

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ  
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

**ОТОБРАЖЕНИЕ ИНФОРМАЦИИ  
НА ЖИДКОКРИСТАЛЛИЧЕСКИХ  
ИНДИКАТОРАХ С КОНТРОЛЛЕРОМ  
HD-44780**

**Самара 2015**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ  
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

ОТОБРАЖЕНИЕ ИНФОРМАЦИИ  
НА ЖИДКОКРИСТАЛЛИЧЕСКИХ  
ИНДИКАТОРАХ С КОНТРОЛЛЕРОМ  
HD-44780

Составитель *В.Г. Иоффе*

Самара 2015

УДК СГАУ:681.32

*Составитель: В.Г. Иоффе*

Рецензент: к.т.н., доц. А.В. Полулех

**Отображение информации на жидкокристаллических индикаторах с контроллером HD-44780** [Электронный ресурс]: метод. указания / сост. *В. Г. Иоффе*. М-во образования и науки РФ, Самар. гос. аэрокосм. ун-т им. С.П. Королева (нац. исслед. ун-т). Электрон. текстовые и граф. дан.(0,704 Мбайт).- Самара, 2015.- 36 с., ил. 1 эл.опт. диск (CD-ROM).

Методические указания содержат описание структуры и программной модели контроллера HD44780, предназначенного для управления алфавитно-цифровыми жидко-кристаллическими индикаторами, предложена методика программирования HD44780. Приведены примеры программ на ассемблере 8051.

Предназначены для подготовки бакалавров, обучающихся по направлению 09.03.01 –”Информатика и вычислительная техника “.

Электронные методические указания разработаны на кафедре информационных систем и технологий.

Стр.36, Ил.8, Табл.9, Библ.3 назв.

УДК СГАУ:681.32

© Самарский государственный  
аэрокосмический университет, 2015

## Содержание

Введение	5
1. Структура контроллера HD44780	8
2. Программная модель контроллера	11
3. Взаимодействие с микропроцессорной системой	19
4. Программирование контроллера на ассемблере 8051	25
5. Порядок выполнения лабораторных работ	33
6. Контрольные вопросы	34
7. Список использованных источников	35

## Введение

В микропроцессорной технике наибольшее распространение получили полупроводниковые индикаторы, принцип действия которых основан на преобразовании энергии электрического поля в световой поток, и жидкокристаллические индикаторы (ЖКИ), которые модулируют внешний световой поток под действием электрического, магнитного полей или других воздействий. В последних разработках ЖКИ находят наибольшее применение, так как они обладают низким энергопотреблением и большими информационными возможностями.

Работа ЖКИ основана на свойствах жидких кристаллов, представляющих собой органическое соединение, находящееся в промежуточном состоянии между твердым (кристаллическим) и изотропно жидким. Молекулы жидкого кристалла под действием электрического, магнитного полей и/или температуры становятся способными ориентироваться определенным образом. Вещество, прозрачное в отсутствие поля, становится непрозрачным при его воздействии.

Различают два основных вида жидких кристаллов: холестерики ( реагируют на температуру) и нематики, реагирующие на электрическое поле.

ЖКИ свет не излучают, поэтому для отображения информации необходим источник отраженного или проходящего света.

Конструктивно ЖКИ представляют собой две прозрачные стеклянные пластины, между которыми находится жидкокристаллическое вещество. На внутренней поверхности пластин расположены электроды, представляющие собой прозрачные электропроводящие пленки. На верхней пластине расположены электроды ( сегменты ) требуемой формы, на нижней - общий электрод. На электроды подают управляющие напряжения.

Индикаторы, работающие в отраженном свете, имеют нижний электрод с большим коэффициентом отражения. В качестве источника падающего света служит естественное освещение. Чем ярче освещение, тем контрастней изображение.

В условиях низкой освещенности применяют ЖКИ , работающие в проходящем свете. У них нижний электрод прозрачный, а под нижней стеклянной пластиной расположен источник света и матово-черный экран. При подаче напряжения на электроды прозрачность ЖКИ нарушается и в проходящем свете отображается необходимый знак. В качестве источника света используются светодиодные, электролюминисцентные и флюоресцентные панели. Недостатком последних двух панелей является необходимость использования повышенного напряжения.[ 1 ]

Цвет и яркость свечения индикатора зависит от цвета и яркости источника проходящего света и свойств используемых жидких кристаллов.

К недостаткам ЖКИ следует отнести небольшой угол обзора, невысокое быстродействие и низкую контрастность при недостаточном освещении.

Для отображения информации необходимы схемы управления электродами, формирующие многофазные последовательности импульсов требуемой амплитуды, и средства отображения информации заданного вида. Решение этих задач наиболее целесообразно выполнять с использованием модулей , в состав которых входят ЖКИ и специализированный контроллер. В этом случае синтез отображаемой информации и формирование сигналов выполняет программно управляемый контроллер.

По типу индикаторов модули подразделяются на семи-сегментные, символьные ( алфавитно-цифровые ), графические.

Алфавитно-цифровые модули (АЦМ) представляют собой недорогое и удобное средство, позволяющее сэкономить время и ресурсы при разработке новых изделий, обеспечивая отображение большого объема информации при хорошей различимости и низком энергопотреблении. Эти модули можно встретить в самых разнообразных устройствах: измерительных приборах, медицинском оборудовании, промышленном и технологическом оборудовании, офисной технике – принтерах, телефонных, факсимильных и копировальных аппаратах. Возможность оснащения ЖКИ модулей задней подсветкой позволяет эксплуатировать их в условиях с пониженной или нулевой освещенностью, а исполнение с расширенным диапазоном температур (-20 С...70 С) – в

сложных эксплуатационных условиях, в том числе в переносной, полевой, бортовой аппаратуре.

Основными характеристиками АЦМ являются:

- размер символа (как правило, 5x8 точек);
- формат экрана ( количество символов x количество строк). Минимальный- 8x2 или 16x1, максимальный -40x4;
- размер модуля;
- параметры жидкого кристалла ( нематики с твист-эффектом TN, супернематики с твист-эффектом STN и так далее);
- режим работы ( на отражение, на просвет, на полупросвет);
- тип подсветки ( светодиодная, электролюминисцентная, флюоресцентная с холодным катодом);
- цвет подсветки и жидкого кристалла;
- варианты знакогенератора: англо/русский, англо/европейский, англо/азиатский и т.д.;
- температурный диапазон : (0 -60) град.С, (-20 - +70)град.С;
- напряжение питания, ток потребления;
- внешний интерфейс контроллера (параллельный, последовательный).

Данные методические указания посвящены алфавитно – цифровым ( символьным) модулям ЖКИ на основе параллельного контроллера HD 44780. Контроллер HD 44780 фирмы Hitachi фактически является промышленным стандартом при производстве АЦМ. Фирмы Epson, Toshiba, Sanyo, Samsung, Philips и другие выпускают аналоги этого контроллера или микросхемы, совместимые с ним по типу аппаратного интерфейса и программной модели.

