё	И
B			+	;	K	[	k	[]			Ч	Л	"	И	Э	И
C			,	<	L	φ	l	[]			Ш	М	И	И	И	И
D			-	=	M	]	m	[]			Ь	И	И	И	И	И
E			.	>	N	^	n	ε			Ы	П	И	И	И	И
F			/	?	O	_	o	ε			Э	Т	И	И	И	И

### 3.3. Клавиатура

Нажатие клавиш клавиатуры всегда осуществляется в произвольные моменты времени, поэтому обслуживание клавиатуры является асинхронным. На практике в основном используются следующие режимы асинхронного обслуживания клавиатуры:

- с ожиданием асинхронного ввода;
- периодическое сканирование состояния клавиатуры;
- по прерыванию.

В первом случае предполагается, что сканирование происходит постоянно и является центральной функцией программы, обслуживающей микроконтроллерную систему. Все операции зависят от команд, формируемых оператором с клавиатуры или переключателей.

В режиме периодического сканирования через определённые промежутки времени осуществляется опрос клавиатуры. При обнаружении нажатой клавиши определяется её номер и выполняется соответствующая команда обслуживания. Период сканирования должен быть достаточно коротким ( $T_c \leq 0.1c$ ), чтобы зафиксировать нажатие клавиши с учётомдребезга и удержания в интервале. Несмотря на то, что нажатие является случайным событием, при большом периоде сканирования ( $T_c > 0.1c$ ) для получения необходимой реакции может потребоваться неоднократное нажатие клавиши.

В режиме обслуживания клавиатуры по прерываниям при нажатии клавиш контроллер обеспечивает обслуживание клавиатуры, как в предыдущем случае. Однако в этом режиме сканирование не связано с каким-либо фиксированным периодом, поэтому в промежутках между моментами поступления сигналов прерывания контроллер может выполнять любые действия, предусмотренные программой. При выполнении лабораторных работ данного цикла предполагается использование первого режима обслуживания клавиатуры с ожиданием асинхронного ввода.

Используемый в лабораторном стенде вариант схемы объединения 16-ти клавиш (кнопок) пульта в клавиатуру показан на рис. 6.

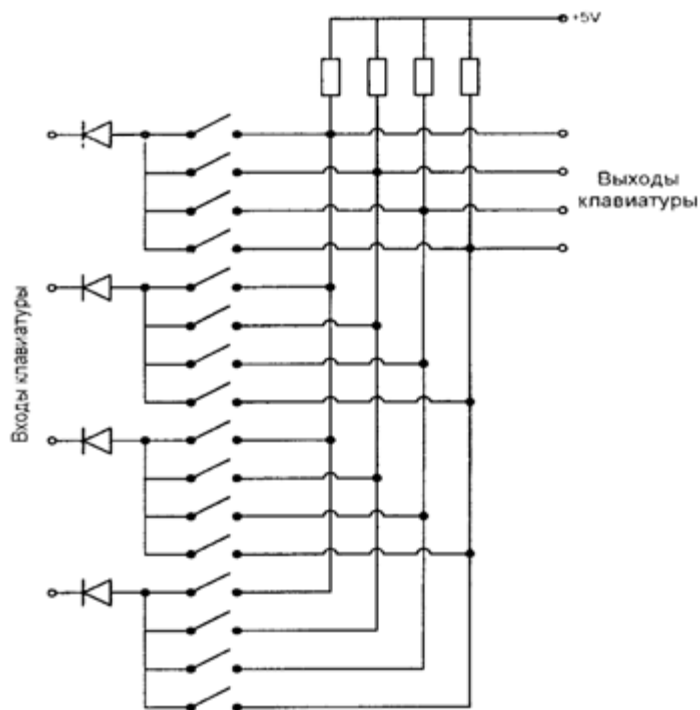


Рис. 6. Схема реализации клавиатуры лабораторного стенда



Клавиши объединены в группы по четыре в каждой группе. Общий вывод каждой группы клавиш через диод соединён с одной из четырёх входных шин клавиатуры. Отдельные выводы клавиш группы соединены с соответствующими выводами клавиш других групп и подключены к выходной шине клавиатуры. Для подключения клавиатуры к МК необходимо использовать два порта МК: порт ввода, подсоединяемый к выходной шине клавиатуры, и порт вывода, соединяемый с входной шиной клавиатуры.

Вариант подключения клавиатуры к микроконтроллеру, используемый при выполнении лабораторной работы представлен на рис. 7.

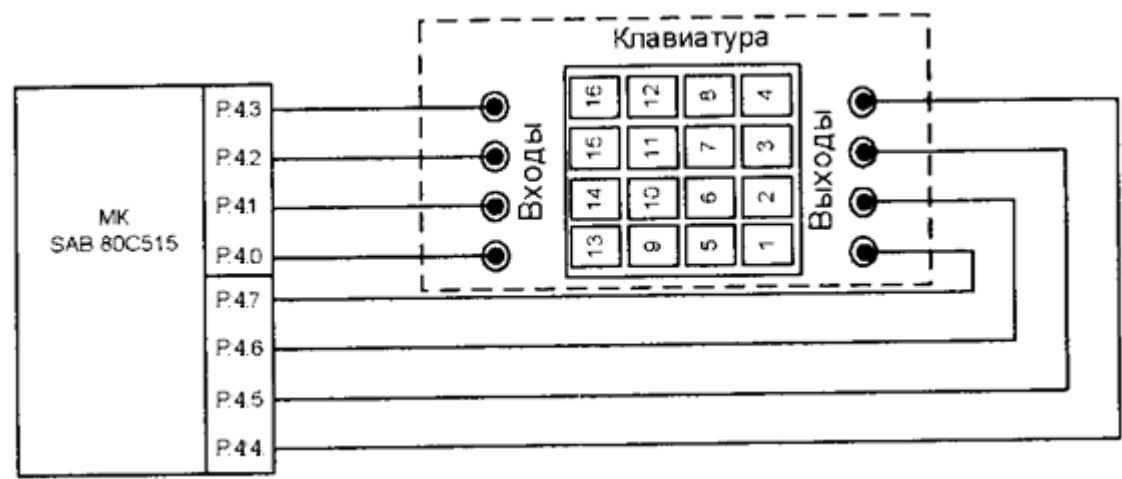


Рис. 7. Вариант схемы подключения клавиатуры к МК.

