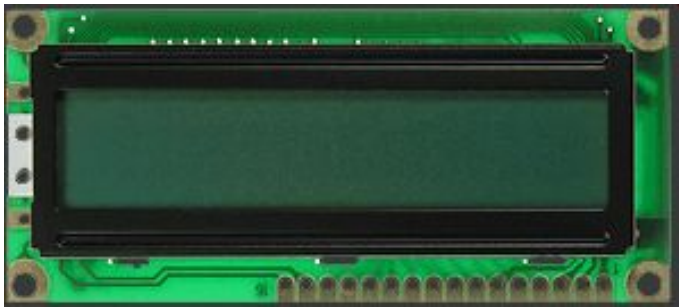


Жидкокристаллический индикатор (ЖКИ-LCD)



Структура ЖКИ



Внешнее устройство (контроллер) соединяется с ЖКИ через:

- параллельную синхронную мультиплексированную шину команд/данных,
- линию выбора регистра (**RS**),
- линию выбора операций чтения/записи (**R/W**),
- линию стробирования и синхронизации (**E**).

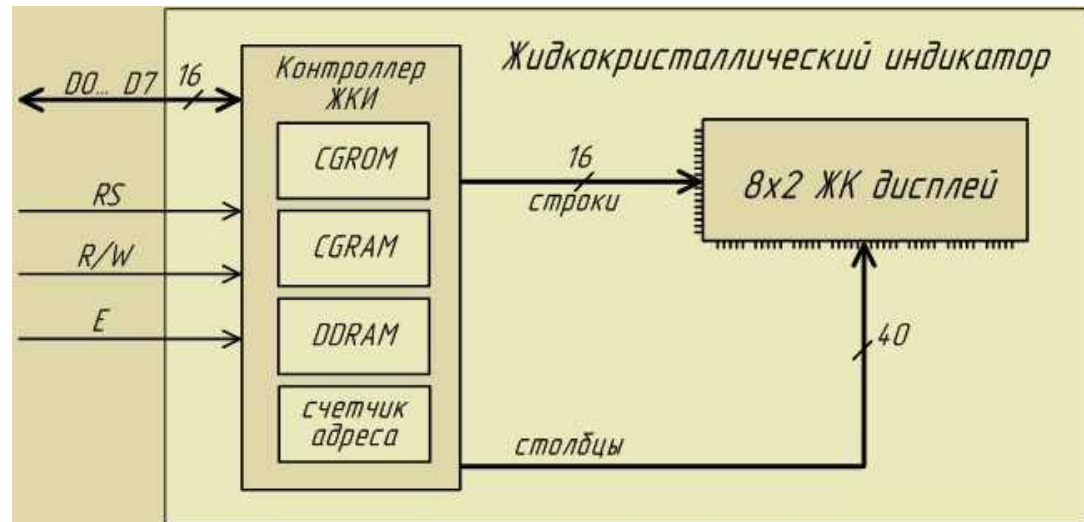


Рисунок 1 – Структура ЖКИ

Название вывода	Описание
VSS	Питание 0 V (-)
VDD	Питание +5 V (+)
V0	Напряжение смещения, управляющее контрастностью
RS	Вход. Высокий уровень — данные; низкий — команды
R/W	Вход. Высокий уровень — чтение; низкий — запись
E	Вход. Строб, сопровождающий сигналы на шине «Команды/Данные». LCD-модули с размером матрицы 4×40 символов имеют два вывода — E1 и E2, каждый для своей половины дисплея (для своего контроллера)
DB0...DB7	Шина «Команды/данные»

Подача питания на ЖКИ (стандартный диапазон температур)

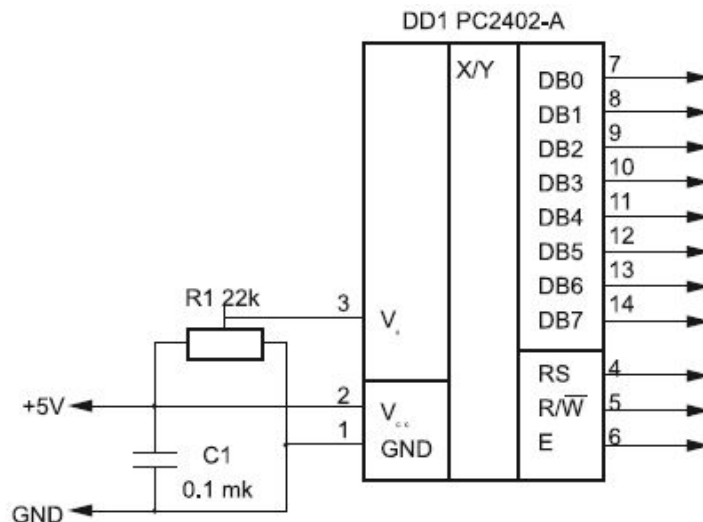


Рисунок 2 – Рекомендуемая схема питания ЖКИ

Подстроечный резистор R1 плавно меняет напряжение питания ЖКИ, что позволяет менять угол поворота жидких кристаллов и как следствие выставлять необходимую **контрастность индикатора** при необходимом угле обзора.

При изменении напряжения на выводе **Vo** сегменты плавно меняют свой цвет от прозрачного к непрозрачному, что **говорит о правильном подключении модуля питания.**

ЖКИ правильно отрегулирован, когда изображение сегментов едва проступает на основном фоне.

После подачи напряжения на модуль и прохождения внутренней инициализации, устройство включается в режиме развертки верхней строки.

Подача питания на ЖКИ (расширенный диапазон температур)

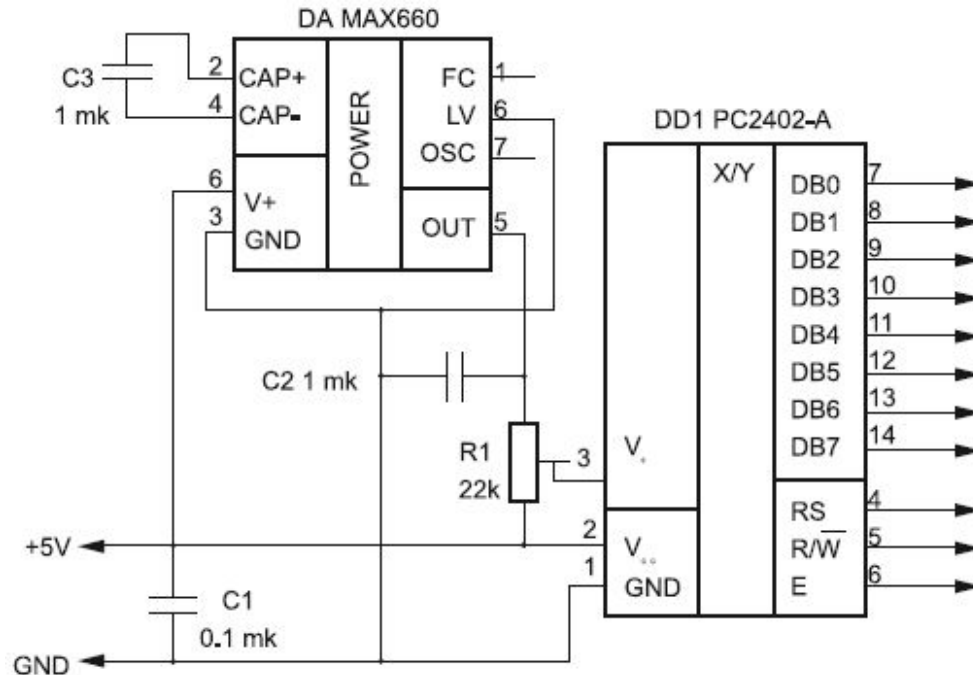


Рисунок 3 – Организация расширенного диапазона питания ЖКИ

Для применения **ЖКИ с расширенным температурным диапазоном**, **необходимо повышенное напряжение питания.**

Для этого необходимо увеличить напряжение питания. Эта задача решается подачей отрицательного напряжения (**максимум -5 V**) на вывод **V_o**. Источник отрицательного напряжения можно реализовать, например, на микросхеме MAX660, как показано на рисунке 3.

Подключение ЖКИ к микроконтроллеру

Управление контроллером ЖКИ может производиться по 4-х или 8-и разрядной шине. При этом обмен можно организовать либо с системной шиной, либо через порты ввода-вывода программными средствами.