На основе HD44780 разработан ряд АЦМ с последовательными интерфейсами. Например, LCD 216S, у которого может быть установлен один из трёх интерфейсов: RS-232, I2C, SPI, CE-110 - контроллер клавиатуры и индикации с I2C. Эти АЦМ дополнительно содержат блоки связи контроллера HD44780 с соответствующими интерфейсами.

## 1. Структура контроллера HD44780

Упрощенная структурная схема контроллера представлена на рис. 1.

В функции контроллера входит:

– организация взаимодействия с микропроцессорной системой МПС;

– прием и дешифрация команд;

– хранение отображаемых символов в видеопамяти;

– формирование изображения;

– управление жидкокристаллическим индикатором.

Контроллер состоит из блока связи с интерфейсом БСИ, процессорного блока ПБ, ОЗУ данных (видеопамять), знакогенератора ЗГ и блока формирования управляющих сигналов БУС.

Обмен информацией с микропроцессорной системой производится через шину данных DB7 –DB0 БСИ. В зависимости от особенностей МПС возможно использование 8-разрядной или 4-разрядной шины данных. В последнем случае цикл обмена увеличивается в два раза. Сигнал RS является идентификатором команды (RS=0) или данных (RS=1). Направление передачи информации определяет сигнал чтения/записи  $R/\overline{W}$ . Сигнал E обеспечивает синхронизацию работы контроллера и МПС. Временные диаграммы работы контроллера приведены на рисунках 5,6.

В состав ПБ входят регистр команд IR, регистр данных DR, счетчик адреса AC, содержащий в старшем разряде флаг готовности BF, и управляющий автомат УА.

ПБ работает под управлением команд, поступающих в регистр IR. Информация, записанная в IR, преобразуется УА в последовательность управляющих сигналов, которые управляют видеопамятью, знакогенератором, БУС.

Чтение регистра IR обеспечивает доступ к текущему значению адреса ( разряды 0 -6 ) и флагу готовности BF.

Данные , записанные в регистр DR , в зависимости от команды могут быть переданы в видеопамять DDRAM или ОЗУ знакогенератора по адресу указанному в AC.

Видеопамять **DDRAM**, имеющая общий объем 80 байтов, предназначена для хранения символов, отображаемых на ЖКИ.

Видеопамять организована в две строки по 40 символов в каждой. Эта привязка является жесткой и не подлежит изменению. Независимо от того, сколько реальных строк будет иметь конкретный АЦМ, например, 80x1 или 20x4 адресация видеопамати всегда производится как к двум строкам по 40 символов.

Знакогенератор состоит из ПЗУ **CGROM**, в котором записаны ASCII коды соответствующего алфавита, и перепрограммируемого ОЗУ **CGRAM**, предназначенного для создания требуемых символов, отсутствующих в ПЗУ. Имеется возможность задать начертание восьми символов с матрицей 5x7 или четырех с матрицей 5x10.

**CGROM** имеет два набора изображаемых символов: 204 символа размером 5x8 точек и 32 символа размером 5x11 точек.

Синхронизация работы контроллера выполняется внутренним генератором с частотой 270 КГц ( на схеме не показан ).

БУС включает драйверы подложек и сегментов, выполняющие непосредственное управление электродами ЖКИ, сдвиговых регистров и схем, формирующих требуемые временные диаграммы.

В АЦМ используется динамическая индикация, применение которой позволяет значительно сократить аппаратные затраты БУС.

ЖКИ организован как матрица, состоящая, в зависимости от режима работы, из 8-ми (одна строка символов 5x7 точек), 11-ти (одна строка символов 5x10 точек) или 16-ти (две строки символов 5x7 точек) строк по 200 сегментов (каждая строка насчитывает по 40 символов) в каждой.

Собственный драйвер контроллера **HD44780** имеет только 40 выходов (SEG1...SEG40) и самостоятельно может поддерживать только 8-ми символьные ЖКИ. Это означает, что АЦМ

форматов до 8x2 реализованы на одной единственной микросхеме **HD44780**, модули, имеющие большее количество символов, содержат дополнительные микросхемы драйверов, например **HD44100**, каждая из которых дополнительно представляет управление еще сорока сегментами