Через порт P1 на вход клавиатуры подается входное воздействие, а с помощью порта P5 фиксируется ее состояние. Опрос клавиатуры удобно выполнить, организовав “бегущий ноль” по строкам клавиатуры. Результат опроса клавиатуры записывается во внутреннюю память МК по адресам 30 - 33h. Указанную область памяти можно назвать картой памяти клавиатуры. Входные воздействия и выходная реакция при нажатии только одной клавиши показаны в таблице 4.

Таблица 4. Входные и выходные воздействия при нажатии одной клавиши

Входной код	Опрашиваемая клавиша	Выходной код
0111 xxxx	0, 1, 2 или 3	xxxx 0111 – нажата клавиша левого столбца
1011 xxxx	4, 5, 6 или 7	xxxx 1011 – нажата клавиша среднего левого столбца
1101 xxxx	8, 9, 10 или 11	xxxx 1101 – нажата клавиша среднего правого столбца
1110 xxxx	12, 13, 14 или 15	xxxx 1110 – нажата клавиша правого столбца
		xxxx 1111 – не нажата ни одна клавиша
		Остальные коды => нажато больше одной клавиши

### 3.4. Аналого-цифровой преобразователь

Для ввода аналоговых сигналов и их последующего преобразования в цифровой код МК SAB 80C515 содержит в своём составе многоканальный аналого-цифровой преобразователь (АЦП) с числом каналов 8. АЦП обеспечивает преобразование аналоговых сигналов напряжения в диапазоне от 0 В до 5.12 В в цифровой 8- или 10-разрядный код.

Функциональная схема АЦП микроконтроллера SAB80515 показана на рис. 8.

Она содержит три регистра специальных функций:

- ADCON - регистр управления АЦП;
- ADDAT - регистр данных АЦП;
- DAPR - регистр программирования ЦАП;

Последний вместе со встроенным ЦАП служит для программного изменения опорных напряжений АЦП, что позволяет программировать диапазон преобразуемых напряжений.

Регистр управления АЦП ADCON (адрес 0D8h) служит для управления входным мультиплексором, установки работы АЦП и фиксации бита готовности.

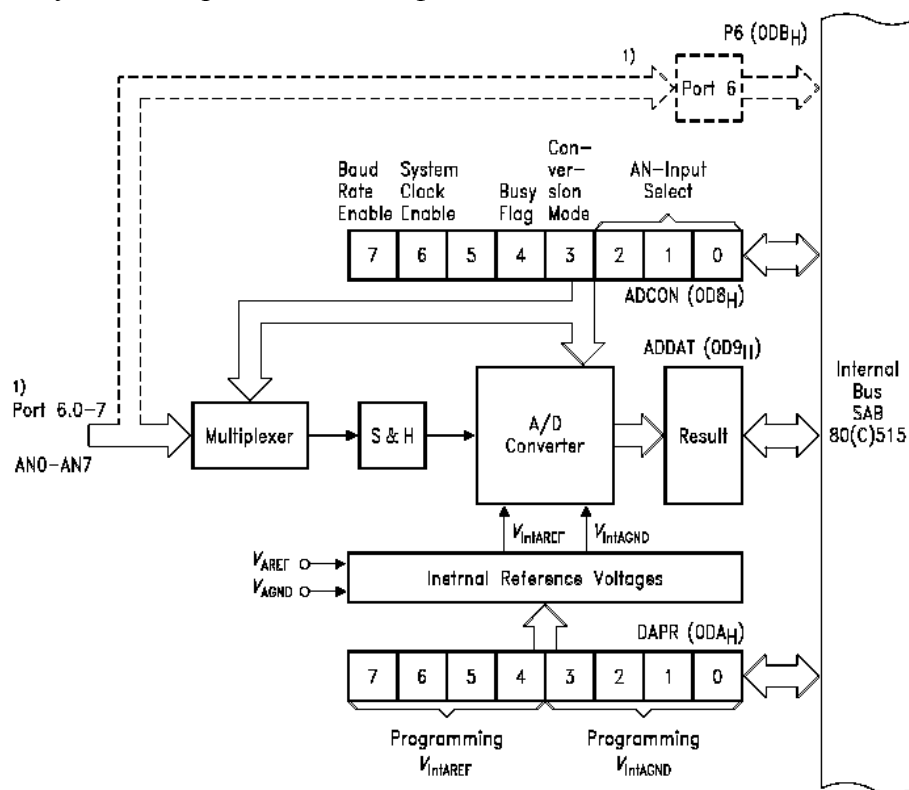


Рис. 8. Функциональная схема АЦП микроконтроллера SAB 80C515

Назначение битов регистра ADCON представлено в таблице 5.

Таблица 5. Назначение битов регистра ADCON

7	6	5	4	3	2	1	0	Бит
BD	CLK	0	BSY	ADM	MX2	MX1	MX0	
MX0	№ канала							
MX1								
MX2								
ADM	0 - режим одиночный 1- режим непрерывный							
BSY	Флаг занятости							
CLK	Подключение частоты 1/12 к выводу P1.6							
BD	Управление режимом последовательного порта							

Регистр данных АЦП ADDAT (адрес 0D9h) служит для хранения байта данных результата аналого-цифрового преобразования. Данные остаются в регистре до следующей перезаписи. Новые данные появляются в ADDAT в 15-ом машинном цикле после начала преобразования. Регистр ADDAT доступен и по чтению и по записи, поэтому, если АЦП не используется, то ADDAT может быть использован как регистр общего назначения.

Регистр специальных функций DAPR (адрес 0DAh) предназначен для программирования внутреннего опорного напряжения АЦП. При этом может изменяться как внутренняя "земля" относительно внешней, так и положительное напряжение также относительно внешней земли. Значения этих напряжений с шагом 1/16 от внешнего опорного напряжения задаются в регистре DAPR, причем четыре младших разряда задают напряжение внутренней "земли" IVAGND, а четыре старших - значение внутреннего опорного напряжения IVAREF. При внешнем опорном напряжении VAGND = 0, а VAREF = +5В шаг будет равен  $5/16 = 0,3125В$ .

На программируемое напряжение накладывается ограничение: разность между внутренними напряжениями IVAGND и IVAREF не должна быть меньше 1В, т.е. разность между кодами должна быть не менее четырех шагов. Для удобства при коде 00, записанном в DAPR, внутреннее напряжение равняется внешнему (0-5)В.