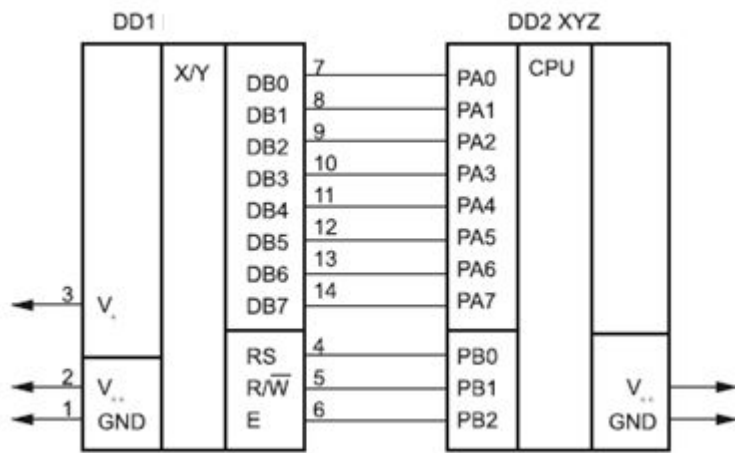


Рисунок 4 – 8-и проводное соединение

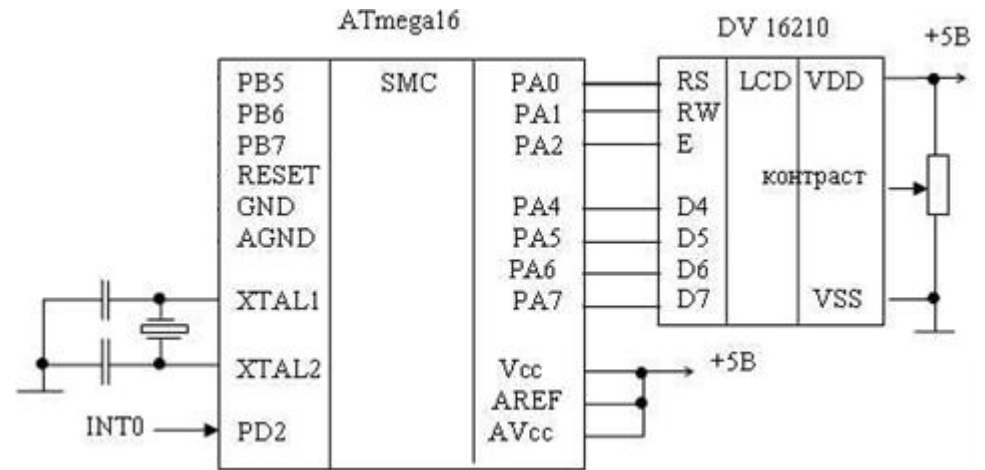


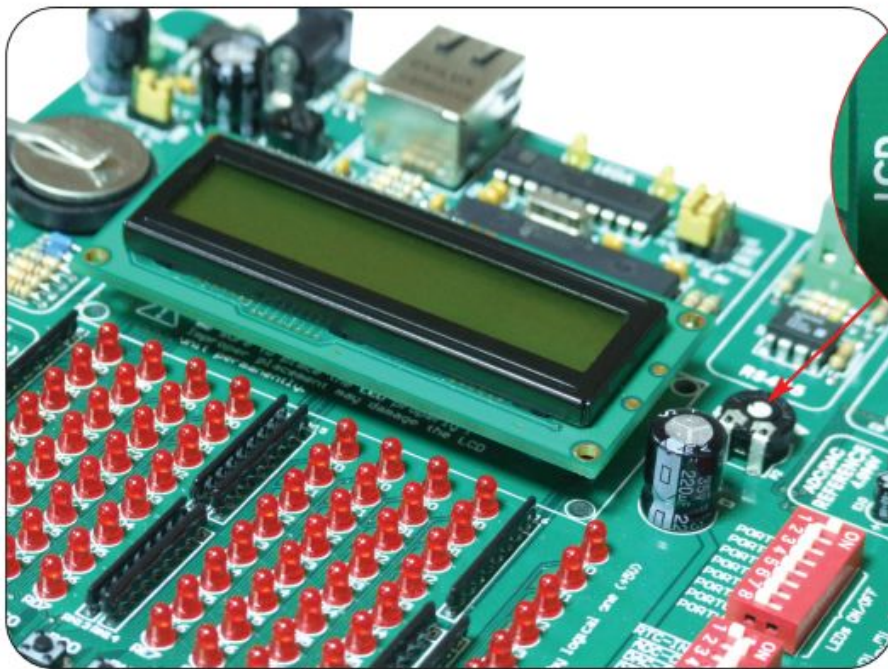
Рисунок 5 – 4-х проводное соединение

На рисунке 4 приведена схема включения контроллера с **8-и разрядной** условной микро-ЭВМ:

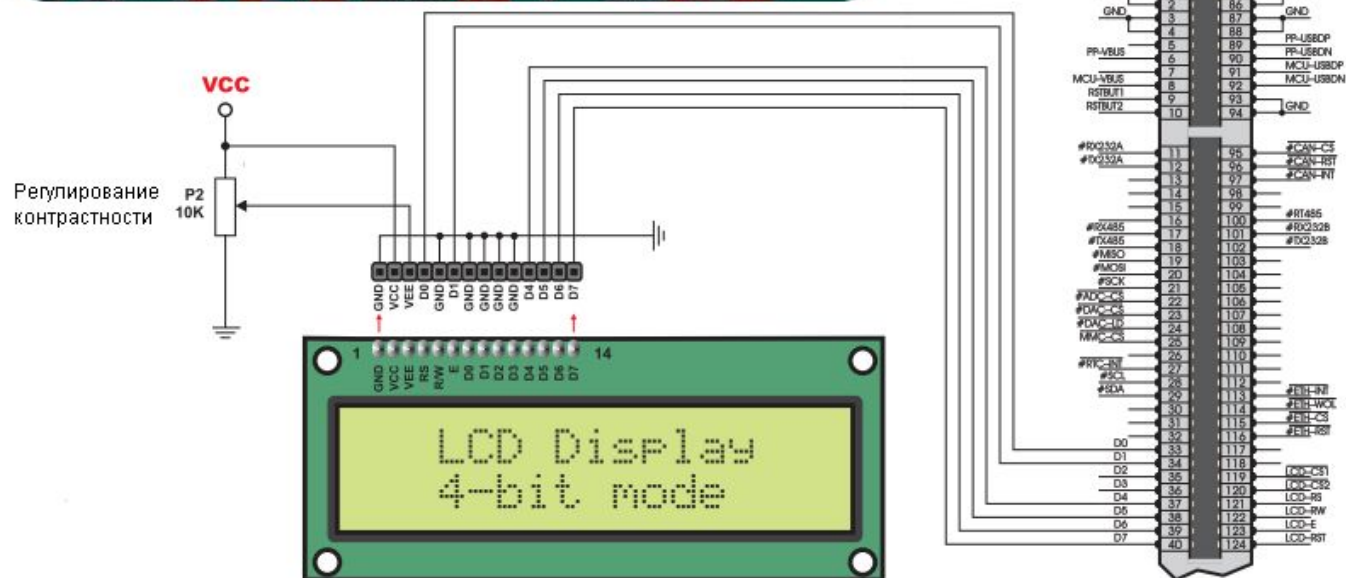
- 8-разрядный двунаправленный порт PA0...PA7 подключен к шине **DB0...DB7** ЖКИ модуля.
- К трехразрядному порту PB0...PB2 подключены управляющие сигналы **E, RS, R/W**.

На рисунке 5 контроллер подключен к микроконтроллеру ATmega16 в **4-разрядном** режиме, при этом используются старшие адреса шины данных **DB4...DB7**.

Схема подключения ЖКИ на лабораторном стенде



ЖКИ 2x16 в 4-х битном режиме

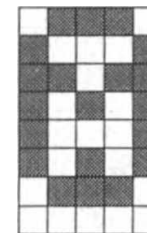


Контроллер HD44780 производства Hitachi для управления ЖКИ

Контроллер **HD44780** производства Hitachi. Это наиболее распространенный контроллер для управления знакосинтезирующем алфавитно-цифровым модулем. Практически все ведущие производители ЖКИ – Epson, Sanyo, Toshiba, Samsung, Philips выпускают аналоги этого контроллера или совместимые с ним по интерфейсу и командному языку микросхемы либо ЖКИ на базе этих контроллеров.

Можно говорить, что **HD44780 это промышленный стандарт**. Модули с этим контроллером применяются в самых разнообразных устройствах: измерительные приборы, промышленные, технологическое и медицинское оборудование, офисная техника.

Контроллер поддерживает размер символа **5x7 точек** и **5x10 точек**.



HD44780 может управлять двумя строками по 40 символов (для управления четырехстрочного модуля по 40 символов используют соответственно два однотипных контроллера).

При существующих стандартах **ЖКИ контроллер не накладывает ограничений на количество и комбинации отображаемых символов** (их количество может быть от 1 до 80).

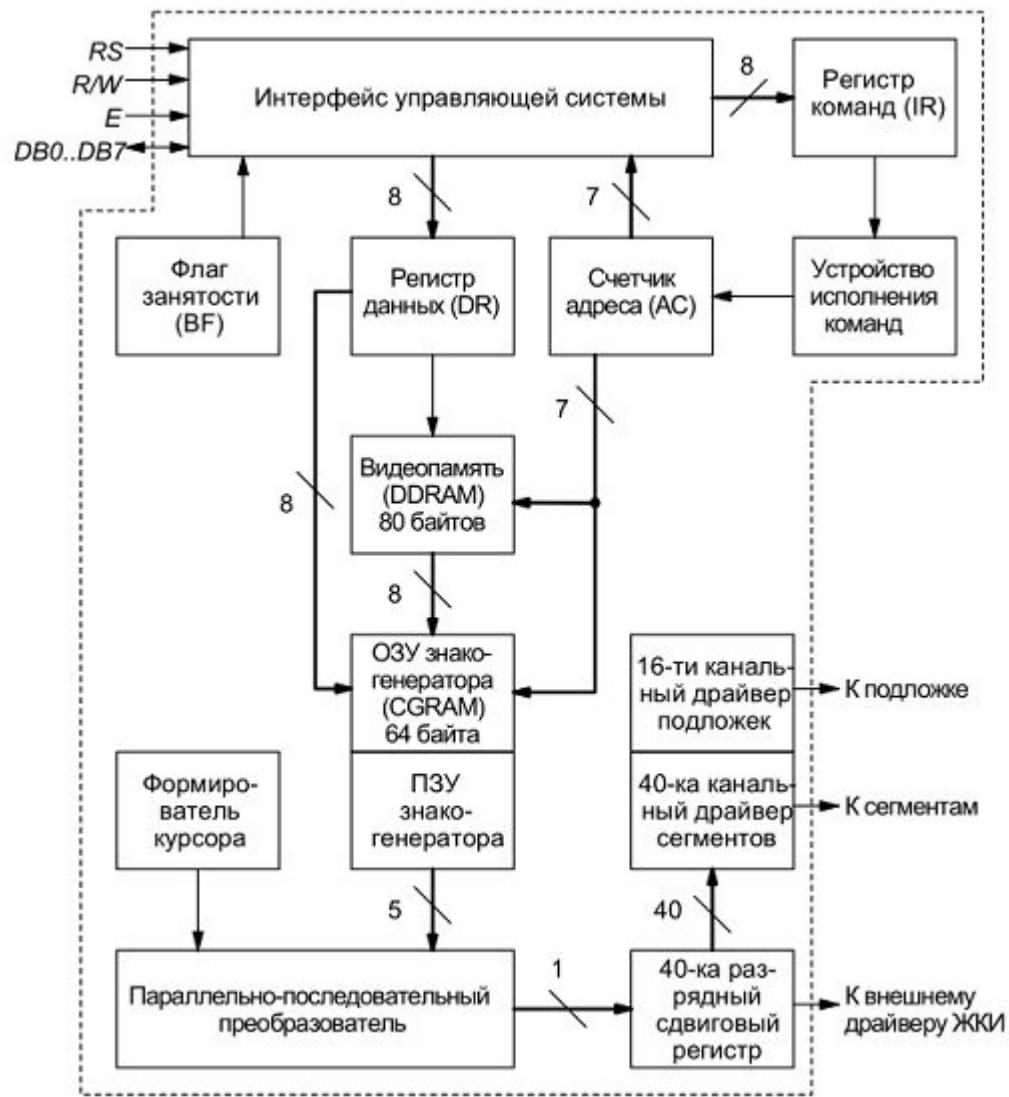


Рисунок 6 – Упрощенная структурная схема контроллера HD44780

DR (*Data register*) – регистр данных, **IR** (*Instruction register*) – регистр команд, **DDRAM** – видеопамять (80 байт), **CGRAM** – ОЗУ знакогенератора (16 байт), **AC** – счетчик адреса памяти, **BF** – флаг занятости контроллера.