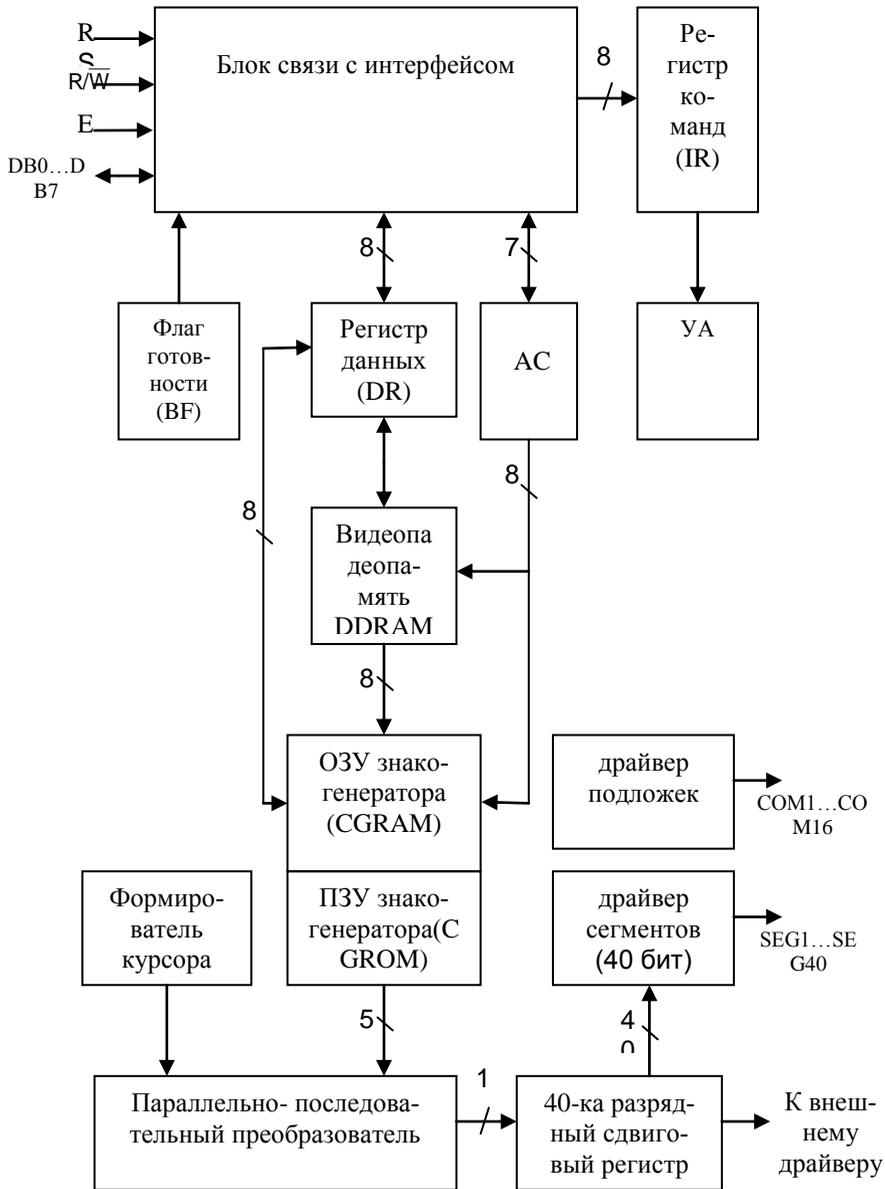


Рис.1- Упрощенная структурная схема контроллера

АЦМ формата 16x1 также реализованы с помощью одной единственной микросхемы **HD44780**, но одна 16-ти символьная строка в них фактически составлена из двух 8-ми символьных [ 2 ].

Такая организация усложняет программирование, так как символьная строка составлена фактически из двух восьми-символьных, но для реализации требуется только один контроллер.

## 2. Программная модель контроллера

К числу программно доступных регистров относятся регистр данных **DR**, регистр команд **IR**, видеопамять **DDRAM**, **ОЗУ(CG RAM)** и **ПЗУ( CGROM)** знакогенератора, счетчик адреса **AC**, флаг готовности **BF**. [ 1, 2 ]

Табл. 1. Система команд контроллера **HD44780**.

Код										Описание ко-манды	Вре-мя ис-пол-не-ния
R S	R/ W	D 7	D 6	D 5	D 4	D 3	D 2	D 1	D 0		
0	0	0	0	1	D/L	N	F	*	*	Определение параметров развертки и ширины шины данных	40 мкс
0	0	0	0	0	0	1	D	C	B	Вкл./выкл. дисплея; управление курсором	40 мкс
0	0	0	0	0	0	0	1	I/D	S	Направление сдвига курсора или дисплея	40 мкс
0	0	0	0	0	0	0	0	0	1	Очистка дисплея, <b>AC=0</b> , адресация <b>AC</b> на <b>DDRAM</b>	82 мкс 1.64 мс
0	0	0	0	0	0	0	0	1	*	Возврат в	40

											начальное положение. <b>AC = 0</b> , адресация на <b>DDRAM</b> , сброшены сдвиги	мкс 1.6 мс	
0	0	0	0	0	1	S/C	R/L	*	*		Сдвиг курсора/дисплея	40 мкс	
0	0	0	1	A <sub>CG</sub>								Установка адреса <b>CGRAM</b>	40 мкс
0	0	1	A <sub>DD</sub>								Установка адреса <b>DDRAM</b>	40 мкс	
0	1	<b>BF</b>	AC								Чтение флага готовности <b>BF</b> и счетчика адреса <b>AC</b>	40 мкс	
1	0	Данные								Запись данных в <b>CGRAM</b> или <b>DDRAM</b>	40 мкс		
1	1	Данные								Чтение данных из <b>CGRAM</b> или <b>DDRAM</b>	40 мкс		

Программирование контроллера выполняется с помощью команд, представленных в табл. 1, а назначение флагов, используемых в командах, отражается в табл. 2.

Табл. 2. Флаги, управляющие работой контроллера **HD44780**.

Флаг	Назначение	Значение Флага	
		<b>0</b>	<b>1</b>
<b>I/D</b>	Режим смещения счетчика адреса <b>AC</b>	Уменьшение	Увеличение
<b>S</b>	Режим сдвига содержимого дисплея (*)	Сдвиг дисплея не производится	После записи в <b>DDRAM</b> очередного кода дисплей сдвигается в направлении, определяемым флагом <b>I/R</b> : 0 – вправо; 1 – влево

<b>S/C</b>	Производит вместе с флагом <b>R/L</b> операцию сдвига содержимого дисплея или курсора. Определяет объект смещения (*)	Сдвигается курсор	Сдвигается дисплей
<b>R/L</b>	Производит вместе с флагом <b>S/C</b> операцию сдвига дисплея или курсора. Уточняет направление сдвига	Влево	Вправо
<b>D/L</b>	Флаг, определяющий ширину шины данных	4 разряда	8 разрядов
<b>N</b>	Режим развертки изображения на ЖКИ	Одна строка	Две строки
<b>F</b>	Размер матрицы символов	5x8 символов	5x10 символов
<b>D</b>	Наличие изображения	Выключено	Включено
<b>C</b>	Курсор в виде подчеркивания	Выключен	Включен
<b>B</b>	Курсор в виде мерцающего знакоместа	Выключен	Включен

\*При сдвиге не производится изменение содержимого **DDRAM**, изменяются только внутренние указатели расположения видимого начала строки **DDRAM**.

После подачи питания контроллеру флаги принимают значения, указанные в табл. 3.

Так как на момент включения АЦМ ничего не отображает (флаг **D = 0**), то для того, чтобы вывести какой-либо текст необходимо как минимум, включить изображение, установив флаг **D = 1**.

Процесс инициализации контроллера АЦМ состоит из двух частей: сброс контроллера и задание режима работы.

При подаче питания необходимо выдержать паузу не менее 15 мс для установления рабочего питания (> 4.5 В).

Для сброса котроллера рекомендуется произвести следующую последовательность действий.

Выполнить команду, задающую разрядность шины данных и параметры развертки экрана (38h – для 8-ми разрядной шины, размером символов 5x8 и двумя строками; 30h – для 8-ми разрядной шины, размером символов 5x8 и одной строкой). Перед

выполнением этой команды нет необходимости проверять значение флага BF.

Табл. 3. Значения управляющих флагов после подачи питания.

Флаг	Значение	Режим
<b>I/D</b>	1	Режим увеличения счетчика на 1
<b>S</b>	0	Без сдвига изображения
<b>D/L</b>	1	8-ми разрядная шина данных
<b>N</b>	0	Режим развертки одной строки
<b>F</b>	0	Символы с матрицей 5x8 точек
<b>D</b>	0	Отображение выключено
<b>C</b>	0	Курсор в виде подчеркива выключен
<b>B</b>	0	Курсор в виде мерцающего знакоместа выключен

Выдержать паузу не менее 4.1 мс и повторить предыдущую команду без проверки флага BF.

Выдержать паузу 100 мкс и в третий раз повторить команду без проверки флага BF.

Эти действия призваны перевести контроллер в исходный режим работы из **любого** состояния.

Затем необходимо задать режим работы контроллера. Режим определяется теми задачами, для которых используется АЦМ.