Возможность программирования внутреннего опорного напряжения позволяет согласовать его с диапазоном внешних преобразуемых напряжений и таким образом сохранять точность преобразования при различных диапазонах входных сигналов. Кроме того, при изменении входного сигнала в полном диапазоне 0-5 В возможно увеличение разрядности преобразования до 10-ти двоичных разрядов с помощью выполнения повторного преобразования, производимого после подгонки диапазона по результатам первого преобразования. При этом первое преобразование дает четыре старших разряда, которые должны быть записаны в младший полубайт регистра DAPR для подгонки диапазона, а следующее преобразование дает результат, из которого шесть старших разрядов являются значащими.

Процедура преобразования начинается с операции обращения к регистру DAPR. При этом, если не менялся диапазон опорных напряжений, то преобразование происходит за 15 машинных циклов, (при частоте 12МГц составляет 15 мкс). При этом в течении первых пяти машинных циклов (5 мкс) происходит выборка и фиксация преобразуемого сигнала, а затем уже его преобразование. Источник входного сигнала должен быть достаточно мощным, чтобы зарядить входную емкость 25пФ в течение 5 мкс. Перепрограммирование каждого внутреннего опорного напряжения требует дополнительно семь машинных циклов. Таким образом,

максимальное время преобразования при программировании обоих внутренних опорных напряжений составляет 29 машинных циклов т.е. 29 мкс.

В течение всего цикла преобразования бит занятости BSY находится в состоянии "1" и сбрасывается только по окончании преобразования, при этом также выставляется флаг запроса прерывания IADC. Если в АЦП установлен режим непрерывного преобразования, то он переходит к выполнению нового цикла преобразования, в противном случае останавливается до следующей операции "write-to-DAPR".

## 4. Выполнение работы

### 4.1. Задание №1

Разработать и выполнить программу, реализующую вывод на экран ЖКИ двухстрочного текста, в котором используются различные алфавиты (латиница и кириллица):

“Microcontrollers 2012

Дорофеев Тимофеев”

При разработке использовать модуль indic.asm.

Для вывода текстовой информации используется жидкокристаллический индикатор (ЖКИ). В его состав входит таблица преобразования ASCII-кода в графическую информацию, необходимую для вывода изображения. Таким образом, необходимо передать в ЖКИ ASCII-коды нужных символов, а также соблюсти правила установления управляющих и информационных сигналов. Общая схема соединений приведена на рис. 9:

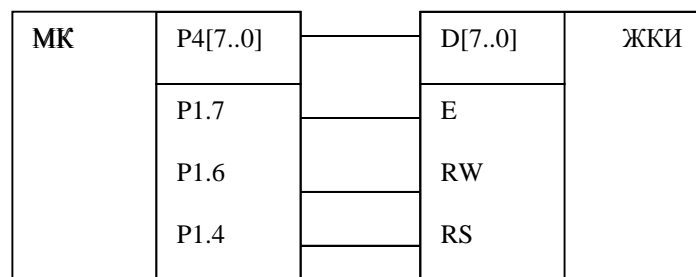


Рис. 9. Общая схема соединений МК 80C515 и модуля ЖКИ

#### 4.1.1. Схема программы вывода информации на ЖКИ



Рис. 10. Схема программы вывода информации на ЖКИ

#### 4.1.2. Текст программы

```

org 8800h

w1:      equ 20h          ;Команда или данные
w0:      equ 21h          ;Выводимая информация
p5:      equ F8h          ;Порт 4
indic:

        mov w1,#0          ;Вывод команд, инициализация ЖКИ
        mov w0,#38h        ;Установка 8-битного режима обмена
        lcall ind_wr        ;Вызов функции записи
        mov w0,#0Ch        ;Активация всех знакомест в ЖКИ
        lcall ind_wr
        mov w0,#80h        ;Установка адреса выводимых данных
        lcall ind_wr
        mov w1,#1          ;Передача данных
        mov dptr,#8800h

wr_str1:          ;Вывод в первую строку
        movx a,@dptr        ;Получение символа (Его ASCII-кода)
        mov w0,a            ;Передача в ЖКИ
        lcall ind_wr        ;Вызов функции записи
        inc dptr            ;Переход на следующий символ
        mov a,dpl           ;Проверка конца строки
        cjne a,#014h,wr_str1
        mov w1,#0          ;Команда
        mov w0,#C0h
        lcall ind_wr
        mov w1,#1

wr_str2:          ;Переходим на следующую строку
        movx a,@dptr
        mov w0,a
        lcall ind_wr
        inc dptr
        mov a,dpl
        cjne a,#028h,wr_str2
        ret

ind_wr:
        mov p5,w0          ;Выдача данных на шину данных
        setb p1.7          ;E = 1 разрешение программирования
        clr p1.6           ;RW = 0 запись
        mov a,w1           ;команда или данные
        mov c,acc.0
        mov p1.4,c         ;RS = асс
        lcall delay        ;Ожидание задержки
        clr p1.7           ;Стробирование записи
        lcall delay
        setb p1.7
        ret

delay:          ;Задержка
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        ret

        org 8800h          ;Установление адреса данных
                           ;Выводимые данные

str1: db 'Microcontrollers 2012'
str2: db 20h, 20h, 20h, E0h, 4Fh, 50h, 4Fh, E4h, 65h, 65h, B3h, 20h, 54h, B8h,
BCh, 6Fh, E4h, 65h, 65h, 20h

```

#### 4.1.3. План испытаний и результаты испытания программы

Проверка работоспособности программы заключалась в запуске программы и проверке вывода всех необходимых символов. Карту памяти приводить смысла не имеет, т.к. в данном задании не происходит изменения значений ячеек памяти.

В результате запуска программы на ЖКИ было отображено следующее:

Microcontrollers 2012

Дорофеев Тимофеев

#### 4.2. Задание №2

Разработать программу klav.asm, реализующую опрос клавиатуры и запись кодов состояния клавиш в ячейки памяти с адресами 30h ... 33h.