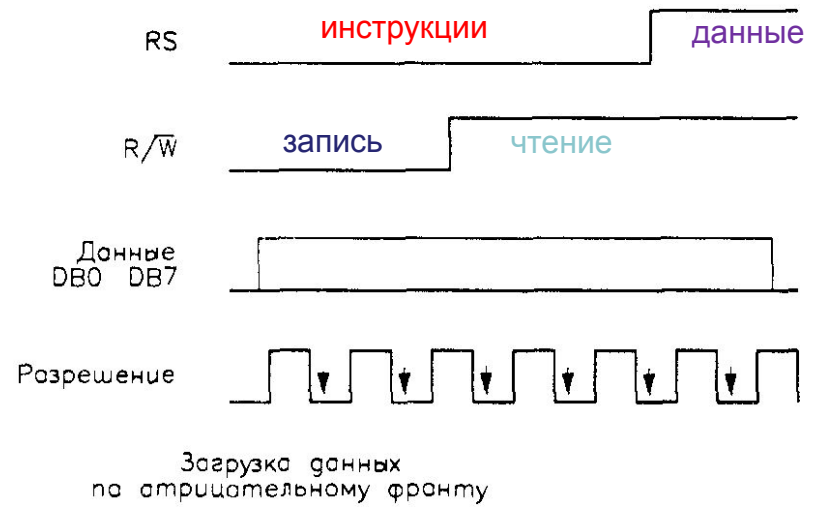
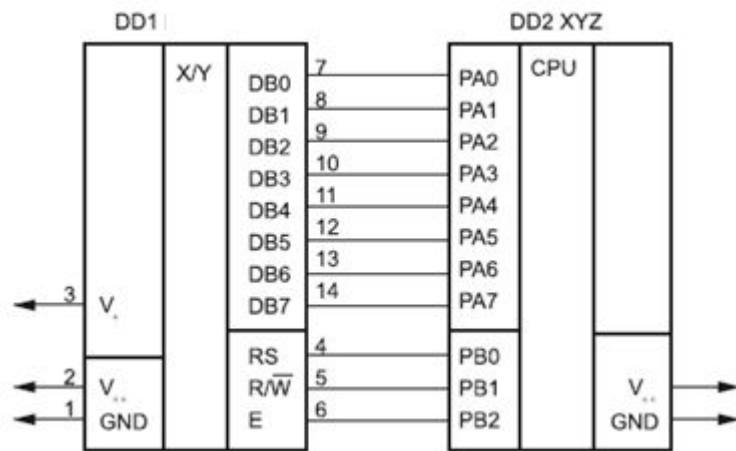


Рисунок 7 – Временная диаграмма управляющих сигналов – стандарт Motorola-M68

Линия **RS** отображает то, что загружается в модуль дисплея в данный момент времени – **инструкции** или **данные**:

- RS = 0** – передаются **инструкции**;
- RS = 1** – передаются **данные**.

Линия **R/W** показывает, какая операция проводится в данный момент времени – чтение или запись:

- R/W = 1** – **чтение**;
- R/W = 0** – **запись**.

Линия **E** (*enable* – разрешение) – управляет **чтением** и **записью** данных.

При загрузке команд/данных все необходимые сигналы подаются на входы RS, R/W, DB0÷DB7, а затем **по отрицательному фронту** на входе **ENABLE** (рисунок 7) команды/данные загружаются в дисплей.

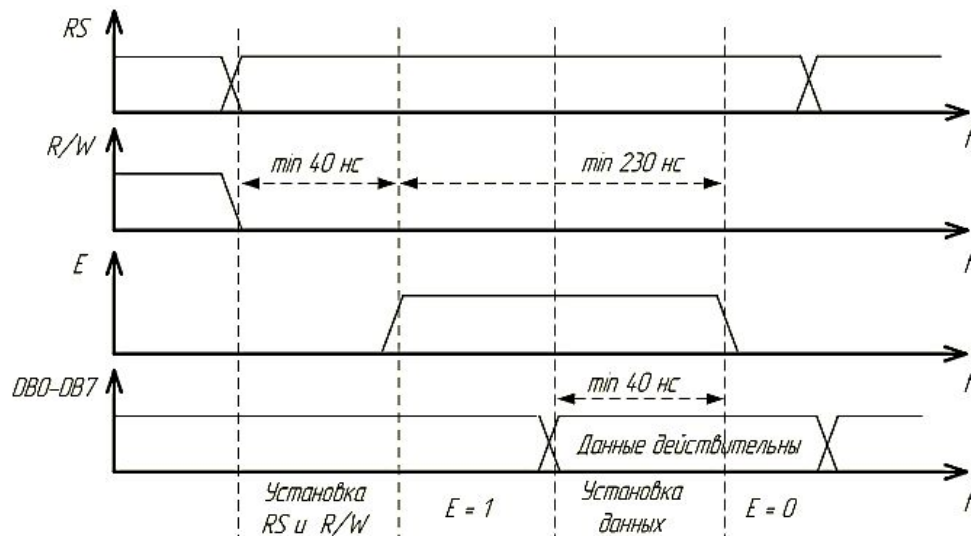


Рисунок 8 – Диаграмма записи информации контроллеру ЖКИ

На временной диаграмме на рисунке 8 показаны состояния управляющих сигналов и шины данных во время операций чтения и записи.

После подачи питания, ЖКИ находится в исходном состоянии, при этом состояние выводов принимает следующие значения:

- **E = 0,**
- **R/W = 0,**
- значение сигнала **RS произвольное,**
- шина данных **DB0...DB7 в состоянии высокого импеданса.**

В промежутках между операциями обмена сигналы **E** и **R/W** должны быть равны 0, в этот момент шина данных свободна и может использоваться в мультиплексном режиме для других целей.

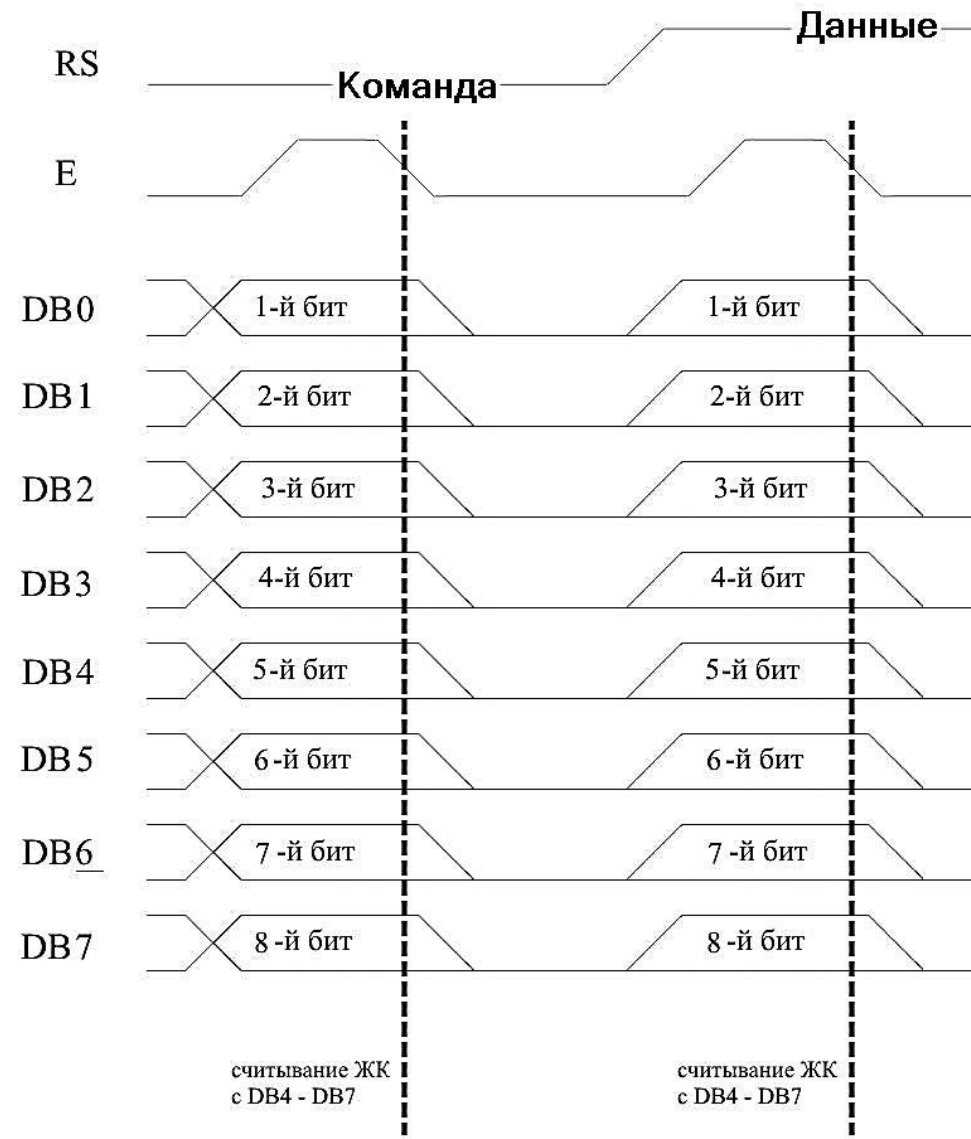
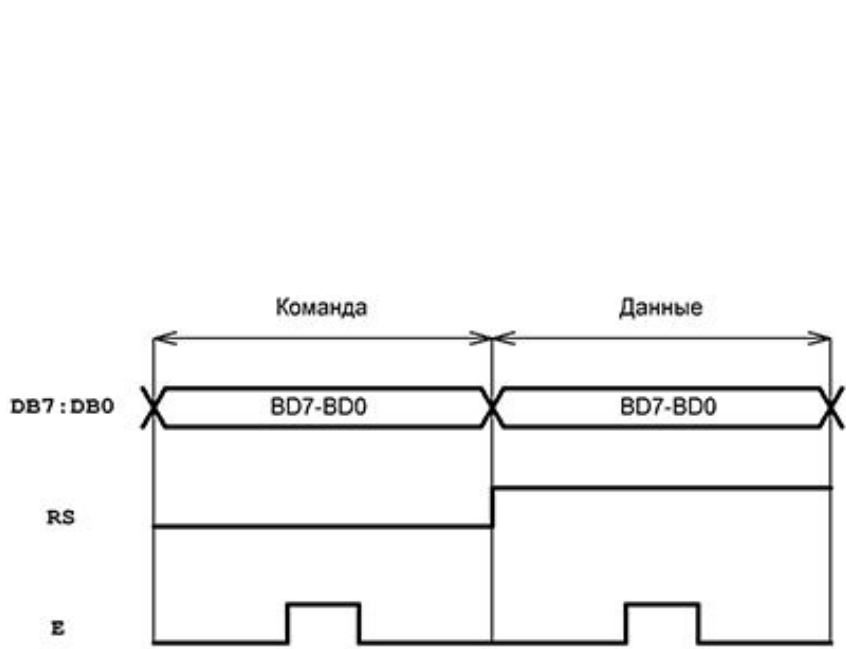