Наиболее часто используют следующую последовательность команд: 38h, 0Ch, 06h, 01h. 38h устанавливает режим отображения 2-х строк с матрицей 5x8 точек и работу с 8-ми разрядной шиной данных, 0Ch включает отображение на экране АЦМ без отображения курсора, 06h устанавливает режим автоматического перемещения курсора слева направо после вывода каждого символа, 01h осуществляет очистку видеопамати АЦМ, обнуление счетчика адреса и его адресацию на **DDRAM**.

Эта последовательность может быть дополнена другими командами.

Например, можно выполнить вывод не изменяемого в процессе работы программы текста.

При отображении информации необходимо учитывать время, необходимое для выполнения команд, которое меняется в диапазоне от 40 мкс до 1,64 мс (при частоте генератора 270 КГц).

Поэтому после выполнения каждой команды необходимо вводить соответствующую задержку (смотри таблицу 1) или контролировать значение флага готовности BF, который сбрасывается в момент завершения команды (**BF=0**). Для проверки флага BF необходимо выполнить чтение **IR** (**RS=0, R/W=1**) и определить значение старшего разряда полученного байта.

Для вывода символа на дисплей необходимо задать значение адреса **DDRAM** или **CGRAM** и записать символ в регистр **DR**.

При записи символов в **DDRAM** необходимо учитывать, что значения адресов первой строки лежат в диапазоне (0 -39) или (0 -27h), второй – (64 -103) или (40h-67h), а диапазон 28h...3Fh (а также 68h...7Fh) являются неопределенными и результаты работы с ними могут быть непредсказуемыми.

Для АЦМ, работающего в двухстрочном режиме, при последовательной записи символов после заполнения одной строки курсор автоматически переходит на новую строку. Для принудительного перемещения курсора на начало второй строки ему следует присвоить значение 40h (64). **В четырехстрочном дисплее адреса первой и второй строки остаются прежними (0 и 40h), адрес третьей строки - 14h, а четвертой 54h.**

Так как буфер данных имеет больше ячеек, чем число знакомест дисплея, то, смещая окно индикатора относительно буфера данных, можно отображать на дисплее различные области буфера (рис. 2). У двухстрочных индикаторов первые 40 ячеек буфера данных, обычно, отображаются на верхней строке дисплея, а вторые 40 ячеек - на нижней строке. Сдвиг окна дисплея относительно буфера данных для верхней и нижней строк происходит синхронно. Курсор будет виден на индикаторе только в том случае, если он попал в зону видимости дисплея (и предварительно была подана команда отображения курсора).

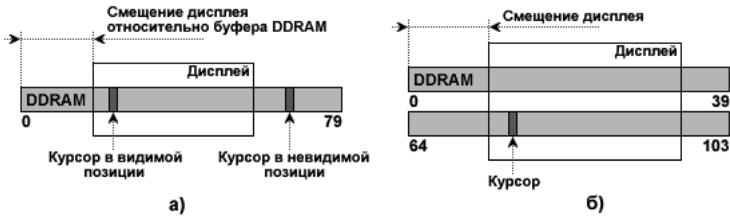


Рис. 2. Отображение на дисплее символов, записанных в **DDRAM** (а - однострочный ЖКИ, б - двухстрочный ЖКИ)

Чтение регистра **DR** приводит к загрузке содержимого **DDRAM** или **CGRAM**, в зависимости от текущего режима, при этом курсор смещается на одну позицию, как и при записи.

Чтение регистра **IR** позволяет определить текущее значение счетчика и флаг готовности. Значение счетчика **AC** размещается в семи младших разрядах (7 разрядов, если адресуется **DDRAM**, и 6 – если **CGRAM**), а старший бит содержит флаг занятости **BF**.

После совершения операции записи или чтения **DDRAM** и анализа **BF**, прочитанное в этом же цикле значение **AC** скорее всего не будет достоверным. Дело в том, что между появлением признака готовности и вычислением контроллером нового значения **AC** существует некоторый временной интервал, составляющий около 4 мкс при тактовой частоте контроллера 270 кГц. Поэтому, если необходимо получить истинное значение **AC**, нужно совершить повторную операцию прочтения **IR** спустя не менее чем 4 мкс. Если контроллер работает на частоте ниже 270 кГц, время ожидания необходимо пропорционально увеличить.

Контроллеры, устанавливаемые на АЦМ, могут иметь различные наборы символов. Это зависит как от производителя контроллера, так и от модификации данной конкретной модели.

Кодировка ПЗУ, содержащая английские и русские символы, представлена в табл. 4.

Для выбора соответствующего символа необходимо указать в команде его адрес, который формируется из номера столбца (старшая тетрада) и номера строки (младшая тетрада). При кодировании английского текста можно воспользоваться макрокомандой ассемблера **DB "TEXT"**.

Чтобы определить собственный символ, необходимо установить счетчик **АС** на адрес начала матрицы требуемого символа в **CGRAM** – 00h, 08h, 10h и т.д. (00h, 10h, 20h для режима 5x10 точек) – и произвести перезапись всех байтов матрицы, начиная с верхней строки.

Для кодирования матрицы используются 8 байт, пять младших битов которых несут информацию о рисунке (1 – сегмент включен, 0 – сегмент выключен), четвертый разряд каждого из 8-ми (или 11-ти в режиме 5x10) байтов матрицы определяет левую колонку символа, а 0-й - правую, старшие три бита не используются.

Матрица программируемых символов допускает использование полной высоты строки (8 строчек для режима 5x7 и 11 строчек для режима 5x10), но в области подчеркивающего курсора (нижняя строка) размещать информацию не рекомендуется.

Пример кодирования символа представлен на рис. 3.

Из допустимых для размещения в **DDRAM** кодов символы с кодами 00h...07h (и их дубликат с кодами 08h...0Fh) имеют специальное назначение – это переопределяемые символы, графическое изображение которых может назначить сам потребитель, разместив соответствующую информацию в области **CGRAM**. Для программирования доступны 8 переопределяемых символов в режиме с матрицей 5x7 точек и 4 с матрицей 5x10 (в режиме 5x10 переопределяемые символы адресуются кодами **DDRAM** через один 00h, 02h, 04h, 06h).

	■	■	■	■	<b>\$0F</b>
■				■	<b>\$11</b>
■				■	<b>\$11</b>
	■	■	■	■	<b>\$0F</b>
		■		■	<b>\$05</b>
	■			■	<b>\$09</b>
■				■	<b>\$11</b>
					<b>\$00</b>

Рис. 3. Пример кодирования символа.