На стенде клавиши объединены в матрицу 4\*4. Для определения нажатой клавиши на стенде выведены входы и выходы матрицы клавиш. Входы соединены со строками матрицы, а выходы, соответственно, отвечают за столбцы матрицы. Для решения поставленной задачи было решено соединить входы матрицы с младшими разрядами порта p4 микроконтроллера, а выходы подать на старшие разряды того же параллельного порта. Общая схема соединений приведена на рис. 10:

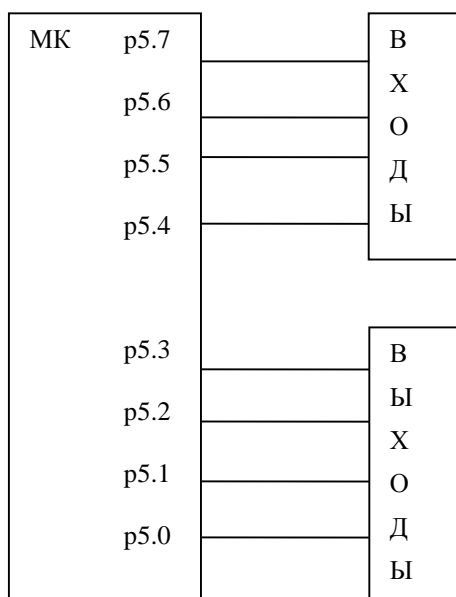


Рис. 11. Общая схема соединений МК 80C515 и клавиатуры

#### 4.2.1. Схема программы опроса клавиатуры



Рис. 12. Схема программы опроса клавиатуры

#### 4.2.2. Текст программы

```
org 8400h
p5: equ F8h

klav:    mov    r0, #30h
         orl    p5, #0fh
         mov    a, #7Fh

m1:      mov    p4, a
         mov    r2, a
         mov    a, p5
         anl    a, #0fh
         mov    @r0, a
         inc    r0
         mov    a, r2
         rr     a
         cjne   a, #f7h, m1

         ret
```



#### 4.2.3. План испытаний и результаты испытания программы

Для проверки работоспособности программы было проведено несколько запусков программы с заранее нажатой клавишей. Для разных экспериментов брались различные клавиши из разных строк и столбцов.

Таблица 6. Результаты проведения экспериментов

Номер клавиши	Входное воздействие Выходы порта P4				Состояние карты памяти			
	P4.7	P4.6	P4.5	P4.4	30h	31h	32h	33h
1	0	0	0	<b>1</b>	0Fh	0Fh	0Fh	<b>07h</b>
2	0	0	0	<b>1</b>	0Fh	0Fh	0Fh	<b>0Bh</b>
3	0	0	0	<b>1</b>	0Fh	0Fh	0Fh	<b>0Dh</b>
4	0	0	0	<b>1</b>	0Fh	0Fh	0Fh	<b>0Eh</b>
5	0	0	<b>1</b>	0	0Fh	0Fh	<b>07h</b>	0Fh
6	0	0	<b>1</b>	0	0Fh	0Fh	<b>0Bh</b>	0Fh
7	0	0	<b>1</b>	0	0Fh	0Fh	<b>0Dh</b>	0Fh
8	0	0	<b>1</b>	0	0Fh	0Fh	<b>0Eh</b>	0Fh
9	0	<b>1</b>	0	0	0Fh	<b>07h</b>	0Fh	0Fh
10	0	<b>1</b>	0	0	0Fh	<b>0Bh</b>	0Fh	0Fh
11	0	<b>1</b>	0	0	0Fh	<b>0Dh</b>	0Fh	0Fh
12	0	<b>1</b>	0	0	0Fh	<b>0Eh</b>	0Fh	0Fh
13	<b>1</b>	0	0	0	<b>07h</b>	0Fh	0Fh	0Fh
14	<b>1</b>	0	0	0	<b>0Bh</b>	0Fh	0Fh	0Fh
15	<b>1</b>	0	0	0	<b>0Dh</b>	0Fh	0Fh	0Fh
16	<b>1</b>	0	0	0	<b>0Eh</b>	0Fh	0Fh	0Fh
Не нажата	0	0	0	0	0Fh	0Fh	0Fh	0Fh

### 4.3. Задание №3

Модифицировать программу «опрос клавиатуры» (п.2), дополнив её модулем, формирующим номер нажатой клавиши. Присвоить ей имя klav1.asm.

Структурная схема соединения связей микроконтроллера с периферийными устройствами не изменится. Необходимо добавить в программу блок, анализирующий позицию нажатой клавиши и выделяющий ее номер. Для данного задания была выбрана следующая нумерация клавиш:

Таблица 7. Нумерация клавиш

16	15	14	13
12	11	10	9
8	7	6	5
4	3	2	1

#### 4.3.1. Схема программы

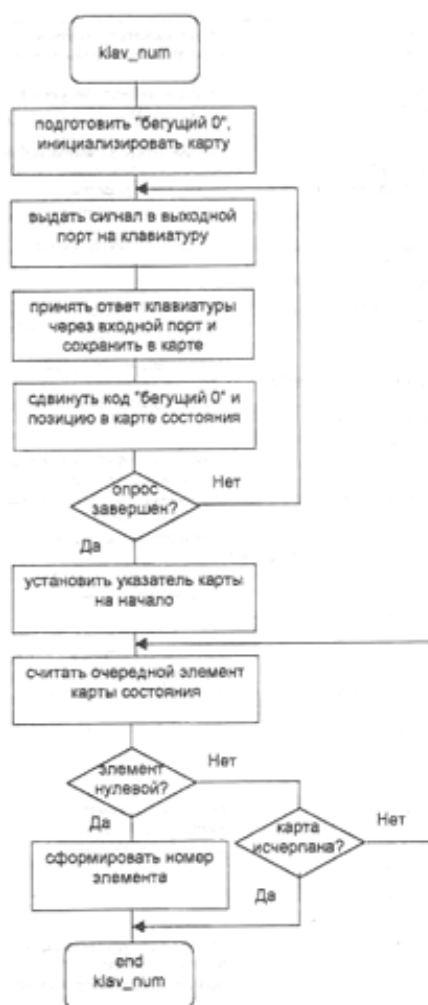


Рис. 13. Алгоритм работы программы

#### 4.3.2. Текст программы

```
include C:\SHELL51\ASMS\4081_4\klav.asm
```

```
        mov     14h, #ff
        clr     a
        add     a, 30h
        add     a, 31h
        add     a, 32h
        add     a, 33h
        clr     c
        subb    a, #2D
        jc      printResult

        mov     r4, #1h
        mov     r5, #0h

        mov     a, 33h
        swap    a
        orl     a, 32h
        setb    c

k1:      rlc     a
        jnc     exit
        inc     r4
        inc     r5
        cjne    r5, #8h, k1

        mov     a, 31h
        swap    a
        orl     a, 30h
        mov     r5, #0h
        setb    c

k2:      rlc     a
        jnc     exit
        inc     r4
        inc     r5
        cjne    r5, #8h, k2

exit:    clr     c
        mov     a, r4
        subb    a, #11h
        jnz     printResult
        mov     r4, #0h

printResult:
        mov     34h, r4
        ret
```

### 4.3.3. План испытаний и результаты испытания программы

Карта памяти клавиатуры изображена на рис. 14 и рис. 15. Она по адресам 30h . . . 33h отображает нажатие клавиш – в первой строке (кнопка '1', рис. 14) и во второй и третьей строках (кнопки '1' и '13', рис. 15).