Рисунок – 9а. Последовательность передачи данных в HD44780 по 8-разрядной шине команд/данных

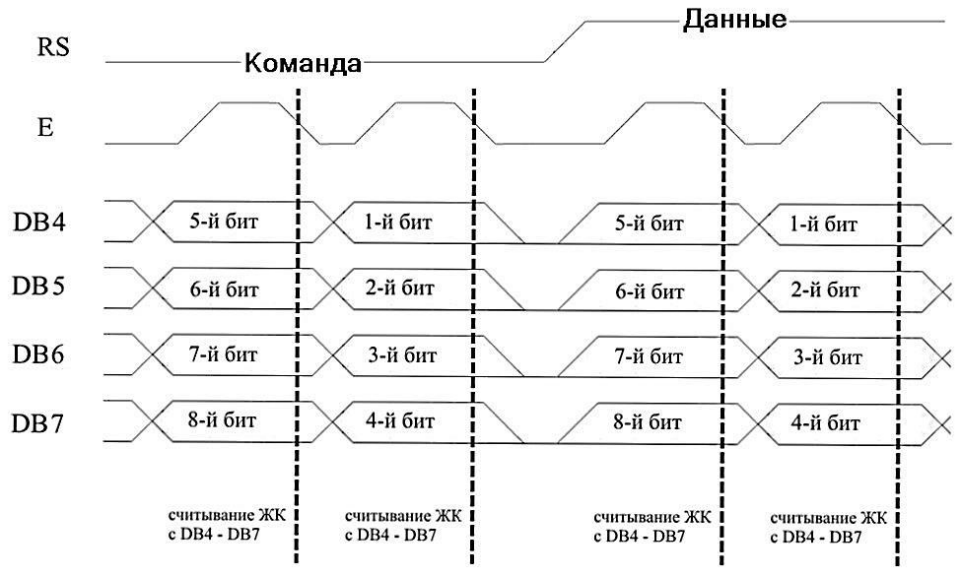
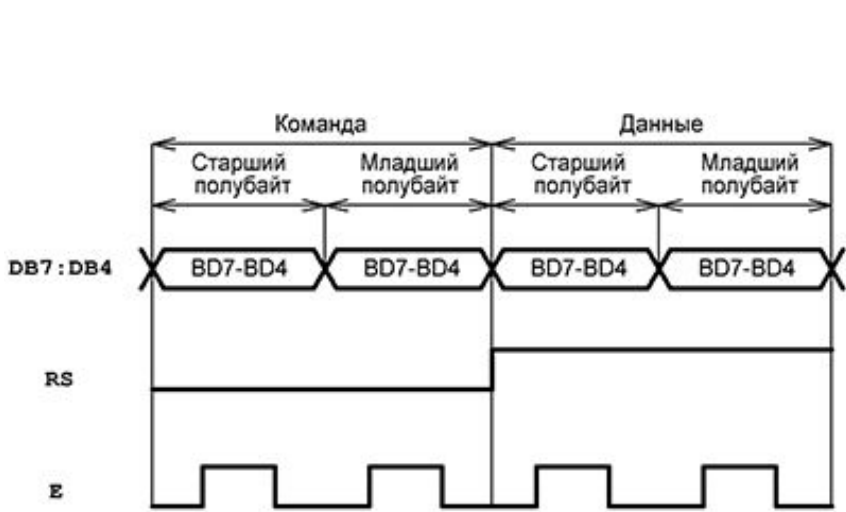


Рисунок – 96. Последовательность передачи данных в HD44780 по 4-разрядной шине команд/данных

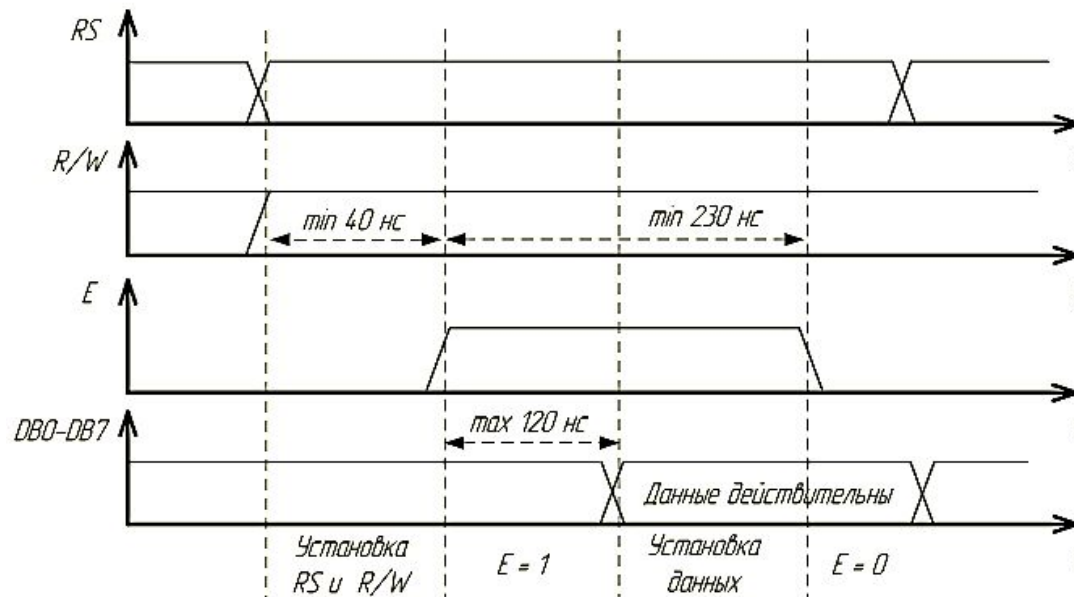


Рисунок 10 – Диаграмма чтения информации из контроллера ЖКИ

Время выполнения каждого шага не менее 250 нс

После приема информации контроллеру ЖКИ требуется некоторое время на выполнение команд, в это время управляющий контроллер не должен давать следующую команду или пересылать данные.