Табл. 4. - Коды символов русифицированного индикатора.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	\	p			Б	Ю	ч	□	Д	¼
1			!	1	A	Q	a	q			Г	Я	ш	□	Ц	1 1/ 3
2			"	2	B	R	b	r			ё	б	ь	□	Щ	½
3			#	3	C	S	c	s			Ж	в	ы	□	д	¾
4			\$	4	D	T	d	t			З	г	ь	□	ф	□
5			%	5	E	U	e	u			И	ё	э	×	ц	□
6			&	6	F	V	f	v			Й	ж	ю	□	щ	□
7			'	7	G	W	g	w			Л	з	я	□	□	□
8			(	8	H	X	h	x			П	и	«	□ □	..	□
9			)	9	I	Y	i	y			У	й	»	↑	~	□
A			*	:	J	Z	j	z			Ф	к	“	↓	é	□
B			+	;	K	[	k	□			Ч	л	”	□	□	□
C			,	<	L	Ф	l	□			Ш	м	□	□	ij	□
D			-	=	M	]	m	□			Ь	н	¿	□	□	§
E			.	>	N	^	n	↵			Ы	п	f	□	□	¶
F			/	?	O	_	o	□			Э	т	£	·	о	■

□-символы отсутствуют в ASCII

### 3. Взаимодействие с микропроцессорной системой

При соединении АЦМ с МПС необходимо учитывать назначение выводов, особенности сигналов и их временные соотношения, используемые режимы работы. [ 2]

В качестве примера рассмотрим АЦМ фирмы Data International DV – 0802.

Описание выводов приведено в табл. 5.

Названия линий ввода-вывода являются стандартными, но могут быть и другие варианты расположения контактов у конкретной конструкции АЦМ.

Табл. 5. Описание выводов DV – 0802.

Номер вывода	Название вывода	Описание
1	<b>GND</b>	(-) Питание 0 V
2	<b>VCC</b>	(+) Питание +5 V
3	<b>V<sub>0</sub></b>	Напряжение смещения, управляющее контрастностью
4	<b>RS</b>	Вход. Высокий уровень – данные; низкий – команды
5	<b>R / <math>\bar{W}</math></b>	Вход. Высокий – чтение, низкий – запись.
6	<b>E</b>	Вход. Строб, сопровождающий сигналы на шине "команды/данные".
7	<b>DB0</b>	Шина "команды/данные".
8	<b>DB1</b>	
9	<b>DB2</b>	
10	<b>DB3</b>	
11	<b>DB4</b>	
12	<b>DB5</b>	
13	<b>DB6</b>	
14	<b>DB7</b>	
15	<b>LEDA</b>	Линии подсветки
16	<b>LEDB</b>	

Схема включения питания АЦМ показана на Рис. 4.

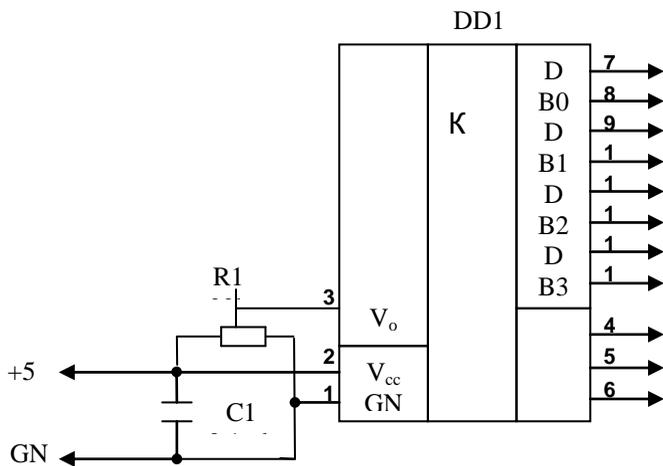


Рис. 4. Схема подключения питания АЦМ

Подстроечный резистор R1 позволяет плавно менять напряжение питания драйвера ЖКИ, что приводит к изменению угла поворота жидких кристаллов. Этим резистором можно регулировать контрастность.

Последовательность формирования основных сигналов представлена в табл. 5 -8.

В исходном состоянии сигнал  $E = 0$ , сигнал  $R/\overline{W} = 0$ , значение сигнала  $RS$  – произвольное (в зависимости от вида передаваемой информации - данные или команда),  $DB0 - DB7$  в состоянии высокого импеданса (**HI**). Такое состояние управляющих сигналов ( $E$  и  $R/\overline{W}$ ) должно поддерживаться все время в промежутках между операциями обмена с ЖКИ модулем. Шина данных в эти моменты в принципе свободна, и может использоваться в мультиплексном режиме для каких либо других целей, например, для сканирования матрицы клавиатуры.

Табл. 5. Операция записи для 8-ми разрядной шины.

---

Установить значение линии <b>RS</b> Линии $R/\bar{W} = 0$ Вывести значение байта данных на линии шины <b>DB0...DB7</b> Установить линию <b>E=1</b> Установить линию <b>E=0</b> Установить линии шины <b>DB0...DB7 = HI</b>
---

---

Табл. 6. Операция чтения для 8-ми разрядной шины.

---

Установить значение линии <b>RS</b> Установить линию $R/\bar{W} = 1$ Установить линию <b>E=1</b> Читать значение байта данных с линии шины <b>DB0...DB7</b> Установить линию <b>E=0</b> Установить линию $R/\bar{W} = 0$
---

---

Табл. 7. Операция записи для 4-х разрядной шины.

---

Установить значение линии <b>RS</b> Вывести значение старшей тетрады байта данных на линии шины <b>DB4...DB7</b> Установить линию <b>E=1</b> Установить линию <b>E=0</b> Вывести значение младшей тетрады байта данных на линии шины <b>DB4...DB7</b> Установить линию <b>E=1</b> Установить линию <b>E=0</b> Установить линии шины <b>DB0...DB7 = HI</b>
--

---

Табл. 8. Операция чтения для 4-х разрядной шины.

---

Установить значение линии <b>RS</b> Установить линию $R/\bar{W} = 1$ Установить линию <b>E=1</b> Читать значение старшей тетрады байта данных с линии шины <b>DB4...DB7</b> Установить линию <b>E=0</b> Установить линию <b>E=1</b> Читать значение младшей тетрады байта данных с линии ши- ны <b>DB4...DB7</b> Установить линию <b>E=0</b> Установить линию $R/\bar{W} = 0$
---

---

Приведенные в табл. 5 -8 операции подразумевают, что время выполнения каждого шага составляет не менее 250 нс. Поэтому необходимо тщательно контролировать минимальные значения временных интервалов, чтобы они всегда находились в области допустимых значений, указанных в табл. 9, и при необходимости вводить задержки.

Табл. 9. Значение временных характеристик при записи/чтении.