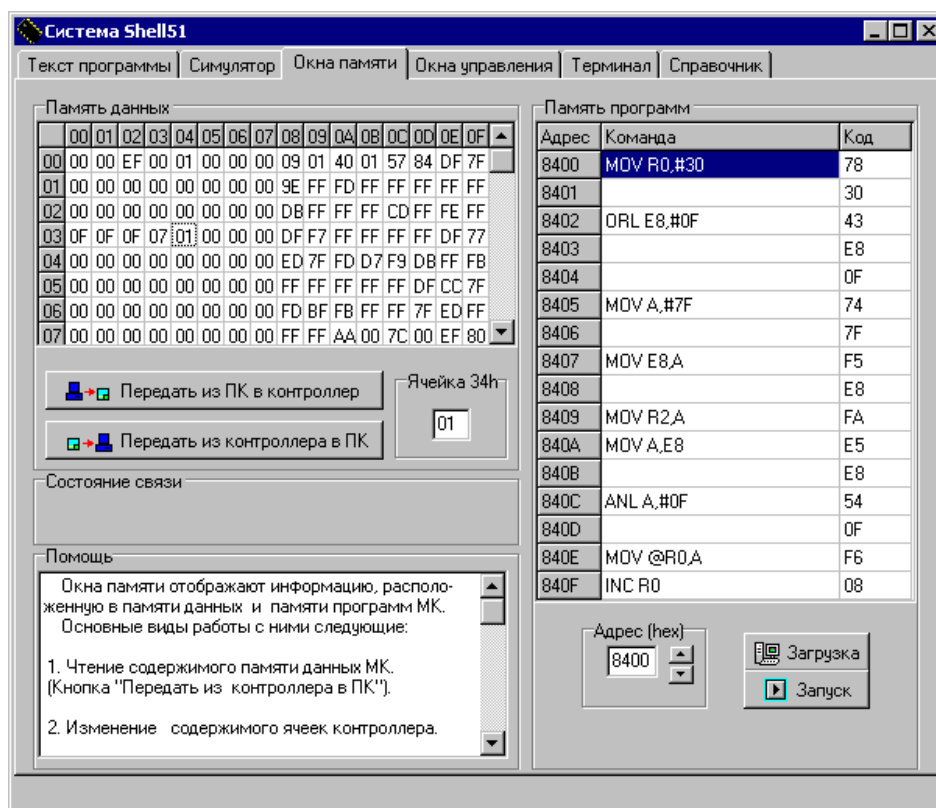


Рис. 14. Карта памяти клавиатуры (нажата кнопка '1')

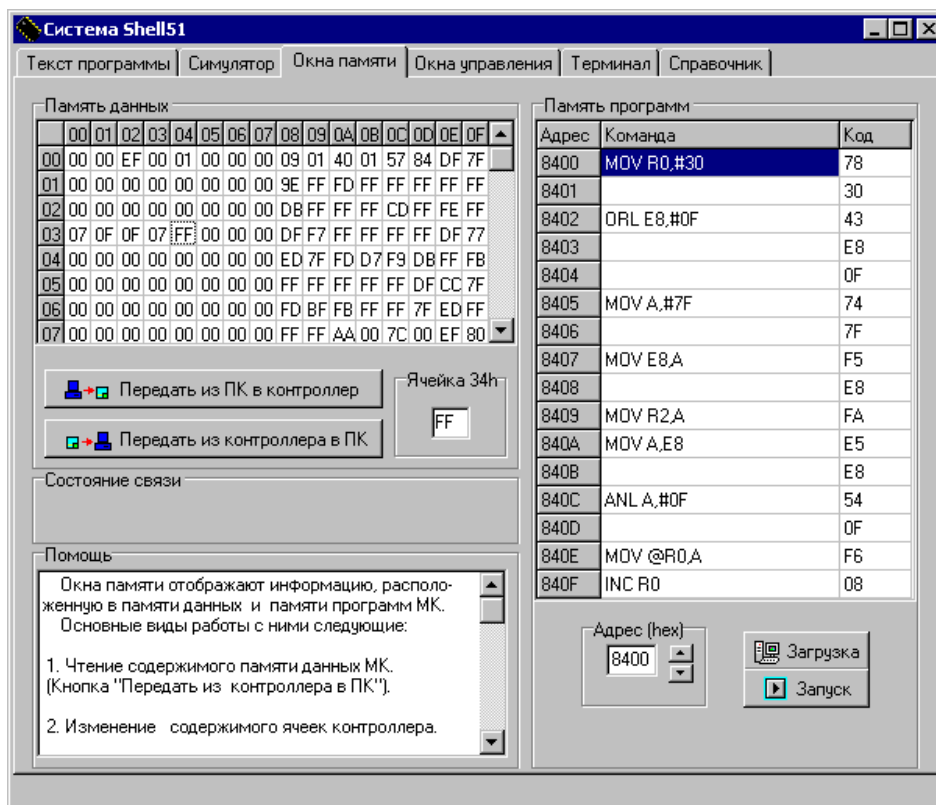


Рис. 15. Карта памяти клавиатуры (нажаты кнопки '1' и '13').

#### 4.4. Задание №4

Разработать программу, отображающую на экране модуля ЖКИ номер нажатой клавиши, используя программы indic.asm и klav1.asm, подключив их к разрабатываемой программе директивой include<имя файла>.

Для выполнения поставленной задачи необходимо было объединить две работающие процедуры, обеспечить согласованное пользование памятью, поскольку изначально процедуры могли использовать одни и те же ячейки памяти, что могло привести к некорректной работе. В процедуру вывода информации на ЖКИ необходимо добавить блок перевода числа в ASCII-коды. Также необходимо было предусмотреть новое соединение микроконтроллера с периферийными устройствами. Общая схема соединений приведена на рис. 16.