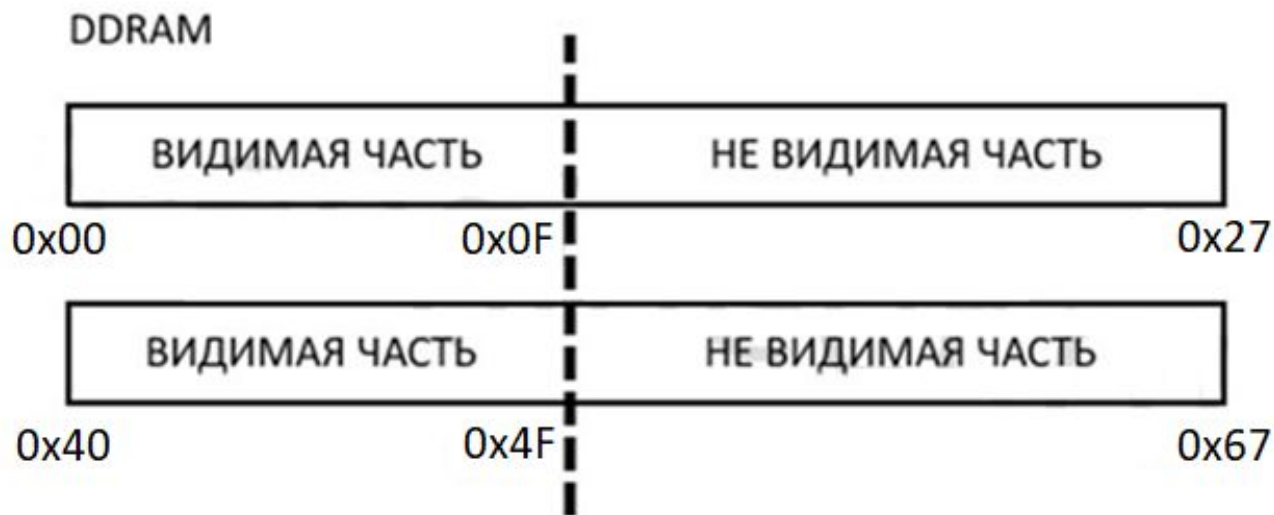
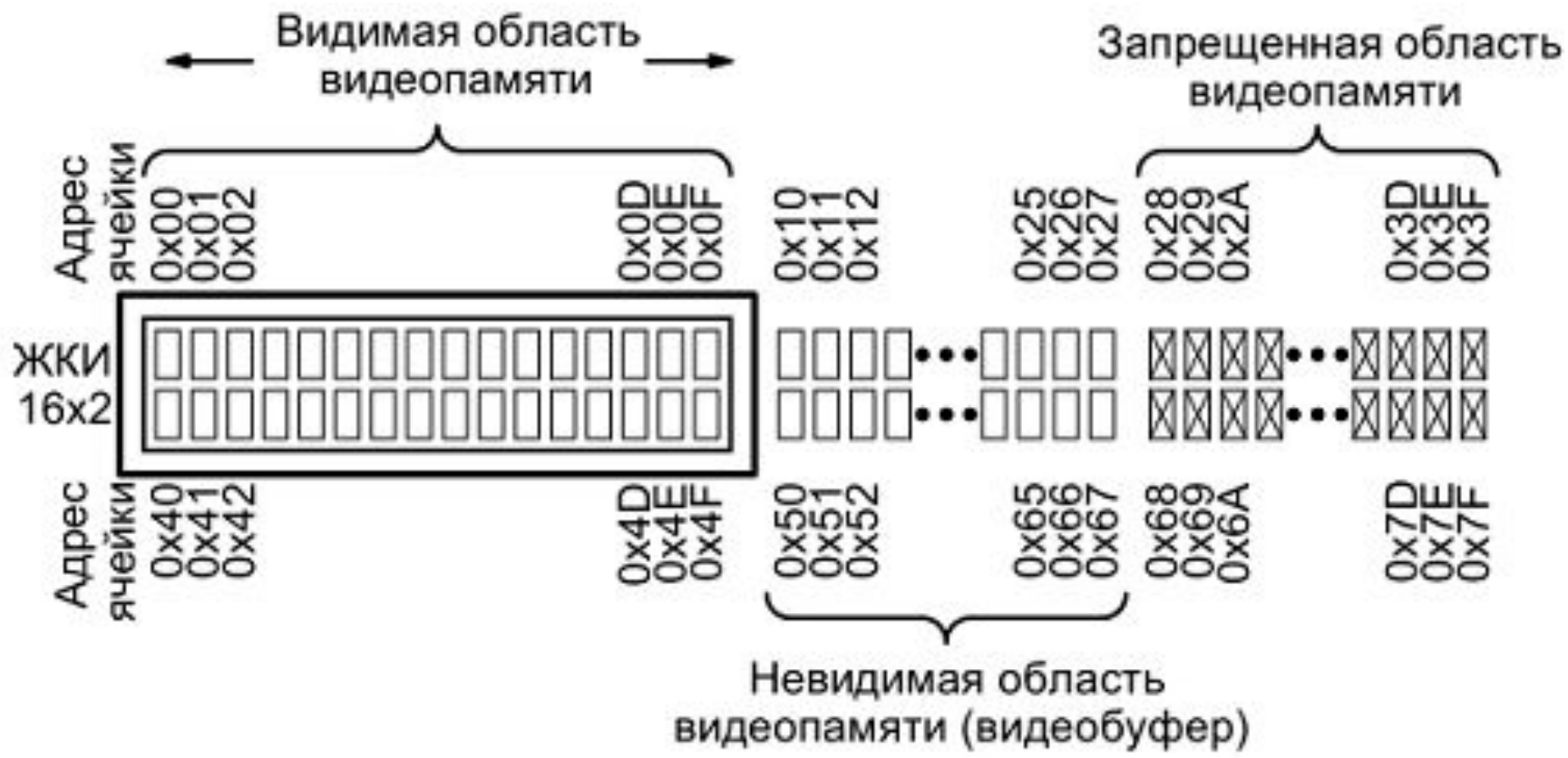


Рисунок 11 – Отображение символов из видеобуфера

В состав контроллера ЖКИ входят три вида памяти: CGROM, CGRAM, DDRAM.

Когда микроконтроллер передает в контроллер ЖКИ ASCII-коды символов, то они записываются в **DDRAM** (Display Data RAM – ОЗУ ASCII-кодов отображаемых символов), такую память называют **видеопамятью** или **видеобуфером**. Объем DDRAM зависит от числа строк и позиций на экране. Videобуфер в символьных индикаторах обычно содержит 80 ячеек памяти – обычно больше, чем отображаемое число знакомест дисплея.

У двухстрочных индикаторов ячейки с адресами от **0x00** и до **0x27** отображаются на верхней строке дисплея, а ячейки с адресами **0x40** ... **0x67** – на нижней строке. Смещая видимое окно дисплея относительно начального адреса DDRAM, можно отображать на дисплее различные области видеопамяти. Сдвиг окна отображения для верхней и нижней строк происходит синхронно, как это показано на рисунке 11. Курсор будет виден на индикаторе только в том случае, если он попал в зону видимости дисплея (и если предварительно была подана команда отображать курсор).



Старшая часть байта (D4 – D7)

	0h	1h	2h	3h	4h	5h	6h	7h	8h	9h	Ah	Bh	Ch	Dh	Eh	Fh	
0h				0	1	2	3	4	5	6	7	8	9	.	,	;	:
1h			!	"	#	\$	%	&	'	()	*	+	,	-	.	/
2h			"	2	3	4	5	6	7	8	9	.	,	;	:	;	:
3h			#	3	4	5	6	7	8	9	.	,	;	:	;	:	:
4h			#	4	5	6	7	8	9	.	,	;	:	;	:	:	:
5h			%	5	6	7	8	9	.	,	;	:	;	:	:	:	:
6h			&	6	7	8	9	.	,	;	:	;	:	:	:	:	:
7h			^	7	8	9	.	,	;	:	;	:	:	:	:	:	:
8h			(8	9	.	,	;	:	;	:	:	:	:	:	:	:
9h)	9	.	,	;	:	;	:	:	:	:	:	:	:	:
Ah			*	*	J	Z	j	z	.	,	;	:	;	:	:	:	:
Bh			+	*	K	C	k	c	.	,	;	:	;	:	:	:	:
Ch			,	*	L	1	l	1	.	,	;	:	;	:	:	:	:
Dh			-	=	M	J	m	j	.	,	;	:	;	:	:	:	:
Eh			.	>	N	^	n	^	.	,	;	:	;	:	:	:	:
Fh			/	?	O	_	o	_	.	,	;	:	;	:	:	:	:

Младшая часть байта (D0 – D3)

Матрицы начертания символов хранятся в памяти знакогенератора.

Память знакогенератора включает в себя **CGROM** (Character Generator ROM – ПЗУ знакогенератора), в которую на заводе-изготовителе загружены начертания символов таблицы ASCII. **Содержимое CGROM изменить нельзя.**

Для того, чтобы самостоятельно задать начертание нужных пользователю символов, в знакогенераторе имеется специальное ОЗУ – **CGRAM** (Character generator RAM).

Под ячейки CGRAM отведены первые (младшие) 16 адресов таблицы кодов.

Рисунок 12 – Таблица символов знакогенератора

Таблица 1 – Команды контроллера ЖКИ

Команда	Код										Описание	Время исполнения команды	
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0			
Очистка дисплея	0	0	0	0	0	0	0	0	0	0	1	Записывает код 0x20 (пробел) во все ячейки DDRAM, устанавливает счетчик адреса DDRAM в 0x00.	1,5 мс
Возврат в начальную позицию	0	0	0	0	0	0	0	0	0	1	x	Устанавливает счетчик адреса DDRAM в 0x00 и возвращает курсор в начальную позицию. Содержимое DDRAM не изменяется.	1,5 мс
Режим ввода	0	0	0	0	0	0	0	0	1	L/R	SH	Задаёт направление перемещения курсора (L/R) и разрешает сдвиг сразу всех символов (SH).	38 мс
Включение-выключение дисплея	0	0	0	0	0	0	0	1	D	C	B	Устанавливает/отключает биты, отвечающие за включения дисплея (D), отображение курсора (C), мерцание курсора (B).	38 мкс
Сдвиг курсора или видимой области дисплея	0	0	0	0	0	0	1	D/C	R/L	x	x	Бит D/C определяет то, что будет перемещаться – видимая область дисплея или курсор (при D/C = 1 перемещается видимая область, при D/C = 0 – курсор), R/L задает направление перемещения. DDRAM не изменяется	38 мкс

В таблице 1 приведены команды контроллера ЖКИ и время, необходимое для выполнения этих команд.

Команда	Код										Описание	Время исполнения команды
	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0		
Начальные установки	0	0	0	0	1	DL	N	F	x	x	Определяет разрядность шины интерфейса (DL = 1 8-бит, DL = 0 4-бита), количества строк на дисплее (N = 1 – две строки, N = 0 – одна строка) и размера символов (F = 1 – 5×11 точек, F = 0 5×8 точек).	38 мкс
Установка адреса CGRAM	0	0	0	1	A5	A4	A3	A2	A1	A0	Установка счетчика адреса CGRAM	38 мкс
Установка адреса DDRAM	0	0	1	A6	A5	A4	A3	A2	A1	A0	Установка счетчика адреса DDRAM	38 мкс
Чтение BF и счетчика адреса	0	1	BF	A6	A5	A4	A3	A2	A1	A0	Если BF = 1 то контроллер ЖКИ выполняет внутреннюю операцию. A6 – A0 – текущее значение адреса DDRAM.	0
Запись данных в RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Запись данных в ОЗУ (DDRAM или CGRAM)	38 мкс
Чтение данных из RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Чтение данных из ОЗУ (DDRAM или CGRAM)	38 мкс

Для того чтобы можно было определить, когда ЖКИ закончит свои внутренние операции, контроллер ЖКИ содержит специальный флаг занятости – **BUSY**-флаг (BF).