Параметр	Обозначение	Мин.	Макс.
Период сигнала E	$t_{\text{cycE}}$	500 нс	–
Положительный полупериод сигнала E	$PW_{\text{EH}}$	230 нс	–
Фронт/спад сигнала E	$t_{\text{Er}}, t_{\text{Ef}}$	–	20 нс
Установление адреса	$t_{\text{AS}}$	40 нс	–
Удержание адреса	$t_{\text{AH}}$	10 нс	–
Установление данных при записи	$t_{\text{DSW}}$	80 нс	–
Удержание данных при записи	$t_{\text{H}}$	10 нс	–
Установление данных при чтении	$t_{\text{DDR}}$	–	160 нс
Удержание данных при чтении	$t_{\text{DHR}}$	5 нс	–

Формирование сигналов может быть выполнено программно-аппаратными или программными средствами.

Программно-аппаратная реализация предполагает в ОМК наличие команд работы с внешней памятью, в процессе выполнения которых формируются сигналы

чтения/записи, используемые в качестве сигналов E или  $R/\bar{W}$ . Как правило, в этом случае используется восьмиразрядная шина данных. Оставшиеся сигналы формируются программно. К недостаткам такого способа следует отнести:

- ограниченное применение, так как не все ОМК допускают использование внешней шины данных,
- необходимость согласования временных диаграмм ОМК и АЦМ, что может потребовать применения внешних схем задержки,

– использование внешних схем приводит к дополнительным аппаратным затратам и усложнению печатной платы.

В качестве примера приведена схема подключения АЦМ к микроконтроллеру 8051 (рис.5 ), выполненная в виртуальной среде моделирования Proteus.

АЦМ подключен к порту P0, работающему в режиме микропроцессора. Обращение к регистрам HD-4780 выполняется как к ячейкам внешней памяти данных. Поэтому сигналы записи/чтения WR и RD формируются аппаратно и через схему NAND поступают на вход E. Управление линиями RS и R/W осуществляется младшими разрядами шины адреса, которые формируются с помощью регистра U2. Достоинством этой схемы является возможность подключения к порту P0 других устройств, а недостатки описаны выше.

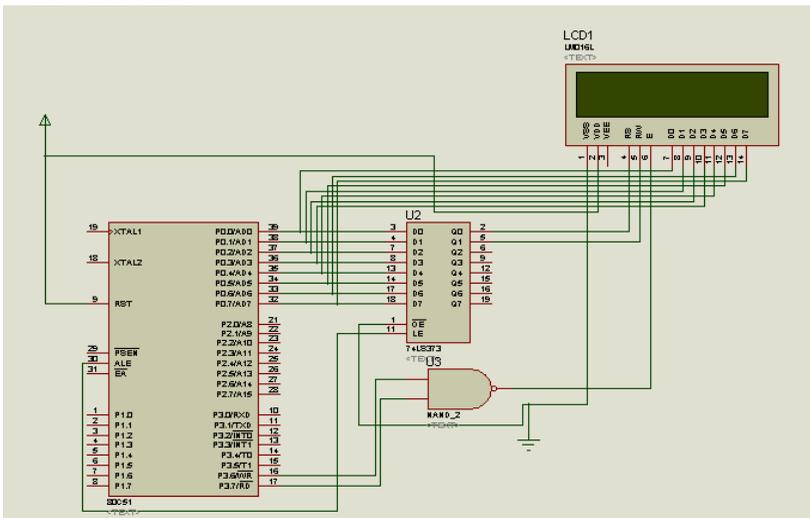


Рис. 5. Подключение АЦМ к микроконтроллеру 8051

Более универсальным является программный способ управления, при котором необходимое формирование сигналов и требуемые задержки формируются процессорным блоком ОМК (рис.8). Обмен информацией выполняется через свободные линии портов. При использовании четырехразрядной шины данных этот

способ приводит к уменьшению требуемых линий ввода-вывода (7 вместо 11).

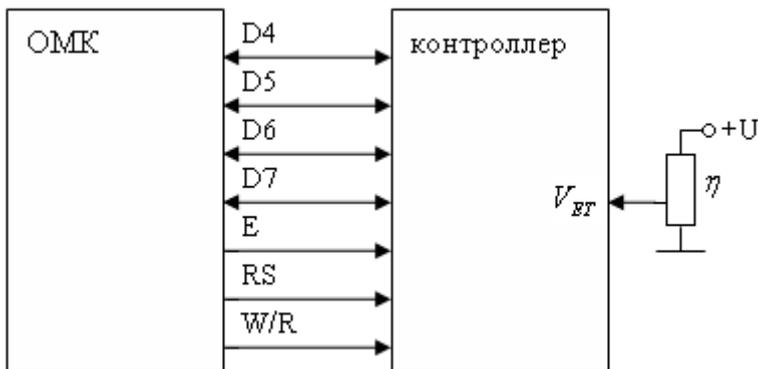


Рис. 6. Схема соединения АЦМ при программном управлении

Организация ввода-вывода с МПС возможна в синхронном или асинхронном режимах.

При синхронном вводе-выводе нет необходимости в анализе флага готовности, линия  $R/\bar{W}$  АЦМ должна быть заземлена, но цикл обмена должен быть рассчитан на самую медленную команду (1,64 мс). Фактически выполняется только запись информации в АЦМ и отсутствует обратная связь с модулем.

Наиболее часто используется асинхронный обмен, при котором очередная команда передается после проверки флага готовности **ВФ**.

В лабораторной работе используется оценочный модуль МСЕ-51 на основе ОМК АТ89С252, к разъему пользователя которого подключен АЦМ на базе контроллера HD-44780.

Управление модулем ЖКИ выполняется линиями А0, А1 шины адреса:

Запись команды	FFF0h ( RS=0, R/W=0)
Чтение команды	FFF2h ( RS=0, R/W=1)
Запись данных	FFF1h ( RS=1, R/W=0)
Чтение данных	FFF3 ( RS=1, R/W=1)

Управление по входу E обеспечивается сигналом STR. Формирование сигнала STR выполнено в соответствии с временными диаграммами (рис. 5, 6) средствами программируемой логической интегральной схемы ПЛИС.

#### 4. Программирование контроллера на ассемблере 8051

Для вывода информации на ЖКИ необходимо преобразовать выводимые данные в код ASCII в соответствии с таблицей 4.

Для отображения информации в десятичном виде двоичный код преобразуют в двоично-десятичный, а затем – в код ASCII.

Наиболее часто используются следующие алгоритмы преобразования в двоично-десятичный код:

- Последовательное деление двоичного числа на двоичные эквиваленты десятичных чисел  $10^n$ ,  $10^{n-1}$ ,  $10^{n-2}$ , ...,  $10$  где n- степень десятичного числа, и сохранения после каждого деления частного и остатка. Если команда деления в ОМК отсутствует, то деление заменяется вычитанием. Например, для преобразования 16-разрядного двоичного числа (65535) необходимо последовательное вычитание 10000(2710h), 1000(3E8h), 100(64h), 10(0Ah) и формирование целочисленного значения соответствующего разряда. Поразрядное вычитание прекращается в том случае, если разность будет меньше соответствующего вычитаемого, после этого значение разности восстанавливается и выполняется переход к следующему разряду. Для организации циклической программы двоичные эквиваленты десятичных чисел хранят в виде таблицы, размещаемой в памяти программ или данных. Время преобразования для кода FFFF с внутренней частотой ОМК 1МГц -482мкс, а объем программы – 82 байта

- Применение схемы Горнера и обработка на основании правил двоично-десятичной арифметики. Особенностью двоично-десятичной арифметики, выполняемой на двоичных АЛУ, является необходимость использования после каждой арифметической операции команды коррекции.