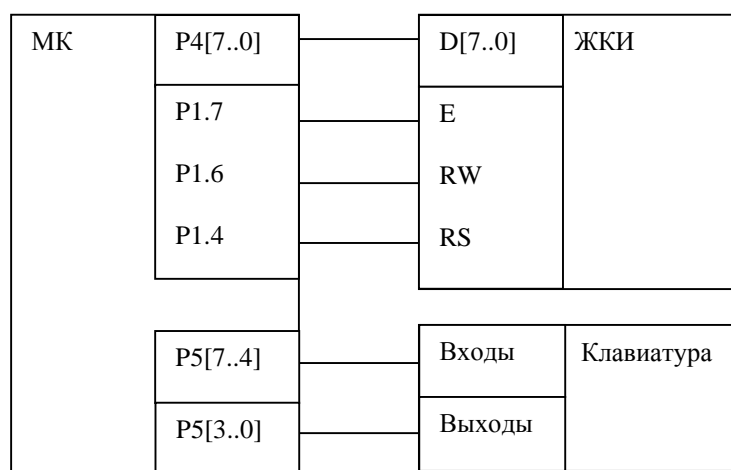


Рис. 16. Схема подключения модуля ЖКИ и клавиатуры к МК

##### 4.4.1. Текст программы

###### Файл incl.asm:

```
include C:\SHELL51\ASMS\4081_4\klav.asm
include C:\SHELL51\ASMS\4081_4\klav1.asm

mov    a, #0h

; indic
w1:    equ    20h
w0:    equ    21h

indic:    mov    w1, #0
          mov    w0, #38h
          lcall  ind_wr
          mov    w0, #0Ch
          lcall  ind_wr
          mov    w0, #80h
          lcall  ind_wr
          mov    w1, #1
          mov    dptr, #FFD0h

wr_str1:    movx  a, @dptr
          mov    w0, a
          lcall  ind_wr
          inc    dptr
          mov    a, dpl
```



#### 4.4.2. План испытаний и результаты испытания программы

Проверка работоспособности программы заключалась в запуске программы при нажатой клавише и наблюдение результатов на ЖКИ.

Нумерация клавиш совпадает с нумерацией в п.3.

Нажата клавиша 1 – на ЖКИ: 1.

Нажата клавиша 16 – на ЖКИ: 16...

#### 4.5. Задание №5

Разработать программу вывода на ЖКИ цифровых кодов аналоговых сигналов параллельно с двух источников: потенциометра и выхода интегратора, подключенного к генератору периодического сигнала.

Для выполнения поставленной задачи необходимо разработать новую структуру связей микроконтроллера с периферийными устройствами. В программе необходимо организовать бесконечный цикл опроса АЦП и выдачи результатов на ЖКИ. Один раз в начале работы необходимо настроить АЦП на работу. АЦП имеет два опорных напряжения, между которыми происходит квантование для измерения сигнала. Поскольку интегратор может выдавать сигнал от 0В до 5В, было решено установить опорные напряжения соответственно 0В и 5В, что соответствует записи 00h в регистр DAPR (0DAh). Для получения точного значения уровня напряжения, подаваемого на АЦП необходимо проследить этот сигнал на осциллографе и замерить напряжение в вольтах. Схема соединения АЦП с потенциометром и интегратором приведена на рис. 14.

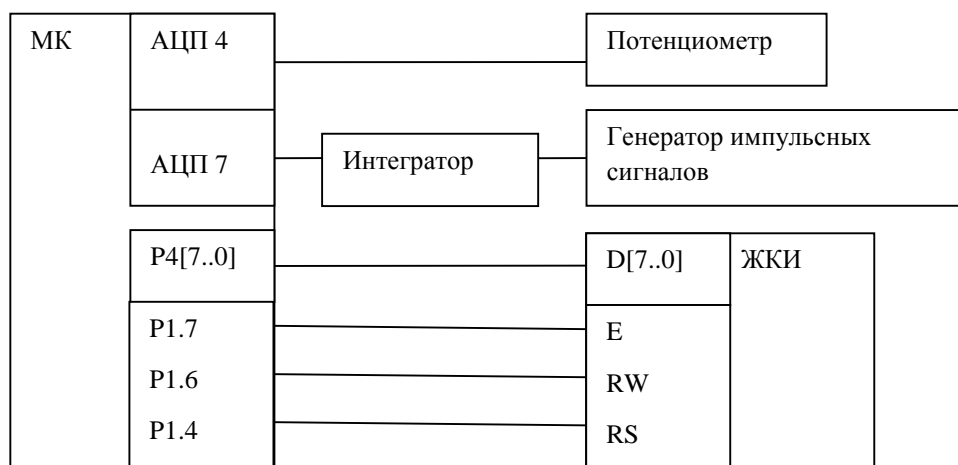


Рис. 17. Схема соединения АЦП МК с двумя источниками сигналов

#### 4.5.1. Схема программы опроса датчиков аналоговых сигналов

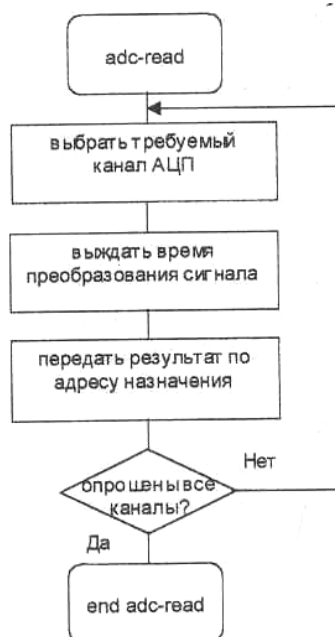


Рис. 18. Схема программы опроса датчиков аналоговых сигналов

#### 4.5.2. Текст программы

##### Файл incl1.asm:

```
org 8100h
lcall adc
lcall daa
lcall indic
ret
include C:\shell151\asms\4081_13\adc.asm
include C:\shell151\asms\4081_13\indic1.asm
include C:\shell151\asms\4081_13\daa.asm
```

##### Файл adc.asm:

```
adc:
ADCON: equ D8h
ADDAT: equ D9h
DAPR: equ Dah
mov A, #00000101 ;режим работы - одиночное преобр-е, 3й канал АЦП
anl ADCON, #E0h
orl ADCON, A
mov DAPR, #0h
mov R7, #15
m1: djnz R7, m1
mov a, ADDAT
mov 40h, ADDAT
ret
```