Если **контроллер занят** выполнением внутренних операций, то **BF установлен** (BF = 1), если же **контроллер готов** принять следующую команду, то **BF сброшен** (BF = 0).

Более простой способ организации обмена с контроллером ЖКИ заключается в том, что управляющий микроконтроллер, зная, сколько времени требуется для ЖКИ на обработку той или иной команды, после каждой передачи информации выдерживал определённую временную паузу.

Программу для работы с ЖКИ следует организовать в виде функций, выполняющих определенные действия, причем более сложные функции могут включать в себя более простые.

Простейшими могут быть такие подпрограммы, как

- функция, отправляющая команду контроллеру дисплея;
- функция, устанавливающая счетчик адреса;
- функция, записывающая данные в DDRAM.

В любом случае, общий алгоритм передачи информации контроллеру не изменится.

Руководствуясь диаграммой передачи информации (рисунок 7), можно определить последовательность действий при передаче информации в ЖКИ следующим образом:

- устанавливается требуемое значение на линии RS (0-команда, 1-данные),
- подать логический ноль (запись) на линию R/W,
- затем на линии E устанавливается логическая единица,
- после чего подаётся на шину DB значение передаваемого байта.
- Затем линия E устанавливается в 0.

Контроллер ЖКИ считывает байт с линий шины данных DB и состояние управляющих линий (RS, R/W) только после перехода на линии E 1→0. При этом, если временные задержки, указанные на диаграмме, меньше длительности машинного цикла, то ими можно пренебречь.

Программирование и управление контроллера ЖКИ

Управление контроллера ЖКИ ведется через интерфейс управляющей системы.

Основные объекты взаимодействия – регистры:

DR (*data register* – регистр данных) и **IR** (*instruction register* – регистр инструкций).

Выбор адресуемого регистра осуществляется линией RS,

если **RS = 0** – адресуется регистр команд (IR),

если **RS = 1** – регистр данных (DR).

Данные через регистр DR могут помещаться в видеопамять (DDRAM) или в ОЗУ знакогенератора (CGRAM) по текущему адресу, на который указывает счетчик адреса (AC).

Информация в регистре IR интерпретируется устройством выполнения команд как управляющая последовательность.

Чтение регистра IR возвращает в 7-ми младших разрядах **текущее значение счетчика адреса** (AC), а в старшем 8-м разряде **флаг занятости** (BF).

Видеопамять имеет общий объем 80 байтов и предназначена для хранения кодов символов.

Видеопамять жестко организована в две строки по 40 символов в каждой и не подлежит изменению.

Независимо от того, сколько строк будет иметь конкретный ЖКИ, **адресация видеопамети всегда производится в две строки по 40 символов.**

Операции чтения/записи для 4-х разрядной шины

Для случая, когда устройство взаимодействующее с ЖКИ имеет ограниченное количество линий подключения ввода/вывода, предусмотрен вариант подключения ЖКИ с использованием 4-битной шины «Команды/данные».

В этом режиме каждый байт данных передаётся по линиям DB4...DB7 последовательно двумя порциями – тетрадами, начиная со старшей.

Операции записи для 4-х разрядной шины	
1	Установка значения линии RS
2	Вывод значения старшей тетрады байта данных DB4...DB7
3	Установка линии E=1
4	Установка линии E=0
5	Вывод значения младшей тетрады байта данных DB4...DB7
6	Установка линии E=1
7	Установка линии E=0
8	Установка шины DB4...DB7 в состояние HI

Рисунок 13 – Последовательность управляющих сигналов для выполнения **операции записи** в режиме 4-х проводного подключения

Операции чтения для 4-х разрядной шины	
1	Установка значения линии RS
2	Установка линии R/W=1
3	Установка линии E=1
4	Считывание значения старшей тетрады байта данных DB4...DB7
5	Установка линии E=0
6	Установка линии E=1
7	Считывание значения младшей тетрады байта данных DB4...DB7
8	Установка линии E=0
9	Установка линии R/W=0

Рисунок 14 – Последовательность управляющих сигналов для выполнения **операции чтения** в режиме 4-х проводного подключения

Набор инструкций контроллера HD44780

Таблица 2 – Набор базовых инструкций контроллера HD44780

Инструкции	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Описание
Очистить дисплей	0	0	0	0	0	0	0	0	0	1	Очистка дисплея возврат курсора в начальную позицию
Возврат в начало	0	0	0	0	0	0	0	0	1	0	Возврат курсора в начальную позицию, ОЗУ не изменяется
Установка режима	0	0	0	0	0	0	0	1	I/D	S	Установка режима дисплея
Включение/выключение дисплея	0	0	0	0	0	0	1	D	C	B	Включение/выключение D – дисплея, C – курсора, B – мигающего курсора
Сдвиг курсора	0	0	0	0	0	1	S/C	R/L	0	0	Перемещение курсора и скроллинг, ОЗУ не изменяется
Установка функций	0	0	0	0	1	DL	N	F	0	0	Настройка интерфейса ввода/вывода
Установка CGRAM-адреса	0	0	0	1	ACG						Установка CGRAM-адреса, после этого передача или прием данных
Установка DDRAM-адреса	0	0	1	ADD						Установка DDRAM-адреса, после этого передача или прием данных	
Чтение адреса и флага занятости	0	1	BF	AC						Чтение флага занятости и счетчика адреса	
Запись данных	1	0	Запись данных								Запись в ОЗУ
Чтение данных	1	1	Чтение данных								Чтение из ОЗУ
	I/D=1 увеличение, I/D=0 уменьшение S=1 относительный скроллинг S/C=1 скроллинг, S/C=0 перемещение курсора R/L=1 сдвиг вправо, R/L=0 сдвиг влево DL=1 8 бит, DL=0 4 бита N=1 2 строки, N=0 1 строка F=1 матрица 5x10, F=0 матрица 5x7 BF=1 работа по внутренним инструкциям BF=0 прием внешних инструкций										DDRAM вывод содержимого ОЗУ CGRAM ОЗУ генератора символов ACG адрес CGRAM ADD адрес DDRAM, адрес курсора AC счетчик адреса, используемый для DDRAM и CGRAM

Управляющие флаги определяющие режимы работы

Управляющие флаги определяют режимы работы различных элементов контроллера.

Переопределение значений флагов производится специальными командами, записываемыми в регистр IR, при этом комбинации старших битов определяют группу флагов или команду, а младшие содержат собственно флаги.

Флаги, управляющие работой контроллера HD44780	
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

Значения управляющих флагов по умолчанию

После подачи питания управляющие флаги устанавливаются в исходное состояние согласно рисунку 15.

Значения управляющих флагов после подачи питания	
I/D=1	Режим увеличения сетчика на 1
S=0	Без сдвига изображения
D/L=1	8-ми разрядная шина данных
N=0	Режим развертки одной строки
F=0	Символы с матрицей 5 x 8 точек
D=0	Отображение выключено
C = 0	Курсор в виде подчеркика выключен
B = 0	Курсор в виде мерцающего знакоместа выключен

Рисунок 15 – Значения управляющих флагов по умолчанию

При записи или считывании буфера данных обращение осуществляется к ячейке, на которую в данный момент указывает курсор.

У двухстрочных ЖКИ первые 40 ячеек буфера данных обычно отображаются на верхней строке дисплея, а вторые 40 ячеек – на нижней.

Вывод на экран символа производится **записью его кода в регистр DR**. При этом символ размещается в **DDRAM** по текущему адресу, указываемому **AC**, а значение **AC** увеличивается или уменьшается на 1.

Чтобы произвести переустановку курсора на нужную позицию, необходимо присвоить **AC** соответствующее (новое) значение.

Когда производится последовательная запись символов и в результате заполняется вся строка, курсор автоматически переходит на вторую строку;

Но Есть одна особенность. Если необходимо принудительно установить курсор, например, в начало второй строки, то будет правильным присвоить **AC** значение **0x40** (64_{10}), а значение **0x28** (40_{10}) является неверным.

Значения адресов DDRAM в диапазоне **0x28...0x3F** (а равно и **0x68...0x7F**) являются **неопределёнными**, и **результаты работы с ними могут быть непредсказуемыми**.

Начало
инициализации

Задержка
> 15 мс

RS = 0
R/W = 0
D = 0x30

Начальная установка. Флаг занятости BF не может быть проверен управляющей системой до этой команды.

Задержка
> 4,7 мс

RS = 0
R/W = 0
D = 0x30

Начальная установка. Флаг занятости BF не может быть проверен управляющей системой до этой команды.

Задержка
> 100 мкс

RS = 0
R/W = 0
D = 0x30

Начальная установка. После этой команды флаг занятости BF может быть проверен управляющей системой.

Задержка
> 100 мкс

Установка количества строк и типа шрифта

Установка не может быть изменена после этой команды.

Включение дисплея и курсора

Очистка дисплея

Установка режима ввода со сбросом или без

Конец инициализации модуль готов к основной работе.

Конец
инициализации

Начальная инициализация ЖКИ

Рекомендованная выполнять следующую последовательность действий для инициализации.

1. **Выдержать паузу не менее 15 мс** между установлением рабочего напряжения питания (более 4,5 В) и выполнением каких-либо операций с контроллером.
2. Первой операцией **выполнить команду, выбирающую разрядность шины** (это должна быть команда независимо от того, какой разрядности интерфейс вы собираетесь использовать в дальнейшем), причем перед выполнением этой операции **не проверять значение флага BF**.
3. Далее опять **выдержать паузу не менее 4,1 мс и повторить команду выбора разрядности шины**, причем перед подачей команды вновь **не производить проверку флага BF**.
4. Следующим шагом необходимо вновь **выдержать паузу**, на этот раз **100 мкс**, и **в третий раз повторить команду установления разрядности шины**, вновь **без проверки BF**. **Эти три операции являются инициализирующими и призваны вывести контроллер в исходный режим работы** (то есть перевести в режим работы с 8-ми разрядной шиной) из любого состояния.
5. Следом за ними **нормальным порядком** (без выдерживания пауз, но с проверкой флага BF или выдерживанием временных пауз) **выполняется инициализация режимов работы с выдачей инициализирующей последовательности**, аналогичной указанной в таблице 1 (содержащей в том числе команду выбора необходимой разрядности шины).