Двоичное число может быть представлено в виде полинома

$$N_B = b_{n-1} \cdot 2^{n-1} + b_{n-2} \cdot 2^{n-2} + \dots + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

Этот полином, записанный по схеме Горнера, имеет вид:

$$N_B = (\dots(b_{n-1} \cdot 2 + b_{n-2}) \cdot 2 + b_{n-3}) \cdot 2 + \dots + b_1) \cdot 2 + b_0$$

Старший разряд двоичного числа удваивается и суммируется со значением следующего разряда, полученная сумма опять удваивается и так далее. Удвоение реализуется путем сложения числа с самим собой. После каждого суммирования необходимо выполнять команду коррекции после сложения DA A, которая имеется, например, в ОМК MCS-51. В процессе выполнения программы преобразования двоичное число сдвигается влево, а выдвигаемый разряд помещается в бит С. В исходном состоянии сумматор обнуляется, а затем после каждого сдвига в нем накапливается результирующая сумма в соответствии с приведенным алгоритмом.

Сравнение этих алгоритмов для 8051 показывает, что второй алгоритм более оптимальный (при тех же условиях время – 361 мкс, а объем – 32 байта).

В других ОМК команда коррекции может отсутствовать (например, в ОМК Atmel). В этом случае необходимо разрабатывать подпрограмму коррекции или использовать первый алгоритм.

Преобразование двоично-десятичного числа в код ASCII выполняется поразрядной конкатенацией числа 3 с полученными значениями. Например, 359 должно иметь следующий вид 33h , 35h, 39h.

Вывод информации в 16-ричном коде проще выполнять табличным способом. Для этого байты разделяют на тетрады, значение которых используют в качестве смещения в таблице, содержащей ASCII коды цифр (0-F). Таблица может быть размещена в памяти программ или EEPROM. Например, для 8051

ORG 30h

Tabl:

30h,31h,32h,33h,34h,35h,36h,37h,38h,39h,41h,42h,43h,44h,45h,46h

Этот способ можно использовать и для преобразования десятичных чисел в семисегментный код.

Преобразование текстовой информации на английском языке в ASCII выполняется на основании таблицы 4 или средствами используемого симулятора. Например, в Pinnacle, VMLab, Proteus текст заключают в кавычки и используют макрокоманду dB:

Org 30h

Text: dB "Hello"

Вывод текста на русском языке выполняется только в соответствии с таблицей 4. Например, слово «Привет»

Org 30h

Text: dB 0a8h,70h,0b8h,0b3h,65h,0bfh

Затем таблица пересылается в ЖКИ для вывода на индикатор.

Разработка программы производится на основании модели, приведенной в разделе 2.

Программирование АЦМ выполняется в следующей последовательности:

сброс, настройка на требуемый режим работы (инициализация), выполнение процедуры вывода информации. Наиболее часто используется асинхронный режим работы, для реализации которого требуются подпрограммы: запись/чтение команды, запись/чтение данных, программируемая задержка, проверка бита готовности. Чтение команды выполняется в подпрограмме проверки бита готовности.

Для АЦМ, используемого в лабораторной работе, эти подпрограммы имеют вид:

Запись команды

ComWR: push ACC

ComWait: acall Status

jb ACC.7, ComWait

pop ACC

```
mov p2, #0FFh
mov r0, #0F0h
movx @r0, A
ret
```

Проверка бита готовности ( чтение команды)

```
Status:      mov p2, #0FFh
             mov r0, #0F2h
             movx A, @r0
             ret
```

Запись данных

```
DataWR:      push ACC
DataWait:    acall Status
             jb ACC.7, DataWait
             pop ACC
             mov p2, #0FFh
             mov r0, #0F1h
             movx @r0, A
             ret
```

Чтение данных

```
DataRD:      mov p2, #0FFh
             mov r0, #0F3h
             movx A, @r0
             ret
```

задержка

```
Wait:        mov A, #125
Again1:      push ACC
             clr A
Again0:      dec A
             jnz Again0
             pop ACC
             dec A
             jnz Again1
             ret
```

**Входные параметры подпрограмм записываются в аккумулятор.**

Подпрограмма задержки рассчитана на время около 50 мс. Это сделано в целях унификации, чтобы использовать одну подпрограмму как для инициализации контроллера, так и реализации «бегущей строки».

При использовании подпрограмм значение указателя стека необходимо задавать в свободной зоне резидентной памяти данных.

Обращение к модулю ЖКИ выполняется как к ячейке внешней памяти данных.

**В подпрограммах рекомендуется использовать команды MOVX @R0,A и MOVX A,@R0, в которых старший байт адреса формируется на выходе P2.** Это упрощает работу с текстом, расположенным в памяти программ, так как в этом случае необходимо использовать в качестве указателя адреса DPTR.

Реализация задания для модуля, используемого в лабораторной работе, включает следующие процедуры

Сброс. Передача кода 38h с задержкой не менее(4-5) мс (2-3) раза

Инициализация. Задать конфигурацию -38h, включить отображение, выключить курсор 0Ch, задать режим экрана(автоматическое перемещение слева направо, экран неподвижен) – 06h, очистить дисплей видеопамати -01h.

Преобразование выводимые данные в ASCII-код.

Вывод символ на требуемое знакоместо. Для первой строки команда вывода имеет вид (80h+n), а для второй - (C0+n), где n(0-7)-номер знакоместа. Например, для вывода “L” в первую строку и первое знакоместо

Mov a,#80h

Acall ComWR

Mov a,#”L”

Acall DataWR

При выводе «бегущей» строки необходимо записать текст в память данных или команд ОМК, загрузить его в видеопамать модуля и командами 18h (сдвиг строки влево), 1Ch (сдвиг строки вправо) реализовать сдвиг текста на экране . При выводе необходимо подобрать задержку для комфортного чтения текста.

Для вывода символов, не содержащихся в ПЗУ знакогенератора, выполнить следующее:

Создать таблицу отображаемого символа и разместить её в памяти команд ОМК. Таблица должна начинаться с верхней строки символа.

Задать адрес размещения сформированного символа в ОЗУ знакогенератора. Для режима 5x7- 00, 08, 10h, 18h и так далее.