### Файл daa.asm:

```
daa:
    mov dptr,#8814h
    mov b, #64h
    div ab
    add a, #30h           ;Переводим в ASCII-код
    movx @dptr, a
    inc dptr
    mov a, b
    mov b, #Ah
    div ab
    add a, #30h
    movx @dptr, a
    inc dptr
    mov a, b
    add a, #30h
    movx @dptr, a
    ret
```

### Файл indic1.asm:

Здесь используется файл indic.asm из п.1:

```
w1:      equ 20h
w0:      equ 21h
p5:      equ F8h
indic:
    mov w1,#0
    mov w0,#38h
    lcall ind_wr
    mov w0,#0Ch
    lcall ind_wr
    mov w0,#80h
    lcall ind_wr
    mov w1,#1
    mov dptr,#8800h

wr_str1:
    movx a,@dptr
    mov w0,a
    lcall ind_wr
    inc dptr
    mov a,dpl
    cjne a,#014h,wr_str1
    mov w1,#0
    mov w0,#C0h
    lcall ind_wr
    mov w1,#1

wr_str2:
    movx a,@dptr
    mov w0,a
    lcall ind_wr
    inc dptr
    mov a,dpl
    cjne a,#028h,wr_str2
    ret

ind_wr:
    mov p5,w0
    setb pl.7
    clr pl.6
    mov a,w1
    mov c,acc.0
    mov pl.4,c
    lcall delay
    clr pl.7
    lcall delay
    setb pl.7
```

```

        ret
delay:
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        nop
        ret
str1: db 'Napryazhenie:      '
str2: db '

```

#### 4.5.3. План испытаний и результаты испытания программы

В результате правильного соединения проводов и запуска программы на экране ЖКИ отобразились числа, которые изменялись при изменении сигнала на входе АЦП, при поворотах регуляторов потенциометра и генератора периодического сигнала. В ходе проверки работоспособности было проведено несколько измерений: фиксировались числа на ЖКИ и измерялось напряжение при том же положении регуляторов. Результаты измерений:

Для потенциометра:

1) ЖКИ: 252                      Осциллограф: 5.0 В.

2) ЖКИ: 146                      Осциллограф: 3.0 В.

С учетом выбора опорных напряжений 0 – 5 В можно построить следующую таблицу для потенциометра:

*Таблица 8. Таблица диапазонов потенциометра*

Диапазон опорных напряжений	Напряжение датчика, измеренное осциллографом	Цифровой эквивалент измеренного значения	Результат преобразования (цифровой код ЖКИ, зафиксированный на выходе АЦП)	Погрешность измерения
0 – 5 В (DAPR = 0)	5.0 В	250	252	2
0 – 5 В (DAPR = 0)	4.0 В	201	197	4
0 – 5 В (DAPR = 0)	3.0 В	149	145	4
0 – 5 В (DAPR = 0)	2.5 В	127	129	2

Для генератора импульсных сигналов через интегратор:

1) ЖКИ: 69                      Осциллограф: 1.3 В.

2) ЖКИ: 138                      Осциллограф: 2.7 В.

Для интегратора измерения производились в двух диапазонах, результат в более узком из них логично оказался более точным:

Таблица 9. Таблица диапазонов интегратора

Диапазон опорных напряжений	Напряжение датчика, измеренное осциллографом	Цифровой эквивалент измеренного значения	Результат преобразования (цифровой код ЖКИ, зафиксированный на выходе АЦП)	Погрешность измерения
0 – 5 В (DAPR = 0)	1.3 В	66	69	3
0 – 5 В (DAPR = 0)	2.7 В	138	137	1
0 – 5 В (DAPR = 0)	2.4 В	123	127	4
1,25 – 2,8125 В (DAPR = 94)	1.4 В	25	24	1
1,25 – 2,8125 В (DAPR = 94)	2.3 В	172	172	0
1,25 – 2,8125 В (DAPR = 94)	2.7 В	237	239	2

С учетом разности напряжений 5В и градации измерений АЦП в 8 двоичных разрядов имеем величину квантования  $d = \frac{5.12 \text{ В}}{256} = 0.0195 \text{ В} \approx 20 \text{ мВ}$ . Величина соответствующего напряжения вычисляется как  $N_{\text{изм}} * d$ . Величина напряжения, вычисленного с помощью осциллографа, находится просто:  $D * M$ , как масштаб на высоту отклонения от нуля. Полученная погрешность имеет несколько составляющих: погрешность квантования 20 мВ, инструментальную погрешность, а также человеческий фактор.

Для диапазона (1.25 – 2.8125 В) получаем расчётное значение числового эквивалента:  $256 \frac{U - 1.25}{2.8125 - 1.25}$ . При работе в этом диапазоне повышается точность измерений, так как погрешность квантования становится равна:  $d = \frac{1.5625 \text{ В}}{256} = 0.0061 \text{ В} = 6 \text{ мВ}$ .

В целом же, можно сказать, что АЦП данного класса может использоваться в измерительных системах, не требующих высокой точности.

#### 4.6. Задание №6

Модифицировать программу п.5 так, чтобы, наряду с отображением цифровых кодов аналоговых сигналов от двух датчиков, можно было бы продемонстрировать скорость обработки изменяющегося во времени аналогового сигнала с выхода выбранного пользователем датчика в цифровой код на экране инструментальной ЭВМ.

Схема соединений аналогична представленной в предыдущем пункте.

Модифицированная программа должна обеспечить совместную работу пользовательской программы и программного модуля однократного сеанса связи с ЭВМ (подпрограммы

монитора), вызываемого командой `lcall 128h`. Для этого в циклический участок модифицированной программы необходимо вставить указанную команду `lcall 128h`.

#### 4.6.1. Текст программы

Текст программы `incl_adc.asm` включил в себя вызов упомянутой команды `lcall 128h` и заикливание выполнения всего комплекса разработанных программ.