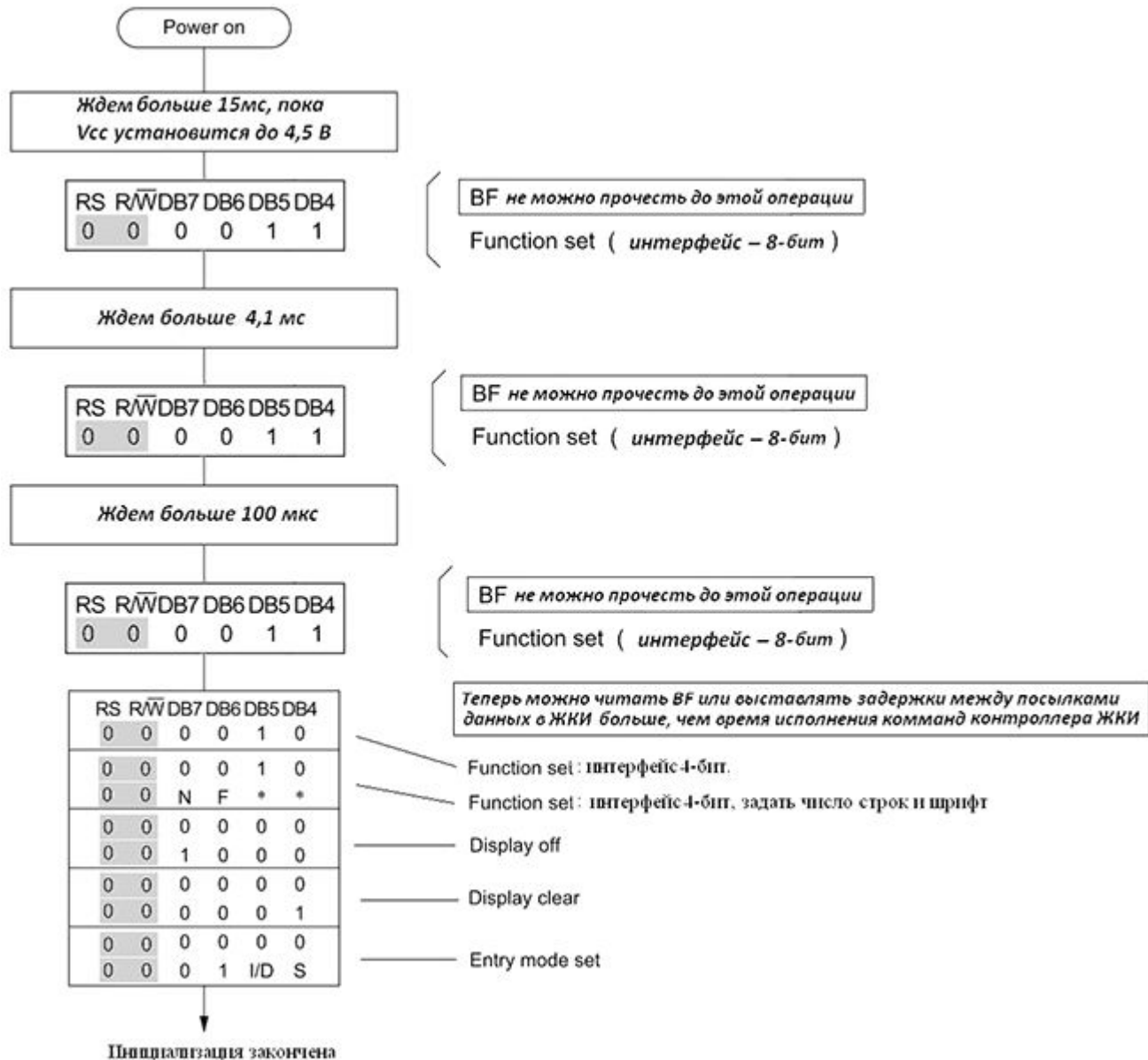
Инициализация ЖКИ в 4-разрядном режиме

Необходимо помнить! Когда объявляется режим работы с 4-разрядной шиной, то делается это обычно из 8-ми разрядного режима, который устанавливается автоматически после подачи напряжения питания, а значит НЕЛЬЗЯ адекватно объявить необходимое значение флагов **N** и **F**, располагающихся в младшей тетраде команды установки разрядности шины. **Поэтому команду необходимо повторить в уже установившемся 4-разрядном режиме путем последовательной передачи двух тетрад, то есть для 4-разрядного режима образом.**

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

Пример широко распространённой последовательности команд для инициализации ЖКИ:
0x33 (три раза), **0x0C**, **0x06**

- **0x33** устанавливает режим отображения 2 строк с матрицей 5x8 точек и работу с 8-битной шиной данных;
- **0x0C** включает отображение на экране ЖКИ без отображения курсора,
- **0x06** устанавливает режим автоматического перемещения курсора слева направо после вывода каждого символа.



Содержимое прошивки знакогенератора CGROM, CGRAM

		Старшие 4 бита (D4...D7)															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Младшие 4 бита (D0...D3)	0	CG RAM (0)			0	1	2	3			E	H	I	J	K	L	
	1	CG RAM (1)		4	5	A	B	C			F	G	H	I	J	K	
	2	CG RAM (2)		8	9	E	F	G			H	I	J	K	L	M	
	3	CG RAM (3)		#	0	1	2	3			4	5	6	7	8	9	
	4	CG RAM (4)		\$	4	D	T	W			3	7	B	Z	6	0	
	5	CG RAM (5)		%	5	E	U	V			N	O	A	X	4	7	
	6	CG RAM (6)		&	6	F	U	P	U		M	K	O	X	4	0	
	7	CG RAM (7)		7	7	G	W	9	W		L	B	A	I	7	E	
	8	CG RAM (0)		8	8	H	X	H	X		P	M	X	4	4	4	
	9	CG RAM (1)		9	9	I	Y	I	W		X	M	X	4	4	4	
	A	CG RAM (2)		*	*	J	Z	J	Z		0	K	X	4	4	4	
	B	CG RAM (3)		+	+	K	L	K	L		4	A	7	H	4	4	
	C	CG RAM (4)		,	,	L	0	L	0		4	M	7	H	4	4	
	D	CG RAM (5)		-	-	M	1	M	1		6	H	0	H	4	4	
	E	CG RAM (6)		.	.	N	2	N	2		H	A	7	Z	4	4	
	F	CG RAM (7)		/	/	O	3	O	3		0	T	7	.	0	4	