Задать соответствующее значение счетчика АС- 40h, 48h, 50h, 58h и так далее

С выбранного адреса переписать таблицу в ОЗУ знакогенератора, начиная с верхней строки.

Записать созданный символ на требуемое знакоместо командой ComWR, указав начальный адрес строки и номер знакоместа (начальный адрес первой строки – 80h, второй – C0h).

Вывести символ на экран командой DataWR. Параметром является адрес ASCII –таблицы. Для символов 5x7 первый имеет адрес 00, второй – 01, третий 02 и т.д. Для символов 5x10 адреса соответственно 00, 02, 03 и т.д.

Пример формирования символа в виде движущегося самолета для двухстрочного 16-разрядного ЖКИ приведен ниже:

```
Ljmp Start
  org 30h
simvol: db 04h,12h,1fh,02h,04h,00,00,00; самолет
DataWR:  push ACC
DataWait: acall Status
          jb ACC.7, DataWait
          pop ACC
          mov p2, #0FFh
          mov r0, #0F1h
          movx @r0, A
          ret
ComWR:   push ACC
ComWait: acall Status
          jb ACC.7, ComWait
          pop ACC
          mov p2, #0FFh
```

```

        mov r0, #0F0h
        movx @r0, A
        ret
DataRD:    mov p2, #0FFh
        mov r0, #0F3h
        movx A, @r0
        ret
Status:    mov p2, #0FFh
        mov r0, #0F2h
        movx A, @r0
        ret
Wait:      mov A, #125
Again1:    push ACC
           clr A
Again0:    dec A
           jnz Again0
           pop ACC
           dec A
           jnz Again1
           ret
InitializeLCD:
mov p2, #0FFh
   mov r0, #0F0h
   mov A, #38h
   movx @r0, A
   acall Wait
   movx @r0, A
   acall Wait
   movx @r0, A
   mov A, #38h
   acall ComWR
   mov A, #0Ch
   acall ComWR
   mov A, #06h
   acall ComWR
   mov A, #01h
   acall ComWR

```

ret

Start:  
mov sp,#70h  
mov r6,#2;Счетчик проходов  
k10: acall InitializeLCD  
mov a,#40H;Установка начального адреса CGRAM(ОЗУ  
знакогенератора)  
acall ComWR  
acall Wait  
mov r2,#8 ; Количество элементов символа  
mov dptr,#simvol; Начальный адрес таблицы символа  
k3: clr a; Формирование символа в ОЗУ знакогенера-  
тора  
movc a,@a+dptr  
acall DataWR; Команда вывода данных и проверка го-  
товности  
acall Wait  
inc dptr  
djnz r2,k3  
djnz r6,k11; При первом проходе вывод в первую строку.  
mov a,#0c0H; Команда вывода во 2строку 1знакоместо  
acall ComWR  
acall Wait  
sjmp k12  
k11: mov a,#80H; Команда вывода в 1строку  
1знакоместо  
acall ComWR  
acall Wait  
k12: mov a,#00H; Адрес сформированного символа в  
таблице ASCII-кодов  
acall DataWR  
acall Wait  
mov r3,#16; Количество сдвигов  
k7: mov a,#1ch; Команда сдвига экрана вправо

```

acall ComWR
mov r5,#7h;Формирование задержки для регулировки
скорости сдвига
k9:  acall Wait;
    djnz r5,k9
    djnz r3,k7
    mov a,r6
    jz Stop;Проверка окончания программы
    ljmp k10
Stop:
    sjmp Stop

```

Принцип программирования, изложенные в методических указаниях, могут быть полезны при работе АЦМ и с другими типами ОМК [3].

## 5. Порядок выполнения лабораторных работ

1. Получить задание у преподавателя.
2. Написать программу на ассемблере и предварительно отладить ее на симуляторе.
3. При выключенном компьютере присоединить макет АЦМ к оценочному модулю, а модуль соединить с USB -портом компьютера.
4. Включить компьютер.
5. Загрузить hex-файл разработанной программы АЦМ в оценочный модуль MCE-51.
6. Запустить программу и приступить к её отладке.
7. Если программа не выполняется, загрузить в MCE-51 тестовую программу LCD5.hex. На индикатор должно быть выведено сообщение “Hello, student!”.
8. Вывод сообщения подтверждает работоспособность АЦМ и тогда следует проверить правильность разработанной программы.
9. Если сообщение не выводится, обратится к преподавателю.
10. При выполнении лабораторной работы **в виртуальной среде моделирования Proteus**, разработать и нарисовать схему, аналогичную рис.7, в соответствии с заданием, написать программу и отладить её.

11. После сдачи лабораторной работы преподавателю оформить отчет, который должен содержать задание, схему соединения АЦМ с оценочным модулем или микроконтроллером, программу на ассемблере с комментариями.

## **6. Контрольные вопросы**

1. В чем отличие полупроводниковых индикаторов от жидкокристаллических?
2. В чем преимущества и недостатки жидкокристаллических индикаторов (ЖКИ).
3. Какие параметры необходимо контролировать при выборе индикаторов?
4. Какие функции реализует алфавитно-цифровой модуль АЦМ на основе контроллера HD-44780?
5. Как подключить АЦМ к ОМК? Привести примеры различных способов подключения.
6. В чем преимущества и недостатки подключения АЦМ к шине данных? к портам ввода-вывода?
7. В каких случаях целесообразно использовать синхронный обмен с АЦМ? асинхронный? Как это реализовать?
8. Какие возможности предоставляет пользователю АЦМ для вывода текстовой? символьной информации?
9. Какие возможности предоставляет АЦМ для формирования произвольных символов? Как их сформировать?
10. Как вывести текстовую информацию на русском языке? на английском?
11. Как организовать просмотр содержимого DDRAM?
12. Как выполнить вывод информации с требуемого знакоместа?
13. Как организовать «бегущую строку»?

## 7. Список использованных источников

1. Жидкокристаллические индикаторы фирмы DATA International Библиотека электронных компонентов. Выпуск 8: [Текст] – М.: ДОДЕКА, 1999, 64 с.:ил.
2. Алфавитно-цифровые индицирующие ЖК-модули на основе контроллера HD44780 [Электронный ресурс]. [www.ccc-mc.ru](http://www.ccc-mc.ru)
3. Кривченко Т. Сопряжение AVR-микроконтроллеров и ЖКИ. [Электронный ресурс]. [www.atmel.ru](http://www.atmel.ru)

Методические материалы

**ОТОБРАЖЕНИЕ ИНФОРМАЦИИ  
НА ЖИДКОКРИСТАЛЛИЧЕСКИХ  
ИНДИКАТОРАХ С КОНТРОЛЛЕРОМ HD-44780**

*Методические указания*

*Составитель Иоффе Владислав Германович*

Самарский государственный  
аэрокосмический университет.  
443086 Самара, Московское шоссе, 34.

---