Файл `incl_adc.asm`:

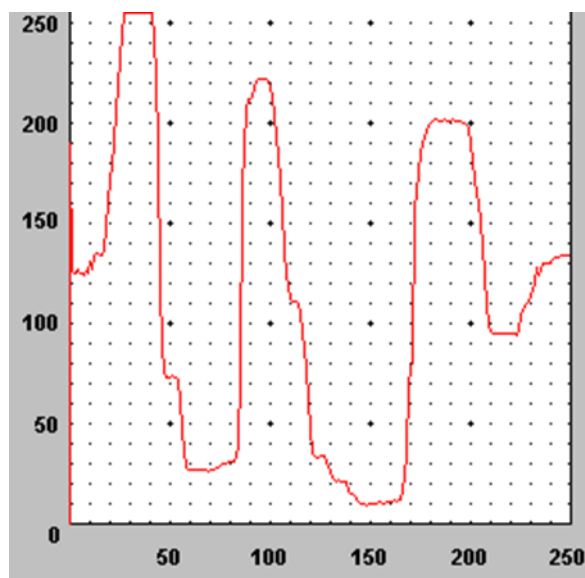
```
start:                org 8400h
                    lcall adc
                    lcall daa
                    lcall indic
                    lcall 128h
                    ljmp start
                    ret

include C:\shell151\asms\4081_13\adc.asm
include C:\shell151\asms\4081_13\indic1.asm
include C:\shell151\asms\4081_13\daa.asm
```

Тексты остальных программ не менялись (см. п.5).

#### 4.6.2. План испытаний и результаты испытания программы

Результатом запуска программы стало отображение цифровых кодов аналоговых сигналов, подаваемых на входы АЦП. Связь с ЭВМ осуществлялась с помощью вызова специальной функции `lcall 128h`. В результате были получены следующие графики изменения цифровых кодов:



*Рис.19. Потенциометр*

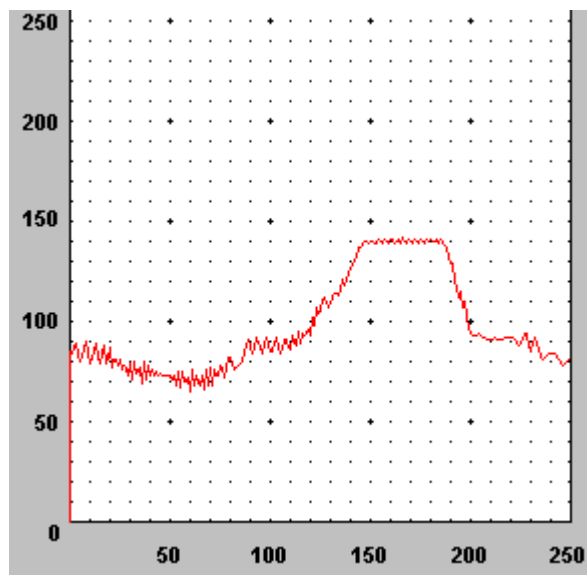


Рис.20. Генератор прямоугольных импульсов через интегратор при  $DAPR = 0$

Видно, что потенциометр позволяет получить любой код или любое напряжение в диапазоне 0-5 В. Что же касается сигнала с генератора, пропущенного через интегратор, то он может выдавать сигнал в более узком диапазоне. Диапазон сигнала с генератора, пропущенного через интегратор, можно настраивать с помощью изменения значения регистра  $DAPR$ .

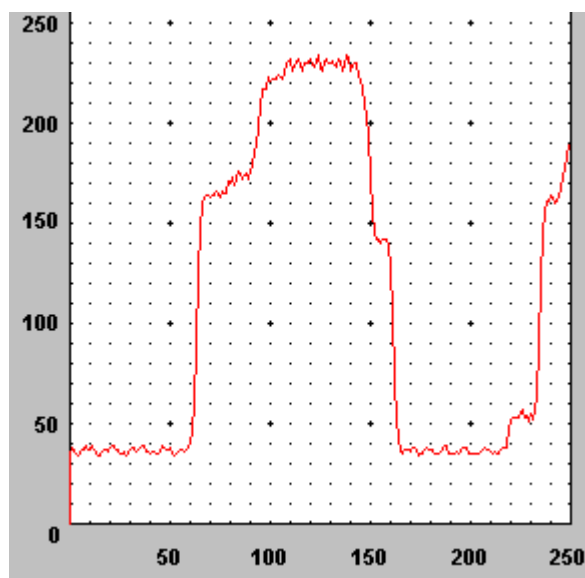


Рис.21. Генератор прямоугольных импульсов через интегратор при  $DAPR = A3h$

## 5. Выводы

В данной лабораторной работе были изучены возможности микроконтроллера по работе с параллельными портами ввода-вывода. С помощью портов P1, P4 и P5 осуществлялось взаимодействие микроконтроллера с периферийными устройствами, входящими в состав используемого стенда.

Для визуального отображения результатов работы микроконтроллера имеется жидкокристаллический индикатор. Существенным плюсом данного устройства является информационное взаимодействие символами по их ASCII-кодам, что сильно облегчает программисту вывод нужной информации, поскольку не требуется переводить символ в кодовую последовательность для отображения каждого пикселя.

В число периферийных устройств входит клавиатура. Она имеет 4 входа, отвечающих за строки матрицы кнопок, и 4 выхода, отвечающих за столбцы матрицы кнопок. Такая клавиатура может использоваться для ручного управления средней сложности, например калькулятор, пульт управления несложным устройством. Микроконтроллер хорошо взаимодействует с этим устройством, что подтверждает простое программное обеспечение и использование всего одного порта ввода вывода.

Также в инструментальный набор стенда включен восьмиканальный АЦП. Наличие АЦП позволяет строить системы контроля аналоговых величин, которые представлены в виде напряжения. Нельзя не отметить, что процедура преобразования является сравнительно длинной (до 25 мкс), поэтому перед считыванием результата преобразования необходимо использовать соответствующую временную задержку.

На основе проделанных измерений, можно сказать, что минимальная погрешность измерения АЦП примерно равна 4.9 мВ, чего достаточно для измерительных систем, не требующих высокой точности. Точность АЦП можно настроить под конкретные нужды, управляя диапазонами опорных напряжений.

Стоит отметить, что мы испытывали по началу проблему с тем, что Ассемблер MCS-51 не знает имен P4 и P5 расширенного пространства ввода/вывода МК SAB80C515. Но в ходе работы поняли, что это не мешает ее выполнению, поэтому для использования в программе имён P4 и P5 надо в начале объявить порты как `r4 equ 0E8h` и `r5 equ F8h`.

Для составления продвинутых программ важно наличие директивы, заключенной в `include`. Однако необходимо помнить о накладываемых ограничениях, связанных с 8-символьными именами, одинаковыми метками, меткой конца программы, невозможностью использования кириллических символов и использованием абсолютных адресов.

Также полезной является возможность среды Shell-51 строить графики изменения во времени значения переменной, находящейся во внутренней памяти. Для нас эта функция оказалась очень полезной, так как она позволяет лучше понять работу интегратора.