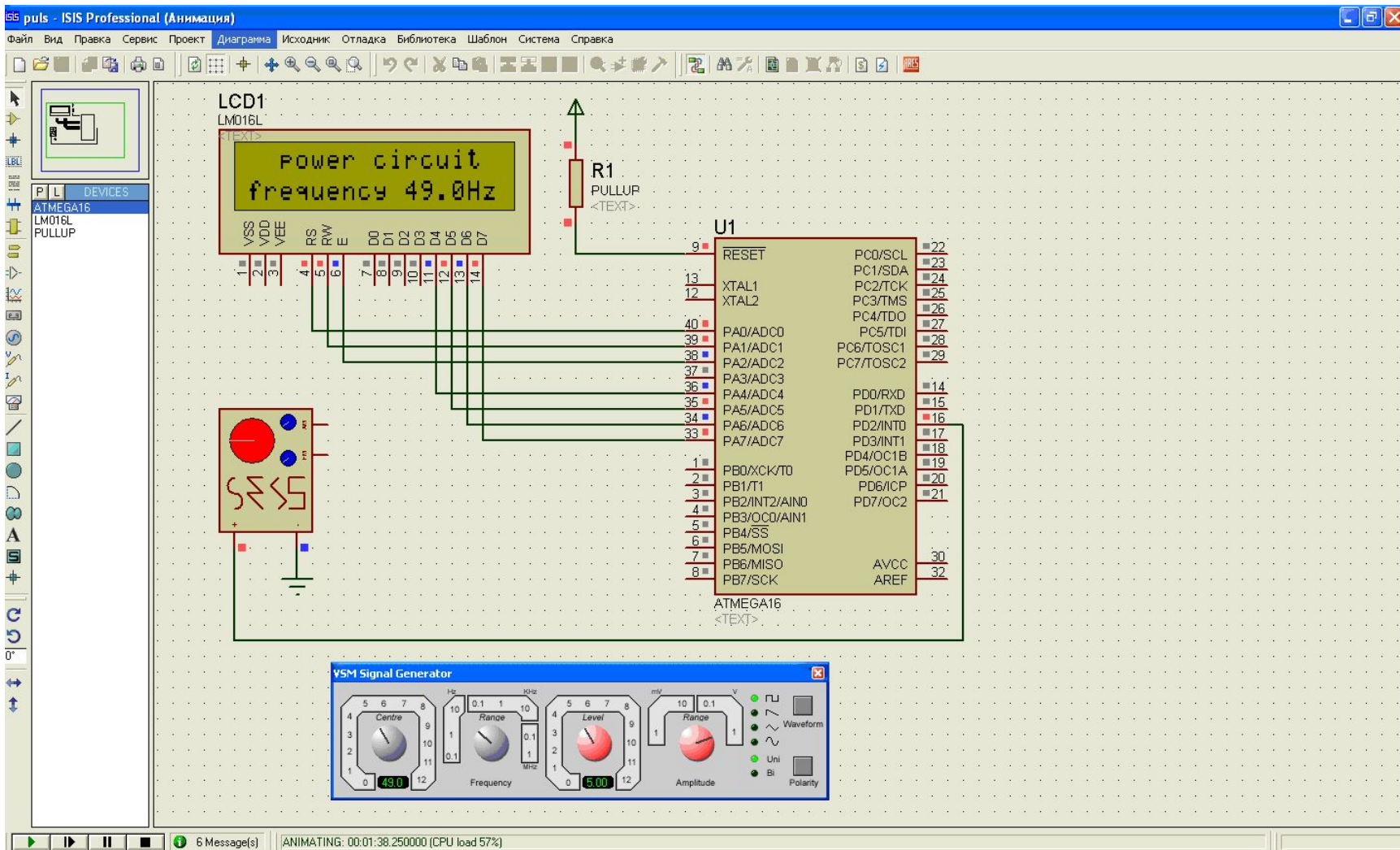


Рисунок 16 – Рабочие окна симулятора Proteus VSM

```

#define LCD_E_SET      PORTC  = PORTC | 0b00000010;
#define LCD_E_CLR      PORTC  = PORTC & 0b11111101;
#define LCD_RS_SET     PORTC  |= 0b00000001;
#define LCD_RS_CLR     PORTC  &= 0b11111110;

void main(void)
{
    PORTC = 0;
    DDRC  = 0xff;
    LCD_RS_SET;

    lcd_init();
    lcd_putsfmy("hello");

    while(1){
        };
    }

// инициализация ЖКИ
void lcd_init(void)
{
    delay_ms(150);          // задержка на выход контроллера дисплея на рабочий режим
    LCD_RS_CLR;
    delay_ms(40);
    lcd_putcharmy(0x33);    // рекомендованный порядок установки ЖКИ дисплея
    delay_ms(40);          // задержка между первой и второй уст 8-и битного режима
    lcd_putcharmy(0x33);    // вторая уст 8-и битного режима
    delay_ms(40);
    lcd_putcharmy(0x28);    // установка 4-х бит. режим,
                            // с двумя строками и матрицей 5*7
    delay_ms(40);
    lcd_putcharmy(0b00001100); // вкл. дисплея, курсор откл., мигание отк.
    delay_ms(40);
    lcd_putcharmy(0x06);    // русский режим направление
    LCD_RS_SET;             // переход в режим передачи данных
    delay_ms(40);
}

```



```

void lcd_puttetrade(char t)
{
    t &= 0x0F;          // Очищаем старшую тетраду входной переменной
                        // и кладем младшую тетраду в старшую
    t <<= 4;
    LCD_E_SET;        // установка 1 ножки Е МК дисплея ,подготовка к приему данных
    delay_ms(50);
    PORTC &= 0x0F;    // зачищаем место под данные
    PORTC |= t;       // запись данных в порт
    LCD_E_CLR;        // установка Ножки МК Е=0,команда начала обработки данных
    delay_ms(50);
}

```

```

// функция вывода строки (переменная это указатель строки,
// * - в ячейки хранится указатель)

```

```

void lcd_putsfmy(char* str)
{
    char d;
    d = *str;          // кладем в значение d нулевого символа строки
    while(d != 0x00){ // цикл пока d не будет равным нулевому значению,
                        // что скажет что мы на конце строки

        d = *(str++); // выбираем следующий символ строки
        lcd_putcharmy(d); // вывод на дисплей значения
    }
}

```

```

// вывод символа на дисплей

```

```

void lcd_putcharmy(char c)
{
    char hc = 0;
    hc = (c >> 4);
    lcd_puttetrade(hc);
    lcd_puttetrade(c);
}

```

Пользовательские символы

Из допустимых для размещения в **DDRAM** кодов символы с кодами **0x00...0x07** (и их дубликаты с кодами **0x08, ..., 0x0F**) имеют специальное назначение — это **переопределяемые символы**, графическое изображение которых может назначить **сам пользователь**, поместив соответствующую информацию в области **CGRAM**.

Для программирования **доступны 8 переопределяемых символов** в режиме с матрицей 5x7 точек.

Для каждого из восьми перепрограммируемых символов в CGRAM отводится по 8 ячеек памяти, каждая из которых соответствует одной строке точек в изображении символа.

Для символа 0 (код символа 0x00) адреса ячеек памяти — **0x00...0x07**, **для символа 1** (код символа 0x01) адреса ячеек памяти — **0x08...0x0F** и т.д. Таким образом, перепрограммируемая часть знакогенератора содержит 64 байта памяти (8x8).

Для кодирования матрицы используются горизонтально «уложенные» байты, **пять младших битов которых несут информацию о рисунке** (причём 1 означает, что сегмент будет включён), **4-й бит** каждого из 8 байтов матрицы **определяет левую колонку символа**, а **0-й — правую**.

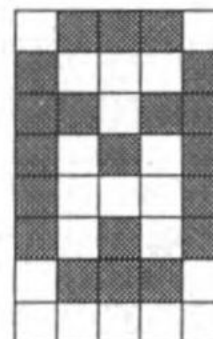
Старшие три бита не используются и могут иметь любые значения.

Адрес в CGRAM	Вид символа	Данные для записи
0x00		XXX00111
		XXX00011
		XXX00101
		XXX01000
		XXX11111
		XXX00010
		XXX00100
		XXX01000
0x01		XXX10001
		XXX01110
		XXX10001
		XXX10101
		XXX10001
		XXX01110
		XXX01010
		XXX11011
0x0F		XXX00010
		XXX00110
		XXX11010
		XXX11010
		XXX11010
		XXX00110
		XXX00010
		XXX00000

Рисунок 17 – Пользовательские символы в таблице CGRAM

Адрес строки символа 2 (код символа 0x02)

- 1-я строка 0x10 (16)
- 2-я строка 0x11 (17)
- 3-я строка 0x12 (18)
- 4-я строка 0x13 (19)
- 5-я строка 0x14 (20)
- 6-я строка 0x15 (21)
- 7-я строка 0x16 (22)
- 8-я строка 0x17 (23)



4 3 2 1 0
(биты)

Данные строки символа 2 (код символа 0x02)

- 0b00001110=0x0E
- 0b00010001=0x11
- 0b00011011=0x1B
- 0b00010101=0x15
- 0b00010001=0x11
- 0b00010101=0x15
- 0b00001110=0x0E
- 0b00000000=0x00

Рисунок 18 – Пример кодирования одного символа (символ 2, код символа 0x02) – незначащие разряды показаны светлым

Особенность вывода на ЖКИ русского текста

Прежде всего **знакогенератор ЖКИ должен иметь русские буквы.**

Строку, предназначенную для вывода на ЖКИ, содержащую английский текст и цифры, в программе Си можно записать несколькими способами, т. к. их коды Windows и коды английского знакогенератора ЖКИ-дисплея совпадают.

Пример:

```
/* Эквивалентная запись определения строки «Text», содержащей английские буквы */  
char string[] = «Text»;  
char string[] = { 'T', 'e', 'x', 't', '\0'};  
char string[] = {0x54, 0x65, 0x78, 0x74, 0x00};
```

Коды Windows русских букв НЕ СОВПАДАЮТ с их кодами знакогенератора ЖКИ-дисплея.

Поэтому русские буквы сходные с английскими берутся из английского знакогенератора, а остальное — из усечённого русского.

Поэтому строку, предназначенную для вывода на ЖКИ, содержащую русский текст, в программе Си можно записать только одним способом, а именно перечислив коды знакогенератора соответствующих букв.

Пример: /* Запись определения строки "Текст", содержащей русские буквы */
char string[] = {0x54, 0x65, 0xBA, 0x63, 0xBF, 0x00};

Алгоритм чтения/записи в LCD контроллер HD44780

Направление, а также команда/данные определяются уровнем сигнала на выводе RS, а чтение и запись осуществляется по переходу строба (вывод E) из 1 в 0

Инициализация портов

RS, RW, E - в режим вывода.

DB7...DB0 - в режим ввода. (Пока их можно не трогать)

Ожидание готовности, чтение флага занятости.

Порт данных на вход с подтяжкой ($DDR_x = 0$, $PORT_x = 1$)

RS = 0 (команда)

RW = 1 (чтение)

E = 1 (Подготовить линию к чтению)

Пауза (например, 14 тактов процессора на 8 МГц)

E = 0 (Прочитать)

Читаем из порта. Если бит 7 (Busy flag) установлен, то повторяем все заново, пока не сбросится.

Запись команды

Ожидание готовности

RS = 0 (команда)

RW = 0 (запись)

E = 1 (Подготовить линию к записи)

Порт на выход

Вывести в порт код команды

Пауза

E = 0 (Записать команду)

Порт на вход, на всякий случай.

Запись Данных

Ожидание готовности

RS = 1 (Данные)

RW = 0 (запись)

E = 1 (Подготовить линию к записи)

Порт на выход

Вывести в порт код команды

Пауза

E = 0 (Записать)

Порт на вход, на всякий случай.

Чтение команды

Ожидание готовности

Порт данных на вход с подтяжкой (DDRx = 0, PORTx = 1)

RS = 0 (команда)

RW = 1 (чтение)

E = 1 (Подготовить линию к чтению. В этот момент данные из LCD устанавливаются на шину)

Пауза

Считываем данные с порта

E = 0 (Прочитать команду)

Чтение Данных

Ожидание готовности

Порт данных на вход с подтяжкой (DDRx = 0, PORTx = 1)

RS = 1 (Данные)

RW = 1 (чтение)

E = 1 (Подготовить линию к чтению. В этот момент данные из LCD устанавливаются на шину)

Пауза

Считываем данные с порта

E = 0 (Прочитали данные!)

С четырех разрядной шиной все точно также, только там каждая операция чтения/записи делается за два переключения строба.

Запись:

$E = 1$

Пауза

Выставили в порт старшую тетраду

$E = 0$ (записали)

Пауза

$E = 1$

Пауза

Выставили в порт младшую тетраду

$E = 0$ (записали)

Чтение

$E = 1$

Пауза

$E = 0$ (Читаем из порта старшую тетраду)

Пауза

$E = 1$

Пауза

$E = 0$ (Читаем из порта младшую тетраду)